Анализ данных

доц. М.Б. Вольфсон, СПбГУТ

Литература

- Вольфсон, М. Б. Анализ данных: учеб. пособие. - СПб. : СПбГУТ, 2015. - 81 с.
- Федин, Ф. О. Анализ данных: учеб. пособие. – М. : МГПУ, 2012 . – Ч. 1. Подготовка данных к анализу. - 204 с.
- Федин, Ф. О. Анализ данных: учеб. пособие. – М. : МГПУ, 2012. – Ч. 2. Инструменты Data Mining. - 308 с.
- Паклин Н., Орешков В. Бизнес-аналитика: от данных к знаниям : учеб. пособие. 2-е изд., испр. – СПб.: Питер, 2013. – 704 с.
- Миркин Б.Г. Введение в анализ данных: учебник и практикум для бакалавриата и магистратуры. – М.: Издательство Юрайт, 2014. – 174 с.

Содержание

- Введение
- Big Data
- Системы поддержки принятия решений
- Хранилища данных
- Оперативный анализ данных (OLAP)
- Data Mining
- Визуализация данных

Цифровая вселенная

К 2020 г. прогнозируется, что общее количество оцифрованной информации в мире превысит 40 зеттабайт.

- 40 зеттабайт - это в 57 раз больше, чем количество песчинок на всех пляжах планеты.
- Если записать 40 зеттабайт данных на Blue-ray, общий вес дисков будет равен весу 424 авианосцев.

Объемы информации будут удваиваться каждые 2 года.

Почти 75% данных являются копиями. Используется менее 3 из 23% потенциально полезных данных.

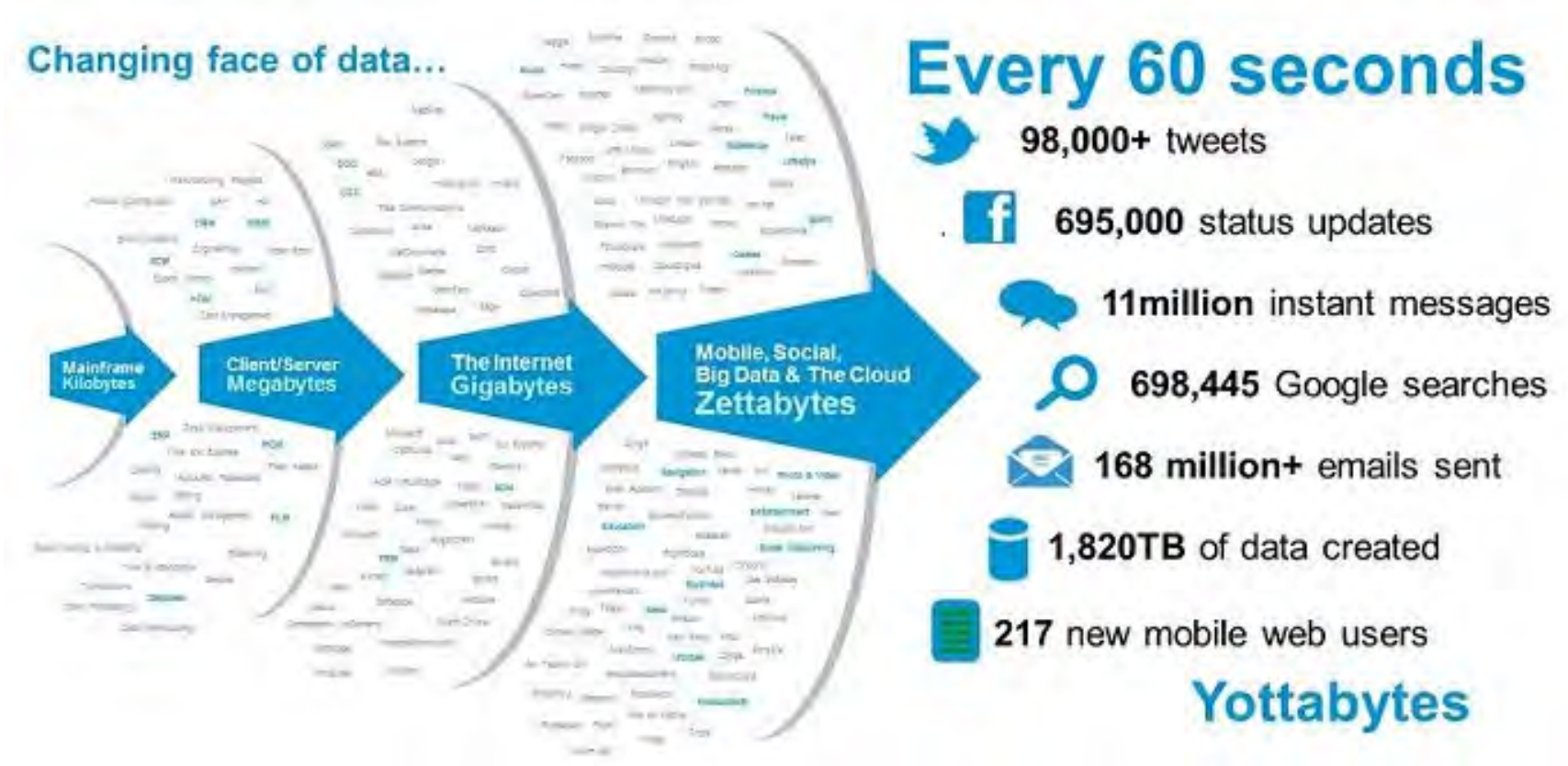
Только 5% информации в мире анализируется.

Объем информации в России ~ 155 экзабайт (2,4% от мирового объема).

Объем (размер) данных

Название	Размер по ГОСТ 8.417-2002 (приставки по СИ)	Символ	Примечание: размер по стандартам МЭК
байт	8 бит	В	
килобайт	10^3 В	КВ	2^{10} =1024 байт
мегабайт	10^6 В	МВ	2^{20} байт
гигабайт	10^9 В	GB	2^{30} байт
терабайт	10^{12} В	ТБ	2^{40} байт
петабайт	10^{15} В	РВ	2^{50} байт
эксабайт	10^{18} В	ЕВ	2^{60} байт
зеттабайт	10^{21} В	ZB	2^{70} байт
йоттабайт	10^{24} В	YB	2^{80} байт

Цифровая вселенная



Где хранить?

В 2013 г. совокупная доступная емкость систем хранения соответствовала 33% объема цифровой информации. К 2020 г. ее будет достаточно для хранения менее чем 15%.

В 2013 г. менее 20% данных размещалось в облаке, к 2020 г. эта величина превысит 30%.

Количество устройств, которые можно подключить к интернету приближается к 200 млрд, из которых 14 млрд (~7%), уже активно передают данные (**Интернет вещей**).

Данные от таких устройств составляют 2% от мирового объема информации. К 2020 г. уже 32 млрд подключенных устройств будут генерировать 10% общего объема данных во всем мире.

Мир в 2020 году

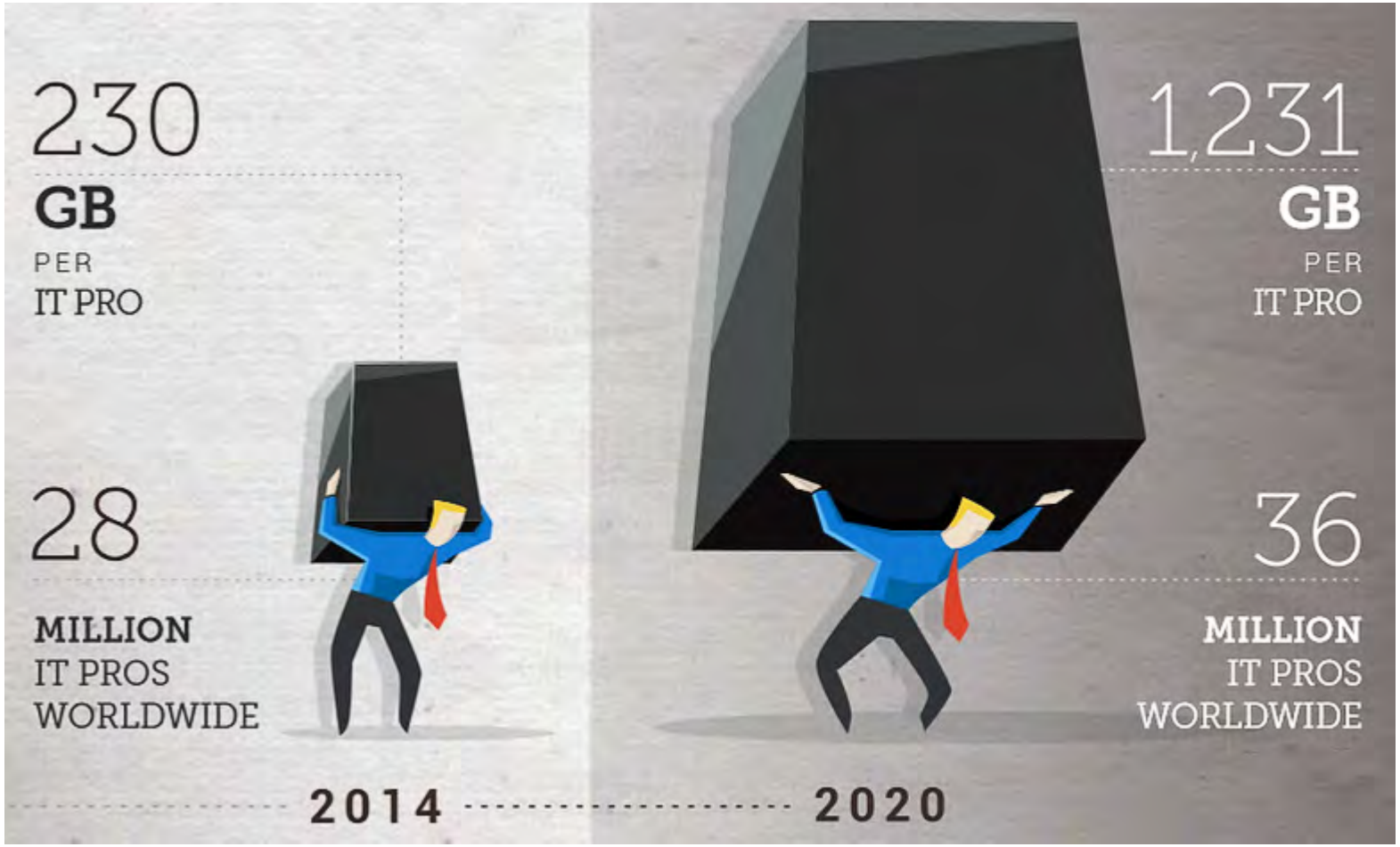
44 трлн Гб
составит накопленный объем цифровых данных

1/3 всех этих данных будет храниться в облаке

1,7 Мб

Данных на человека будет создаваться **каждую секунду**

Объем данных на IT-специалиста



Большие данные

Большие данные (*Big Data*) — совокупность подходов, инструментов и методов обработки данных огромных объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста и распределения по многочисленным узлам вычислительной сети.

Данные

Данные – ресурс для получения информации. Они должны быть представлены в форме, пригодной для хранения, передачи и обработки.

Структурированные данные - организуют в ряды и колонки строго определенного формата, чтобы приложения могли извлекать данные и эффективно обрабатывать их. Обычно хранятся с применением СУБД (~ 20%).

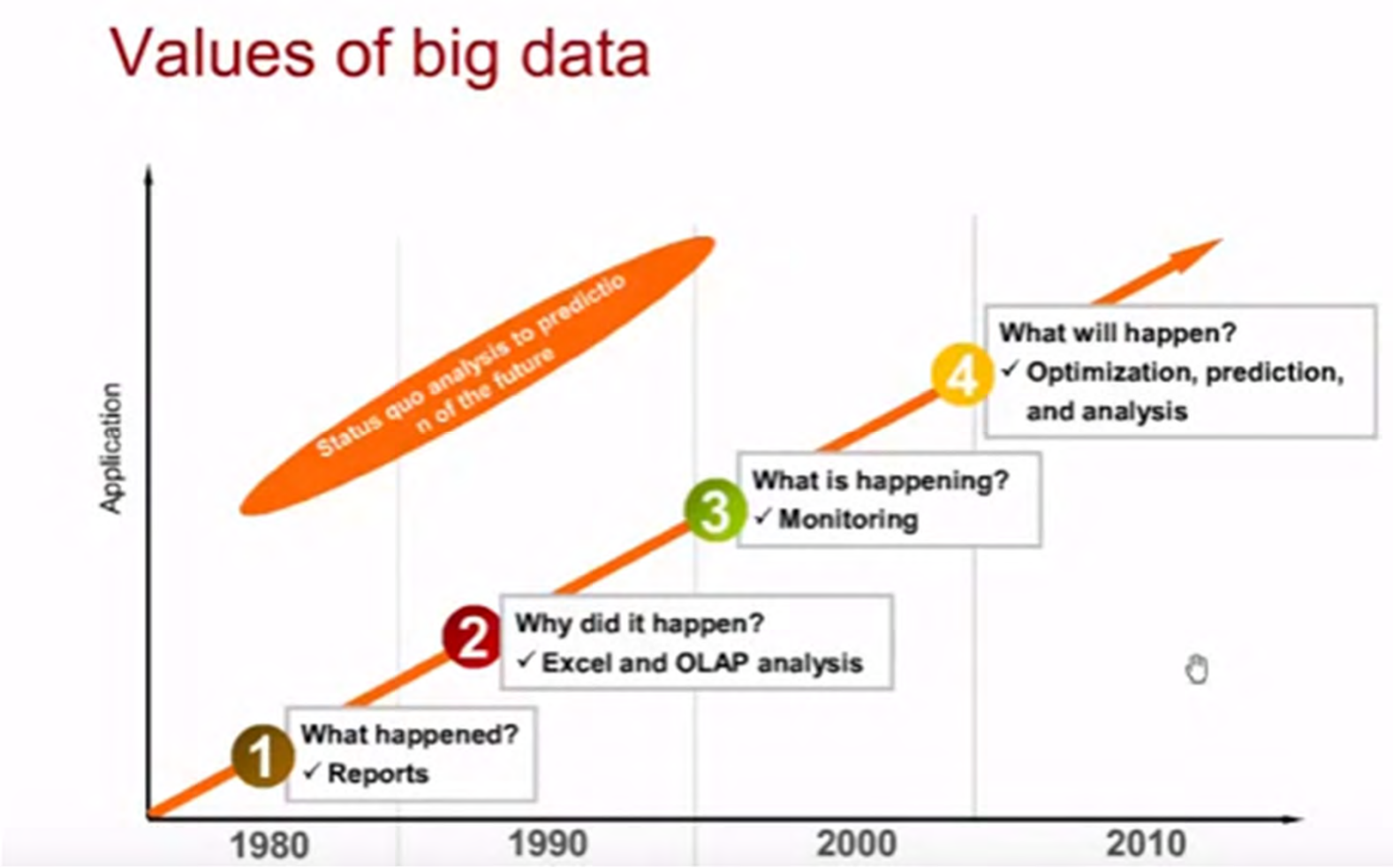
Неструктурированные данные: офисная документация, графические данные, чертежи, веб-страницы, сообщения электронной почты и IM, видео- и аудиофайлы и другие мультимедийные активы (~ 80%) .

Big Data

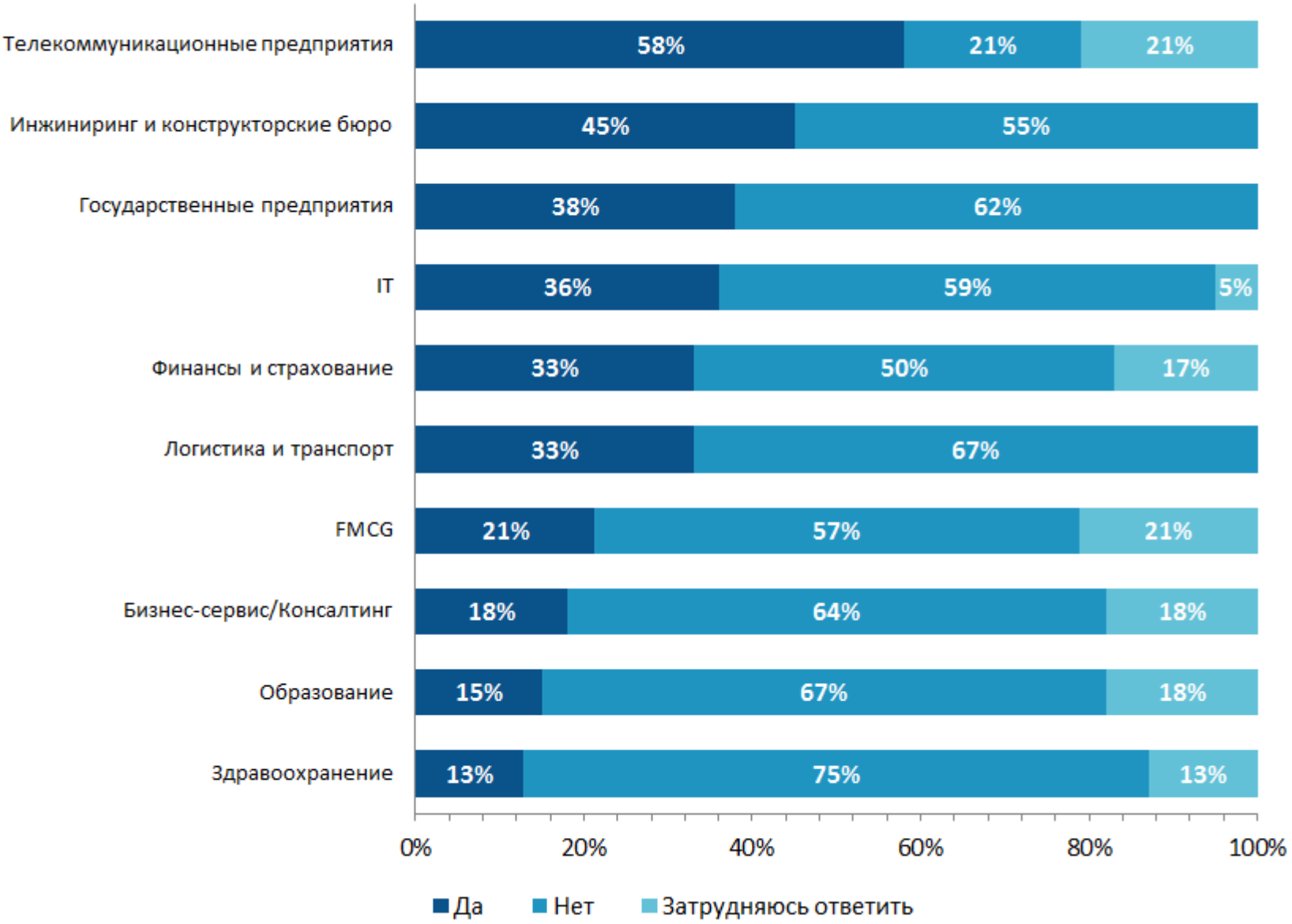
В качестве определяющих характеристик для больших данных отмечают *«5 V's»*:

- ✓ объём (*volume*, физический объём данных),
- ✓ скорость (*velocity,* как скорость прироста, так и скорость обработки и получения результатов),
- ✓ многообразие (*variety*, одновременная обработка различных типов данных; P2P, P2M, M2M),
- ✓ точность (*veracity*, достоверность: верификация и валидация данных),
- ✓ ценность (*value*, экономический эффект для пользователей).

Эволюция работы с данными (повышение ценности данных)



Компании из каких отраслей внедрили технологии Больших Данных?



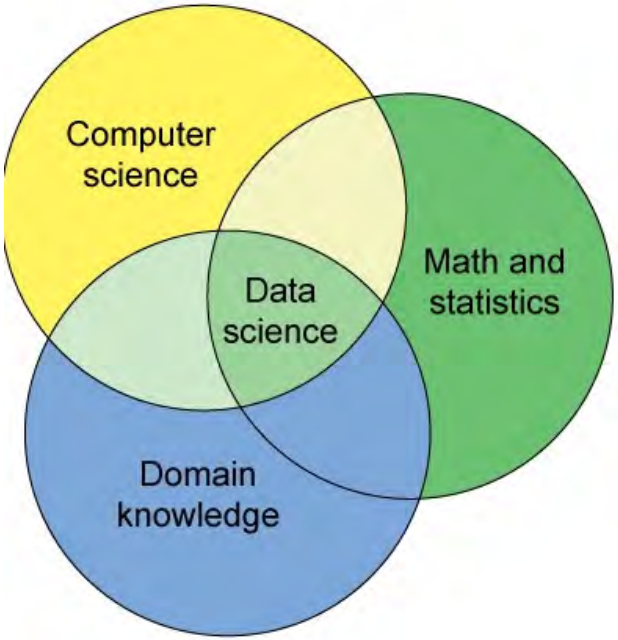
Источник: Tech Pro Research

Data Science

Наука о данных (*data science*) — раздел информатики, изучающий проблемы анализа, обработки и представления данных в цифровой форме.

Специализации в Data Science

- Data Scientist
- Data Engineer



Data Scientist

Сферы деятельности:

- Сбор и преобразование большого количества данных.
- Решение бизнес-задач с использованием данных.
- Работа с различными языками программирования, включая SAS, R и Python.
- Работа с базами данных MySQL и Postgres.
- Работа с платформой и сервисами Hadoop, Weka.
- Работа со статистикой, включая статистические тесты и распределения.
- Использование аналитических методов, таких как машинное обучение, глубокое обучение и текстовая аналитика.
- Сотрудничество с ИТ и бизнесом в равной мере.
- Поиск порядка и шаблонов данных, а также выявление тенденций, которые могут помочь в достижении конечного бизнес-результата.
- Визуализация данных и создание отчетности.

Средняя зарплата в США Data Scientist — 91 тыс. $ в год. В России - от 60-70 тыс. руб. в месяц для новичков и до 300 для опытных специалистов.

Data Engineer

Компаниям следует делегировать техническую работу инженеру данных (80% работы). Data Engineer специализируется на организации процесса сбора, очистки и предобработки данных («garbage in — garbage out»).

1) Понимание сути и сбор данных (источники, типы, структуры данных, организация доступа)
2) Построение архитектуры процесса обработки данных (обработка в режиме реального времени и офлайн).
3) Превращение моделей в готовый продукт или сервис (интеграция в сайт, связь с базой данных и т.п.).

Знание современных технологий и подходов в области обработки данных: MapReduce, Hadoop, Spark, Aerospike, Redis, Storm и т.д.

Знание языков программирования (SAS, R и Python) и библиотек (Pandas, Numpy, Scikit-learn).

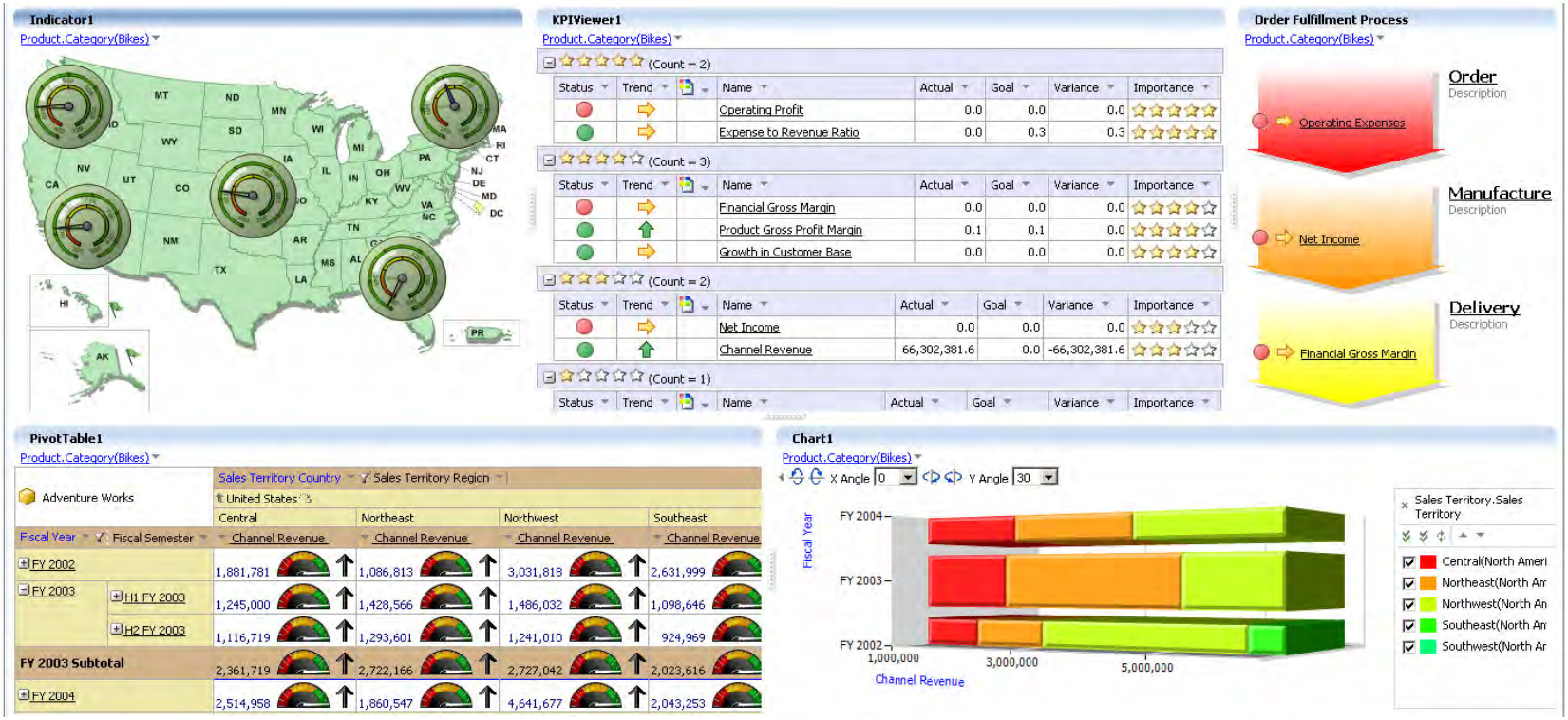
Знание ETL (Extract/Transform/Load – извлечение/преобразование/загрузка).

Типовые цели проекта по аналитике больших данных

- Поиск нового: редких фактов, один из миллионов или миллиардов объектов или событий
- Поиск классов: нахождение новых типов объектов и поведений
- Поиск ассоциаций: нахождение необычных (невероятных) совместно случающихся ассоциаций, идентификация связей между различными вещами, людьми или событиями

Business Intelligence

Business Intelligence – это процесс превращения полученных данных в знания о бизнесе, которые используются для принятия улучшенных решений.



Пример информационной панели (Dashboard)

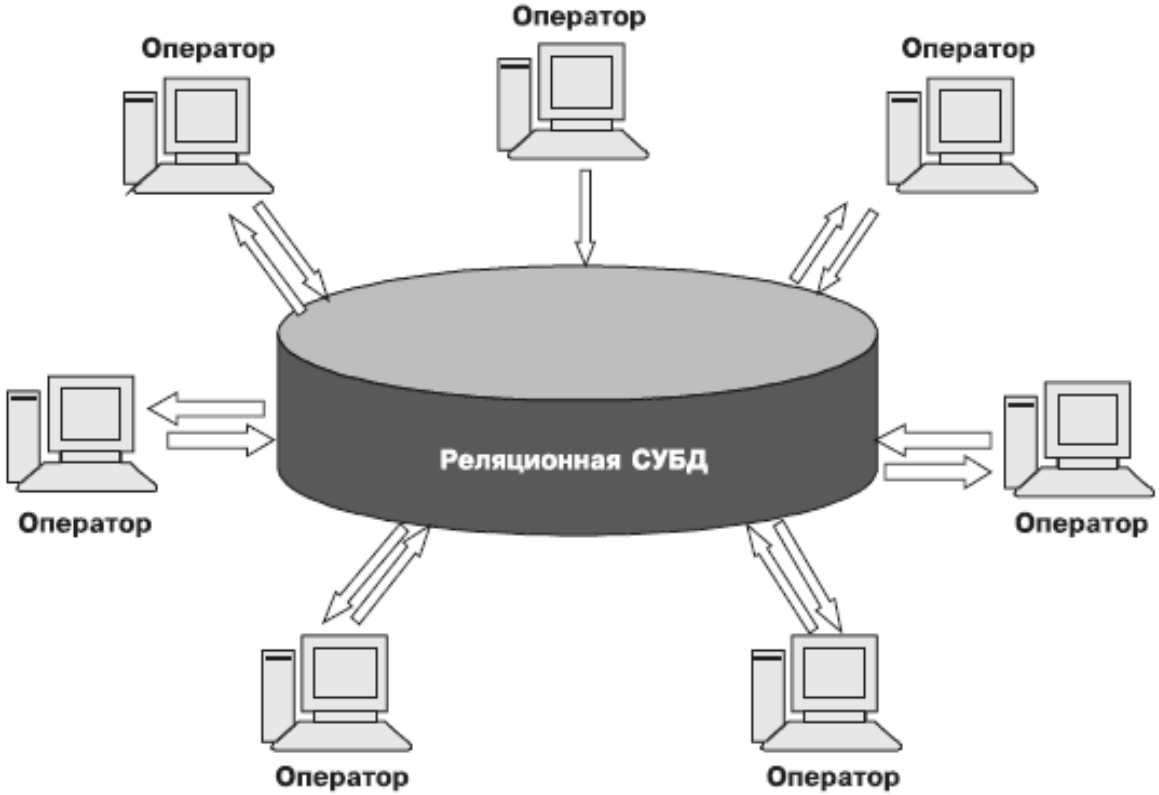
Системы оперативной обработки информации

OLTP (On-Line Transaction Processing) — оперативная, т.е. в режиме реального времени, обработка транзакций.

Транзакция — некоторый набор операций над базой данных, который рассматривается как единое завершенное, с точки зрения пользователя, действие над некоторой информацией, обычно связанное с обращением к базе данных.

Главное требование — быстрое обслуживание относительно (не более нескольких секунд) простых запросов большого числа пользователей.

Обобщенная структура системы OLTP



Примеры: бронирование авиабилетов или оплата услуг телефонных компаний. Два общих свойства: очень большое число клиентов и непрерывное поступление информации.

Системы поддержки принятия решений (постановка задачи)

Со временем в OLTP начали аккумулироваться большие объемы данных.

Сбор данных – не самоцель!

Появилась потребность в ИС, которые позволяли бы проводить глубокую аналитическую обработку (поиск закономерностей, вывод из них правил, принятие решений и прогнозирование их последствий).

Системы поддержки принятия решений

Информационные системы поддержки принятия решений (**Decision Support System, DSS, СППР**) – ориентированы на аналитическую обработку данных с целью получения знаний, необходимых для разработки решений в области управления (BI).

В зависимости от данных, с которыми работают СППР, выделяют два основных их типа: **статические** и **динамические**.

- **статические DSS** (информационные системы руководителя, Executive Information Systems — EIS)

Содержат в себе предопределенные множества запросов и неспособны ответить на все вопросы, которые могут возникнуть при принятии решений. Результатом работы такой системы, как правило, являются многостраничные отчеты, которые нельзя изменить без привлечения программиста.

- **динамические DSS**.

Ориентированы на обработку нерегламентированных, неожиданных (ситуативных) запросов аналитиков к данным.

Отличия СППР и OLTP-систем

Свойство	OLTP-система	СППР
Цели использования данных	Быстрый поиск, простейшие алгоритмы обработки	Аналитическая обработка с целью поиска скрытых закономерностей, построения прогнозов и моделей
Уровень обобщения (детализации) данных	Детализированные	Как детализированные, так и обобщенные (агрегированные)
Требования к качеству данных	Возможны некорректные данные (ошибки регистрации, ввода и т.д.)	Ошибки в данных не допускаются, поскольку могут привести к некорректной работе аналитических алгоритмов
Формат хранения данных	Данные могут храниться в различных форматах в зависимости от приложения, в котором они были созданы	Данные хранятся и обрабатываются в едином формате
Время хранения данных	Как правило, не более года (в пределах отчетного периода)	Годы, десятилетия
Изменение данных	Данные могут добавляться, изменяться и удаляться	Допускается только пополнение; ранее добавленные данные изменяться не должны, что позволяет обеспечить их хронологию
Периодичность обновления	Часто, но в небольших объемах	Редко, но в больших объемах
Доступ к данным	Должен быть обеспечен доступ ко всем текущим (оперативным) данным	Должен быть обеспечен доступ к историческим (то есть накопленным за достаточно длительный период времени) данным
Характер выполняемых запросов	Стандартные, настроенные заранее	Нерегламентированные, формируемые аналитиком «на лету» в зависимости от требуемого анализа
Время выполнения запроса	Несколько секунд	До нескольких минут (важно, но не критично)
Число пользователей	Поддержкой большого числа пользователей	Небольшое число пользователей (аналитики)

Поддержка принятия решений может выполняться в трех базовых сферах

1) Область детализированных данных (OLTP-системы).

Цель - поиск информации (информационно-поисковые системы).

2) Сфера агрегированных показателей (OLAP-системы).

Цель - обобщение информации и многомерный анализ.

3) Сфера закономерностей (Data Mining).

Цель - поиск закономерностей в накопленной информации, построение моделей и правил, которые объясняют найденные аномалии и/или прогнозируют развитие некоторых процессов.

Хранилища данных

Билл Инмон (1989г.): "**Хранилище данных** (Data Warehouse) - это предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки процесса принятия управляющих решений".

Хранилище данных (ХД) управляет данными, которые были собраны как из OLTP-систем, так и из внешних источников данных, и которые длительный период времени хранятся в системе.

Основные характеристики хранилищ данных

- Ориентация на предметную область. Учитывает специфику предметной области (клиенты, товары, продажи), а не прикладных областей деятельности (выписка счетов, контроль запасов, продажа товаров).
- Интегрированность и внутренняя непротиворечивость. Поскольку данные в хранилище поступают из разных источников, необходимо привести их к единому формату.
- Содержит исторические данные с привязкой ко времени (учет хронологии).
- Неизменяемость. Данные не обновляются в оперативном режиме, а лишь регулярно пополняются.
- Поддержка высокой скорости получения данных из хранилища.
- Предназначено для проведения анализа и принятия стратегических решений.
- Полнота (хранит как подробные сведения, так и частично и полностью обобщенные данные) и достоверность хранимых данных.
- Поддержка качественного процесса пополнения данных.
- Обслуживает относительно малое количество работников руководящего звена и аналитиков.

Устройства хранения

Устройства для хранения данных также называются **хранилищами**.

Тип используемого хранилища зависит от типа данных и их применения: DVD, HDD, внешние дисковые массивы и ленты, RAID-массивы и т.п.

Облачные хранилища: Microsoft, Amazon, Dell EMC, Google и др.

Виды данных

Детализированные данные поступают непосредственно из источников данных и соответствуют элементарным событиям, регистрируемым OLTP-системами. Такими данными могут быть ежедневные продажи, количество произведенных изделий и т.д. Это неделимые значения.

Агрегированные данные – обобщенные данные. Если обобщить данные в пределах недели или месяца и взять сумму, среднее, максимальное и минимальное значения за соответствующий период, то полученный ряд может оказаться более информативным, чем конкретные значения.

Поскольку один и тот же набор детализированных данных может породить несколько наборов агрегированных данных с различной степенью обобщения, объем существенно ХД существенно возрастает. Однако, если бы они вычислялись в процессе выполнения запросов, время выполнения запроса увеличилось бы в несколько раз.

Метаданные

Метаданные («данные о данных») необходимы для описания значения и свойств информации с целью лучшего ее понимания, использования и управления ею.

Примеры: Книга (аннотация, глоссарий, оглавление, номера страниц, об авторах и издательстве), Фотография (дата, формат, размер, координаты).

Одно из основных назначений метаданных — повышение эффективности поиска.

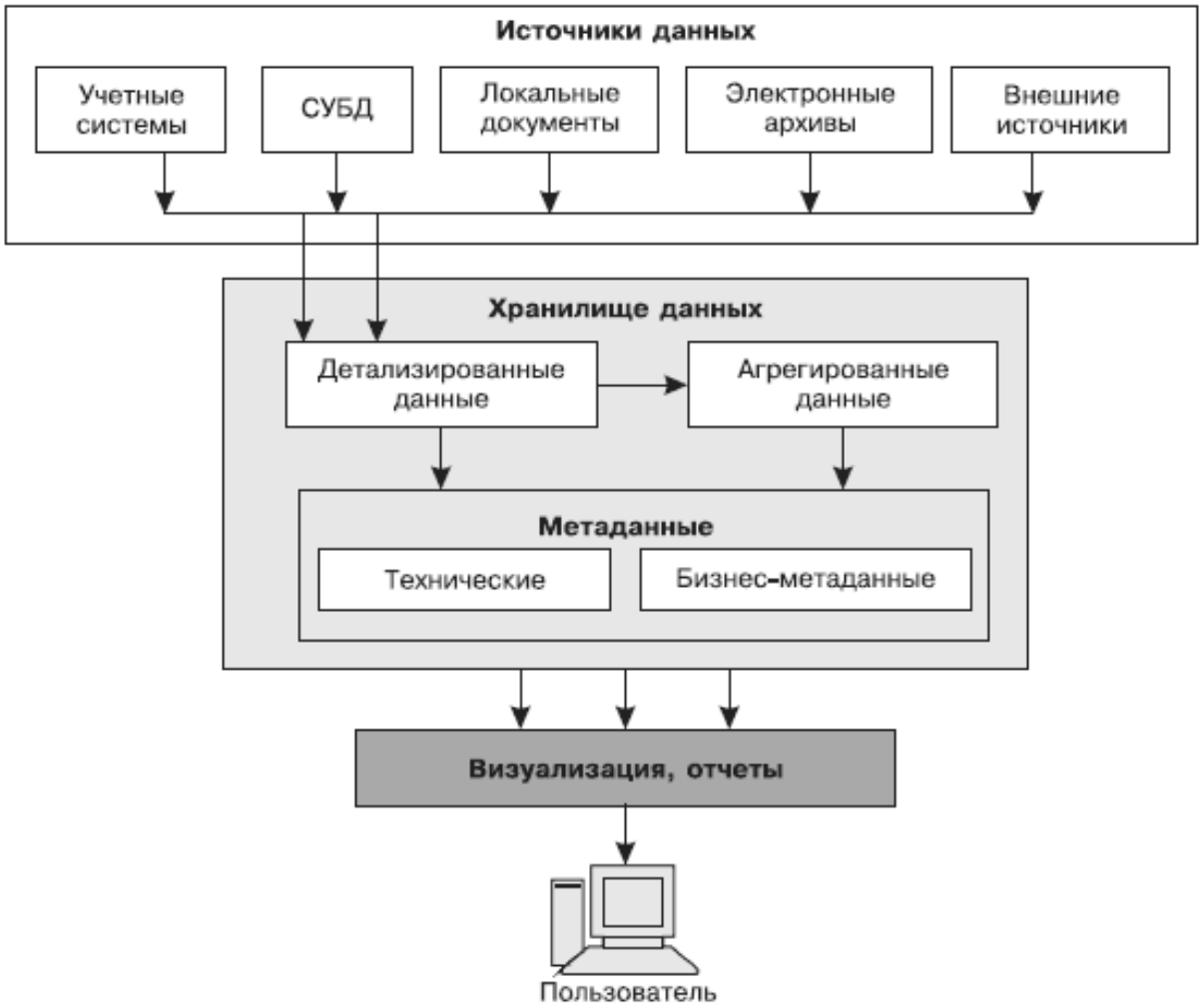
Метаданные хранятся отдельно от данных в **репозитарии метаданных**.

Метаданные

Два уровня метаданных: технический (административный) и бизнес-уровень.

- **Технический уровень** содержит метаданные, необходимые для обеспечения функционирования хранилища (статистика загрузки данных и их использования, описание модели данных и т.д.).
- **Бизнес-метаданные** описывают объекты предметной области — атрибуты объектов и их возможные значения, соответствующие поля в таблицах и т.д.

Обобщенная концептуальная схема хранилища данных



Извлечение данных (ETL)

Извлечение данных из разнотипных источников и перенос их в ХД с целью дальнейшей аналитической обработки связаны с рядом проблем:

- Исходные данные расположены в источниках самых *разнообразных типов и форматов*, созданных в различных приложениях, и, кроме того, могут использовать различную кодировку. Для решения задач анализа данные должны быть преобразованы в единый универсальный формат, который поддерживается ХД и аналитическим приложением.
- Данные в источниках обычно *излишне детализированы*, тогда как для решения задач анализа в большинстве случаев требуются обобщенные данные.
- Исходные данные, как правило, являются «*грязными*» (отсутствующие, неточные или бесполезные данные), что мешает их корректному анализу.

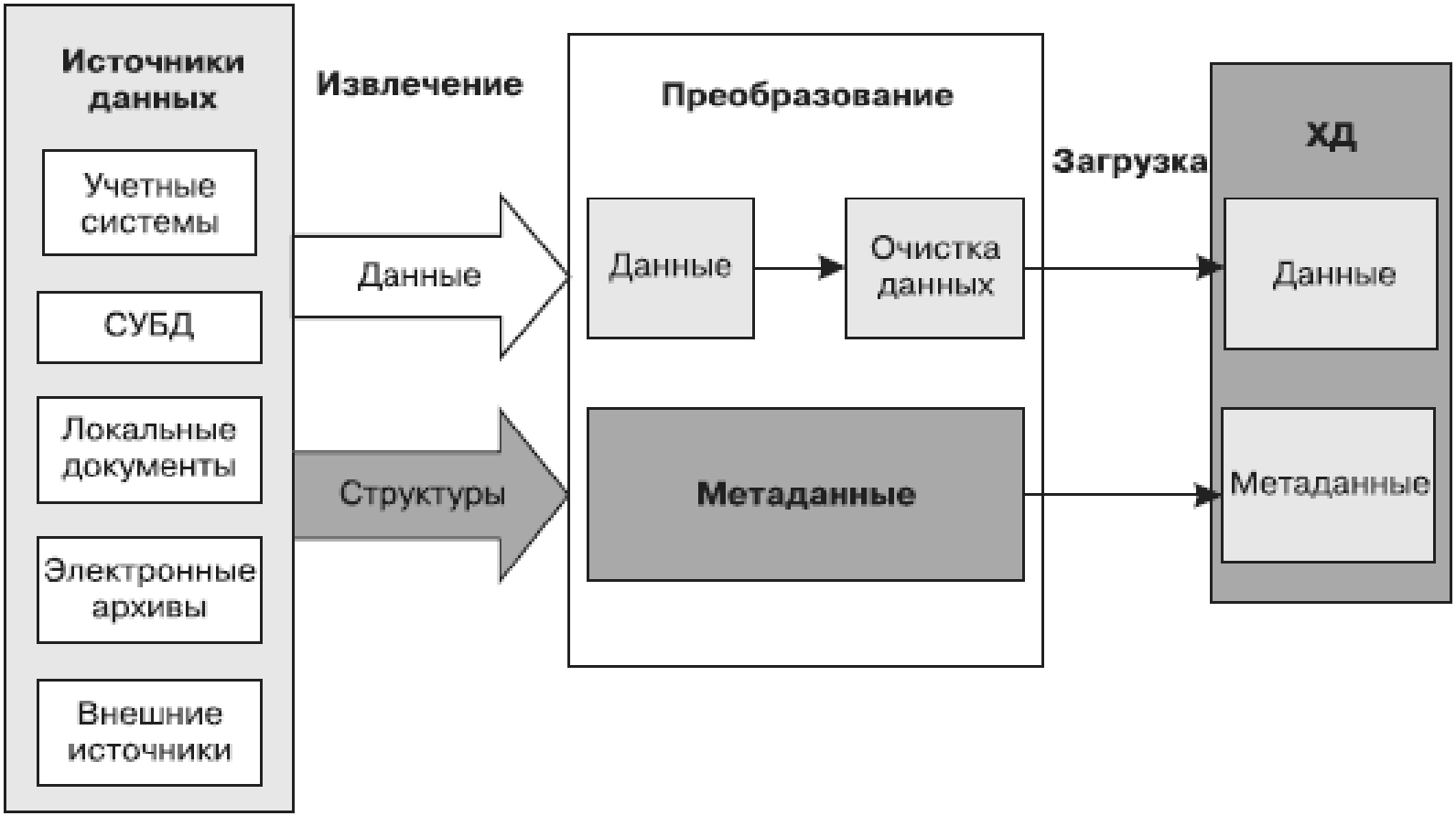
Извлечение данных (ETL)

ETL (extraction, transformation, loading – извлечение, преобразование и загрузка данных) - комплекс методов, реализующих процесс переноса исходных данных из различных источников в аналитическое приложение или поддерживающее его хранилище данных.

Приложения ETL извлекают информацию из источников, преобразуют ее в формат, поддерживаемый системой хранения и обработки, а затем загружают в нее преобразованную информацию.

Процесс ETL реализуется путем либо разработки приложения ETL, либо создания комплекса встроенных программных процедур (SQL), либо использования ETL-инструментария.

Обобщенная структура процесса ETL



Извлечение данных

Данные извлекаются из одного или нескольких источников и подготавливаются к преобразованию. Из источников должны извлекаться не только сами данные, но и информация, описывающая их структуру, из которой будут сформированы метаданные.

Процесс извлечения данных может выполняться ежедневно, или еженедельно. Существует целый класс систем бизнес-аналитики, которые требуют извлечения данных в режиме реального времени: например, системы, анализирующие биржевые операции.

Преобразование данных

Процесс преобразования данных включает в себя:

- Преобразование типов данных
- Преобразования, связанные с нормализацией схемы данных
- Преобразования ключей.
- Преобразования, связанные с обеспечением качества данных в ХД (очистка данных).

Очистка данных

Данные в ХД должны быть:

- точными – должны содержать правильные количественные значения метрик;
- полными – пользователи должны иметь доступ ко всем релевантным данным;
- согласованными – никакие противоречия в данных не допускаются (агрегаты должны точно соответствовать детализированным данным);
- уникальными – одни и те же объекты должны иметь одинаковые наименования и идентифицироваться одинаковыми ключами;
- актуальными – пользователи должны знать, с какой частотой данные обновляются (т.е. на какую дату данные действительны).

Очистку данных можно разделить на следующие типы:

- конвертация и нормализация данных (согласование форматов данных, например, даты),
- обнаружение одинаковых имен атрибутов, с различными по смыслу значениями;
- стандартизация написания имен (ФИО), представления адресов (Улица", "Ул."), устранение дубликатов;
- замещение кодов значениями (например, почтового индекса наименованием населенного пункта);
- исключение ненужных атрибутов (например, комментариев);
- стандартизация наименований таблиц, индексов и т.д.

Загрузка данных

Основная цель процесса загрузки данных состоит в быстрой загрузке данных в ХД.

Загрузка данных, основанная на использовании команд обновления SQL, является медленной, поэтому загрузка с помощью встроенных в СУБД средств импорта/экспорта является предпочтительной.

При загрузке данных должна быть гарантирована ссылочная целостность данных, а агрегаты должны быть построены и загружены одновременно с подробными данными.

Архитектуры хранилищ данных

Под архитектурой ХД понимают совокупность программно-аппаратных компонент, совокупность технологических и организационных решений, предпринимаемых для создания, разработки и функционирования ХД.

Шесть уровней архитектуры хранилища данных

Документы		Ведение НСИ		Тематическая витрина данных	Сценарный анализ
Унаследованные системы		Ведение метаданных		Региональная витрина данных	Статистический анализ
Транзакционные системы	ETL	Центральное хранилище данных	SRD	Витрина данных подразделения	Многомерный анализ
Файлы		Оперативный склад данных		Прикладная витрина данных	Отчетность
Архивы		Зоны временного хранения		Функциональная витрина данных	Планирование
Источники данных	**Извлечение, преобразование, загрузка**	**Хранение данных**	**Выборка, реструктуризация, доставка**	**Предоставление данных**	**Бизнес-приложения**

Уровень хранения данных

- *НСИ* – нормативно-справочная информация (набор классификаторов, справочников, словарей, стандартов, регламентов, используемых предприятием).
- *Оперативный склад данных* необходим тогда, когда требуется как можно более оперативный доступ к пусть неполным, не до конца согласованным данным, доступным с наименьшей возможной задержкой.
- *Зоны временного хранения* нужны для реализации специфического бизнес-процесса, например, когда перед загрузкой данных контролер данных должен просмотреть их и дать разрешение на их загрузку в хранилище. Для этих зон требуется создание дополнительных средств администрирования, мониторинга, обеспечения безопасности и аудита.

Системы SRD

SRD (Sample, Restructure, Deliver) - выборка, реструктуризация и доставка данных.

SRD выполняет выборку из единого ХД и имеет дело с очищенными данными, структуры которых должны быть приведены в соответствие с требованиями различных приложений.

SRD должно доставить данные в различные витрины в соответствии с правами доступа, графиком доставки и требованиями к составу информации.

Витрины данных

Витрина (киоски) данных (data marts) — срез ХД, представляющий собой массив тематической, узконаправленной информации, ориентированный, например, на пользователей одного департамента.

Витрины данных должны иметь структуры данных, максимально отвечающие потребностям обслуживаемых задач.

Витрины данных следует группировать по территориальным, тематическим, организационным, прикладным, функциональным и иным признакам.

Достоинства витрин данных

- Аналитики видят и работают только с теми данными, которые им реально нужны.
- Для реализации витрин данных не требуется мощная вычислительная техника.
- Относительно небольшой объем хранимых данных, на организацию и поддержку которых не требуется значительных затрат.
- Корпоративная информационная система может эффективно наращиваться за счет добавления новых витрин данных.
- Использование витрин данных позволяет снизить нагрузку на централизованное ХД.

Реляционные хранилища данных

В отличие от OLTP систем РХД проектируются так, чтобы добиться минимального времени выполнения запросов на чтение (у OLTP минимизируется время выполнения запросов на изменение данных).

Типичная структура РХД существенно отличается от структуры обычной реляционной БД. Как правило, эта структура денормализована (это повышает скорость выполнения запросов) и может допускать избыточность данных.

Измерения и факты в РХД

Измерения — это категориальные атрибуты, наименования и свойства объектов, участвующих в некотором бизнес-процессе. Измерения могут быть и числовыми, если какой-либо категории (например, наименованию товара) соответствует числовой код, но в любом случае это данные дискретные, то есть принимающие значения из ограниченного набора. Измерения качественно описывают исследуемый бизнес-процесс.

Факты — это данные, количественно описывающие бизнес-процесс, непрерывные по своему характеру, то есть они могут принимать бесконечное множество значений. Примеры: цена товара, их количество, сумма продаж, зарплата сотрудников и т.д.

В основе технологии РХД лежит принцип, в соответствии с которым измерения хранятся в плоских таблицах так же, как и в обычных реляционных БД, а факты — в отдельных специальных таблицах этой же БД.

Схема построения «Звезда»

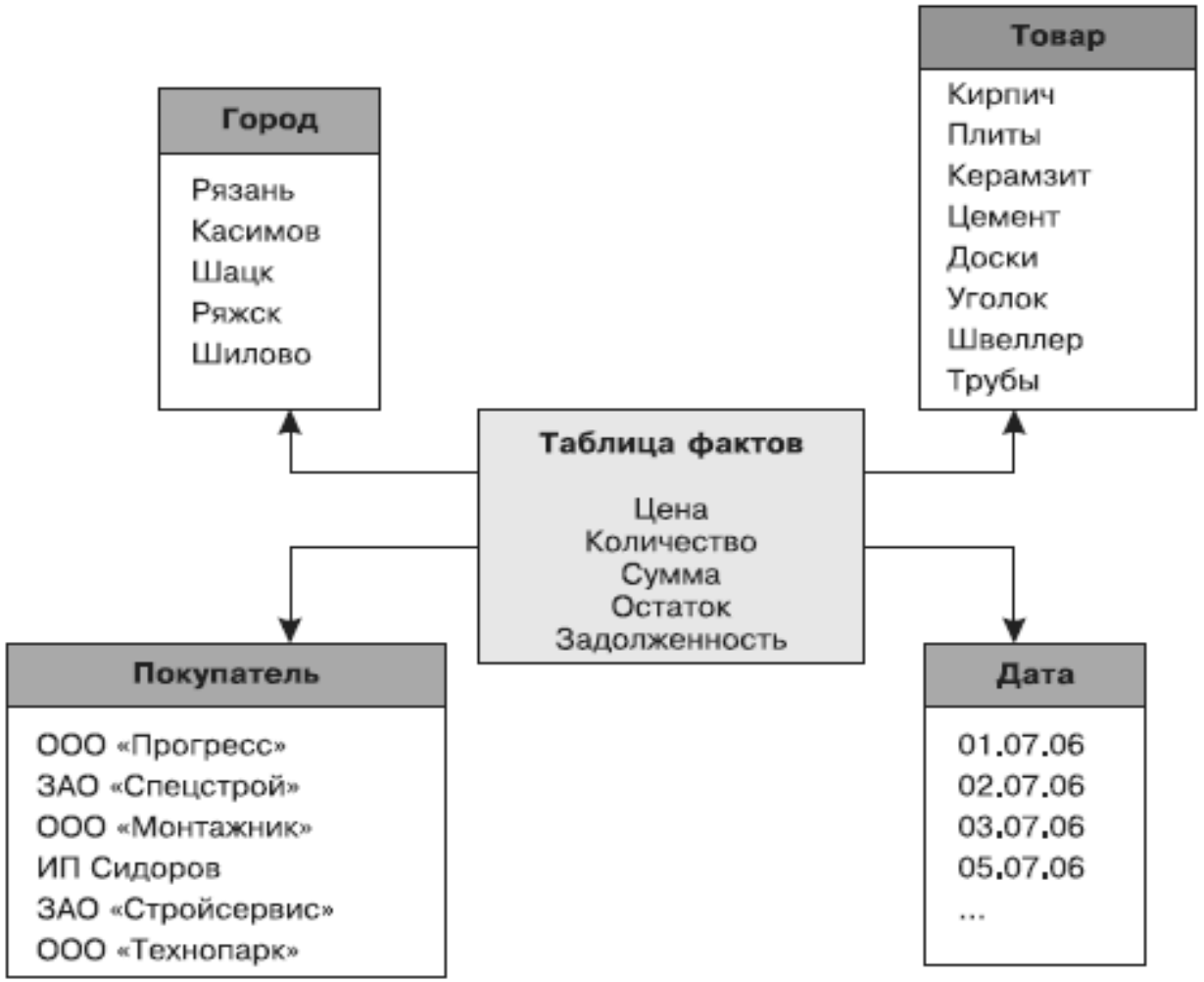


Схема построения «Звезда»

Преимущества схемы «звезда»:

- простота и логическая прозрачность модели;
- простая процедура пополнения измерений, поскольку приходится работать только с одной таблицей.

Недостатки схемы «звезда»:

- наличие иерархий в данных вызовет рост избыточности, замедлит обработку измерений (т.к. одни и те же значения могут встречаться несколько раз в одной и той же таблице) и повысит вероятность возникновения противоречий (например, из-за ошибок ввода).

Схема построения «Снежинка»

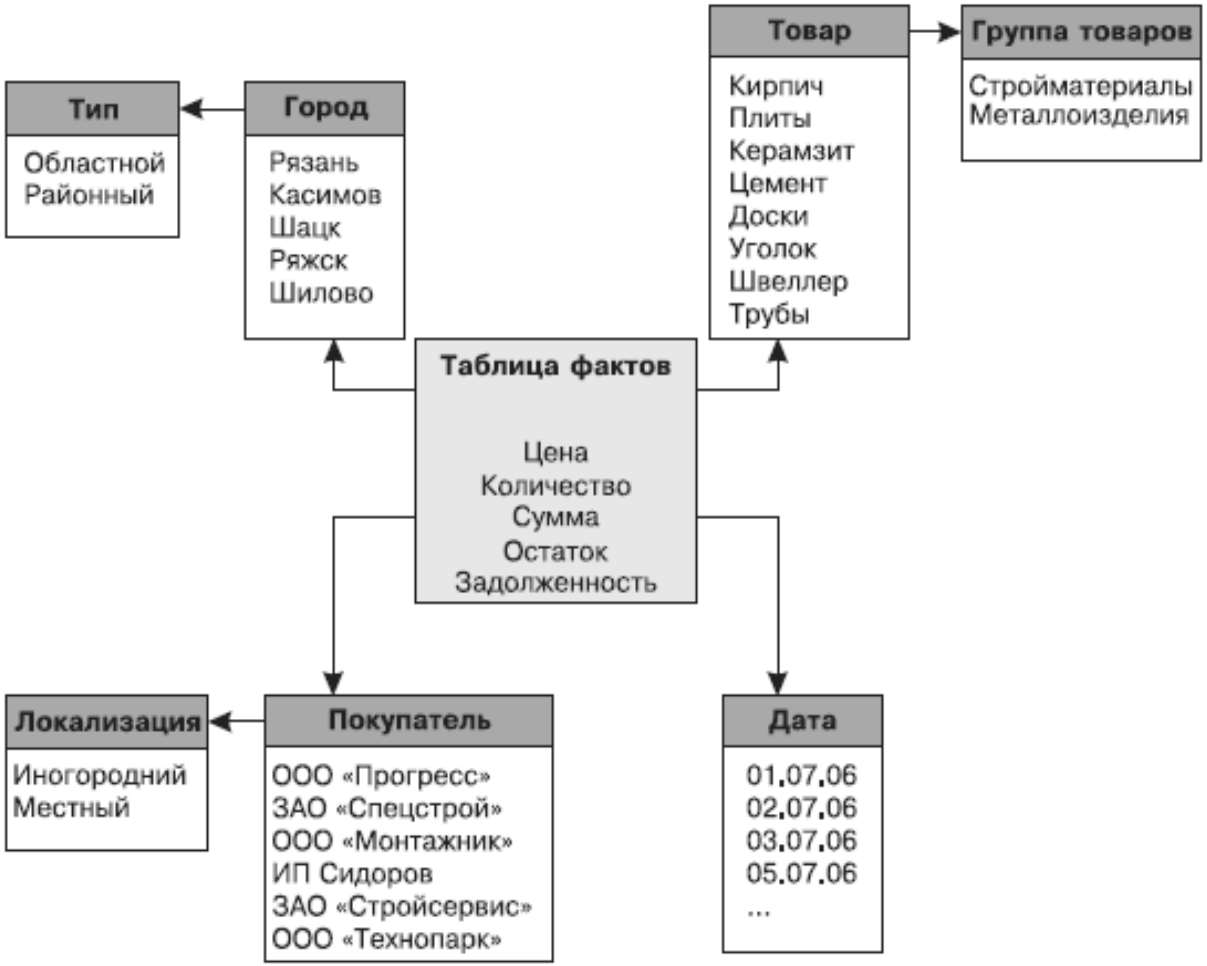


Схема построения «Снежинка»

Преимущества схемы «снежинка»:

- она ближе к представлению данных в многомерной модели;
- процедура загрузки из РХД в многомерные структуры более эффективна и проста, поскольку загрузка производится из отдельных таблиц;
- намного ниже вероятность появления ошибок несоответствия данных;
- большая, по сравнению со схемой «звезда», компактность представления данных, поскольку все значения измерений упоминаются только один раз.

Недостатки схемы «снежинка»:

- достаточно сложная для реализации и понимания структура данных;
- усложненная процедура добавления значений измерений.

Преимущества и недостатки РХД

Основные преимущества:

- практически неограниченный объем хранимых данных;
- поскольку реляционные СУБД лежат в основе построения многих систем OLTP, которые обычно являются главными источниками данных для ХД, использование реляционной модели позволяет упростить процедуру загрузки и интеграции данных в хранилище;
- при добавлении новых измерений данных нет необходимости выполнять сложную физическую реорганизацию хранилища;
- обеспечиваются высокий уровень защиты данных и широкие возможности разграничения прав доступа.

Главный недостаток РХД – невысокая производительность, из-за большого числа таблиц агрегатов.

Таким образом, выбор реляционной модели целесообразен если:

- Значителен объем хранимых данных.
- Иерархия измерений несложная (немного агрегированных данных).
- Требуется частое изменение размерности данных (можно ограничиться добавлением новых таблиц).

Многомерные хранилища данных

Большинство бизнес-процессов описывается множеством атрибутов. Если собрать всю информацию в таблицу, то она окажется сложной для визуального анализа. Более того, она может оказаться избыточной, т.к. в плоской таблице хранятся многомерные данные.

По Э. Кодду **многомерное концептуальное представление** - множественная перспектива, состоящая из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных.

Одновременный анализ по нескольким измерениям определяется как **многомерный анализ**.

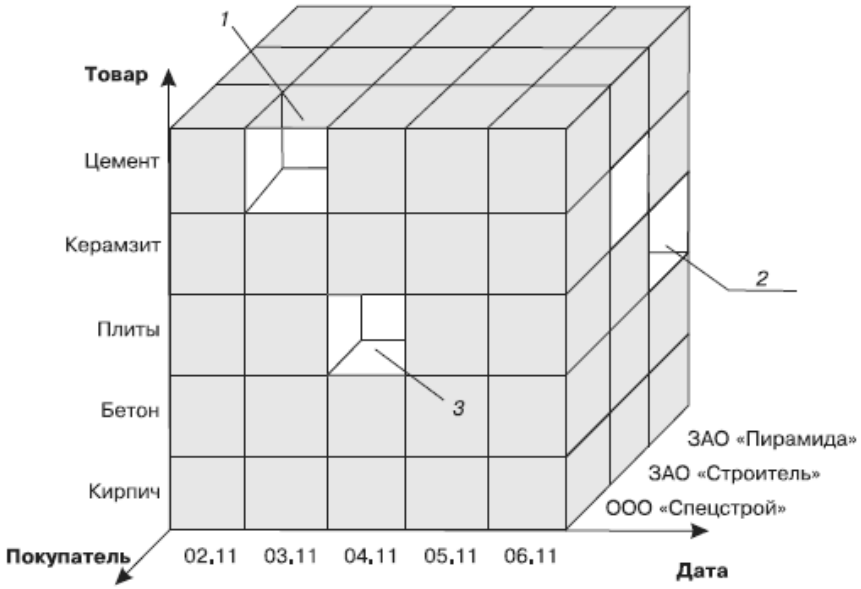
Измерения и факты в МХД

Измерение – это множество объектов одного или нескольких типов, организованных в виде иерархической структуры и обеспечивающих информационный контекст числового показателя. Измерение принято визуализировать в виде ребра (оси) многомерного куба. В качестве одного из измерений используется время.

Факт (мера) - это числовая величина, которая располагается в ячейках гиперкуба. Она количественно характеризует процесс.

Каждому набору значений измерений (например, «дата — товар — покупатель») будет соответствовать ячейка с фактами (мера, measure) связанными с данным набором.

Принцип организации многомерного куба



Преимущества и недостатки МХД

Основные преимущества:

- более наглядная структура, чем совокупность таблиц РХД.
- возможности построения аналитических запросов к системе более широки.
- уменьшение продолжительности поиска, т.к. агрегированные данные вычисляются предварительно и хранятся в многомерных кубах, поэтому тратить время на вычисление агрегатов при выполнении запроса не нужно.

Недостатки:

- требуется больший объем памяти.
- многомерная структура труднее поддается модификации; при необходимости встроить еще одно измерение требуется выполнить физическую перестройку всего многомерного куба.

Таким образом, применение МХД целесообразно в тех случаях, когда объем используемых данных сравнительно невелик, а сама многомерная модель имеет стабильный набор измерений.

Гибридные хранилища данных

Гибридные хранилища сочетают высокую производительность, характерную для многомерной модели, и возможность хранить сколь угодно большие массивы данных, присущую реляционной модели.

Главным принципом построения ГХД является то, что детализированные данные хранятся в РХД, а агрегированные – в МХД.

Если данные, поступающие из OLTP-системы, имеют большой объем (несколько десятков тысяч записей в день и более) и высокую степень детализации, а для анализа используются в основном обобщенные данные, ГХД оказываются наиболее подходящими.

Преимущества:

Построение OLAP-куба выполняется по запросу. Такой подход позволяет избежать взрывного роста данных. При этом можно достичь оптимального времени исполнения клиентских запросов.

Недостатком ГХД является усложнение администрирования из-за более сложного регламента его пополнения, поскольку при этом необходимо согласовывать изменения в реляционной и многомерной структурах.

Управление жизненным циклом информации

Жизненный цикл информации – это изменение ценности информации с течением времени. Например, в заказе на покупку ценность информации меняется с момента размещения заказа до истечения срока гарантии.

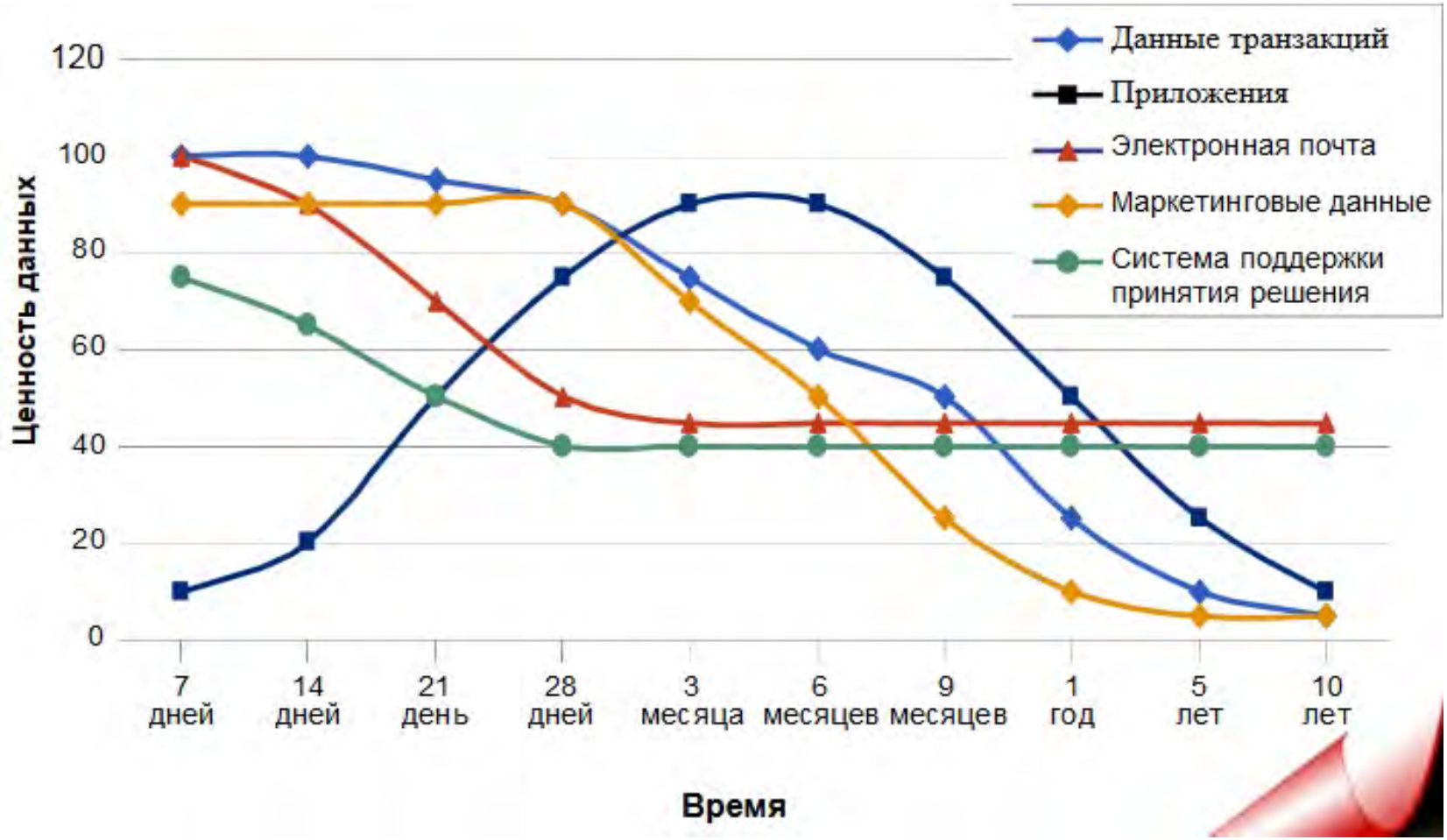
Управление жизненным циклом информации (Information Lifecycle Management – ILM) – это набор политик, процессов, практик, сервисов и инструментов, используемых для того, чтобы соотнести ценность информации с точки зрения бизнеса с наиболее подходящей и эффективной по стоимости инфраструктурой, начиная с момента создания информации и заканчивая ее размещением.

Управление жизненным циклом информации

Проблемы клиента

- В настоящее время расходы на хранение данных составляют более 15% ИТ-бюджетов
- Ежегодно объемы данных растут более чем на 50%
- В большинстве случаев дисковые устройства хранения используются менее чем на 50%, 40% из них являются избыточными
- В мире существуют более 20 тысяч нормативных актов, включающих требования к хранению данных

Управление жизненным циклом информации



Многоуровневое хранение

Многоуровневое хранение – подход к определению различных уровней хранения для снижения затрат на хранение. Каждый уровень имеет различные степени защиты, производительности, частоты доступа к данным и пр. Информация хранится и передается между уровнями, исходя из ее ценности с течением времени.

Пример классификации информации, уровня обслуживания и политики жизненного цикла

Класс данных	Идентификация класса по аттрибутам						Регламент			
	приложение	владелец	файлы	путь	объем	дата создания	дата последнего доступа	скорость доступа	доступность	Политика жизненного цикла
критичные данные для бизнеса	SAPWebAS	*	*	/sap		*	*	15ms	99.99%	
	DB2	db2admin	*.dat	/db2		*	*	15ms	99.99%	
	-	domain\accounting *	*.xls	/home			<6 месяцев	40ms	99%	Перевести в класс важных файлов, если не было доступа в течение 6 месяцев
Важные файлы	-	domain\accounting *	*.xls	/home			>6 месяцев	120ms	99%	
Электронная Почта	LotusDomino	logistics*	*	*	>20MB	<6мес	>30дней			перевести в класс архив почты, если не было доступа в течение 30 дней и размер сообщения больше 20MB
Архив почты								3min	98%	
временые файлы	-	-	*.tmp, *.log, *.dmp, *.~*, tmp*	*		-	>7 дней	-	-	удалить
	-	-	*	/tmp		-	>7 дней	-	-	удалить
дублированные файлы	-	-					>30дней			удалить
ненужные файлы			*.mp3	*						удалить

Оперативный анализ данных OLAP

Концепция OLAP была описана в 1993 году Эдгаром Коддом.

Технология **OLAP** (Online Analytical Processing) представляет собой методику оперативного извлечения нужной информации из больших массивов данных и формирования соответствующих отчетов.

OLAP-технология представляет для анализа данные в виде многомерных наборов данных (многомерный куб, гиперкуб). При этом гиперкуб является концептуальной логической моделью организации данных, а не физической реализацией их хранения, поскольку храниться такие данные могут и в реляционных таблицах.

Для поиска в многомерных структурах данных используется язык запросов MDX (Multidimensional Expressions).

Требования к OLAP-системам

- **основные характеристики**: многомерность модели данных, интуитивные механизмы манипулирования данными, доступность данных, архитектура «клиент-сервер», прозрачность, многопользовательская работа…;
- **специальные характеристики**: обработка ненормализованных данных, хранение результатов отдельно от исходных данных…;
- **характеристики построения отчетов**: гибкое построение отчетов, стабильная производительность при построении отчетов…;
- **управление размерностью**: неограниченное число измерений и уровней агрегирования, неограниченные операции между данными различных измерений...

Тест FASMI

1995 г. Тест FASMI (Fast Analysis of Shared Multidimensional Information – быстрый анализ разделяемой многомерной информации).

- **Fast** (быстрый). OLAP-система должна обеспечивать ответ на запрос пользователя в среднем за пять секунд.
- **Analysis** (аналитический). OLAP-система должна справляться с любым логическим и статистическим анализом, характерным для бизнес-приложений, и обеспечивать сохранение результатов в виде, доступном для конечного пользователя.
- **Shared** (разделяемый). Система должна предоставлять широкие возможности разграничения доступа к данным и одновременной работы многих пользователей.
- **Multidimensional** (многомерный). Система должна обеспечивать концептуально многомерное представление данных, включая полную поддержку множественных иерархий.
- **Information** (информация). Возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Способы хранения данных в OLAP

Гиперкубы строятся на основе исходных и агрегированных данных, которые могут храниться как в реляционных, так и в многомерных базах данных.

Существует три способа хранения данных в OLAP-системах (три архитектуры OLAP-серверов):

- - MOLAP (Multidimensional OLAP) – исходные и агрегатные данные хранятся в многомерной базе данных;
- - ROLAP (Relational OLAP) – исходные данные остаются в той же реляционной базе данных, где они изначально находились, агрегатные же данные помещают в специально созданные для их хранения служебные таблицы в той же базе данных;
- - HOLAP (Hybrid OLAP) – исходные данные остаются в той же реляционной базе данных, где они изначально находились, а агрегатные хранятся в многомерной базе данных.

Работа с измерениями в OLAP

Основные операции:

- сечение (срез);
- транспонирование (вращение);
- свертка (консолидация);
- детализация.

Операция «сечение»

Выделение подмножества ячеек гиперкуба при фиксировании значения одного или нескольких измерений. В результате получается срез или несколько срезов, каждый из которых содержит информацию, связанную со значением измерения, по которому он был построен.

Двумерное представление куба можно получить, "разрезав" его поперек одной или нескольких осей (измерений). В результате фиксации значений измерений получаем двумерную таблицу (**сводная таблица**). При этом набор фактов фактически рассматривается как одно из измерений.

Значения, "откладываемые" вдоль измерений, называются **метками**. Метки используются как для "разрезания" куба, так и для фильтрации выбираемых данных. Значения меток отображаются в двумерном представлении куба как заголовки строк и столбцов.

Примеры сечений

Исходный куб

	США	Канада	Мексика
Январь	20 000	4000	2000
Февраль	30 000	6000	3000
Март	50 000	10 000	5000

Двумерный срез куба для одного факта

	США	Канада	Мексика
Unit Sales	2000	400	200
Store Sales	30 000	6000	3000
Store Cost	10 000	2000	1000

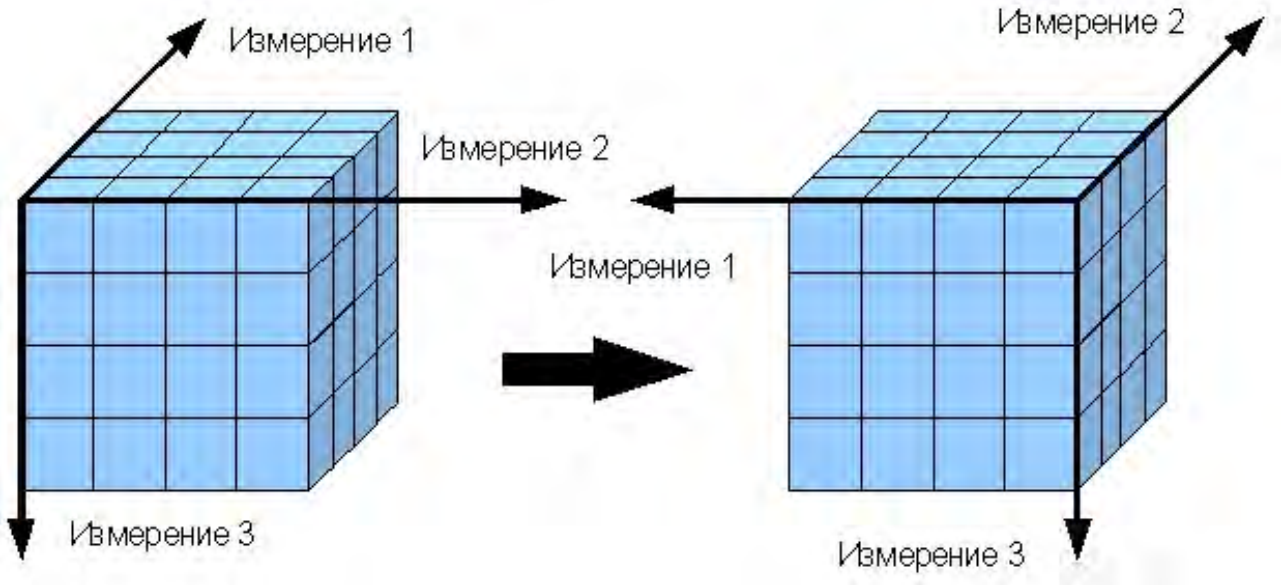
Двумерный срез куба для нескольких фактов

	Январь			Февраль		
	США	Канада	Мексика	США	Канада	Мексика
Unit Sales	500	100	50	500	100	50
Store Sales	7500	1500	750	7500	1500	750
Store Cost	2500	500	250	2500	500	250

Двумерный срез куба с несколькими измерениями на одной оси

Операция «транспонирование»

Транспонирование (вращение) обычно применяется к плоским таблицам, полученным, например, в результате среза, и позволяет изменить порядок представления измерений таким образом, что измерения, отображавшиеся в столбцах, будут отображаться в строках, и наоборот.

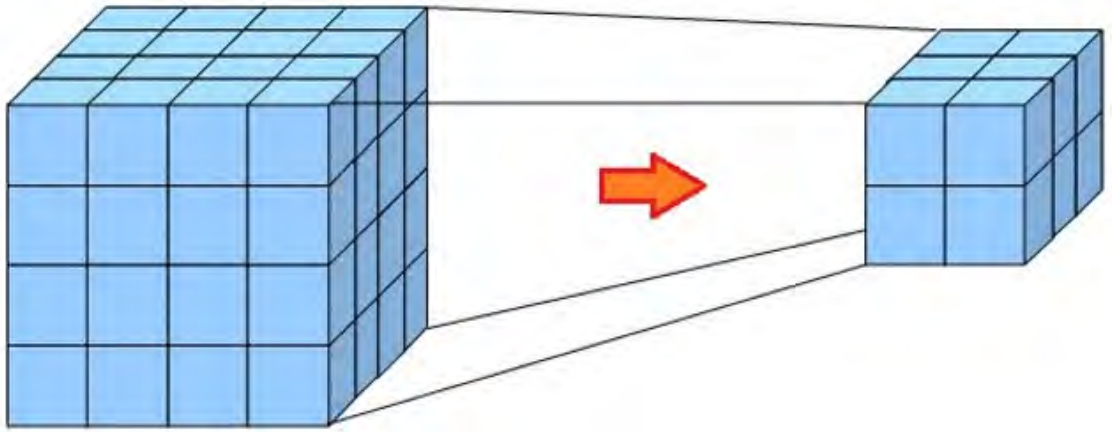


Операция «свёртка»

Каждое измерение включает направления консолидации данных, состоящие из серии последовательных уровней обобщения (**уровней иерархии**), где каждый вышестоящий уровень соответствует большей степени агрегации данных по соответствующему измерению. В этом случае становится возможным произвольный выбор желаемого уровня детализации информации по каждому из измерений.

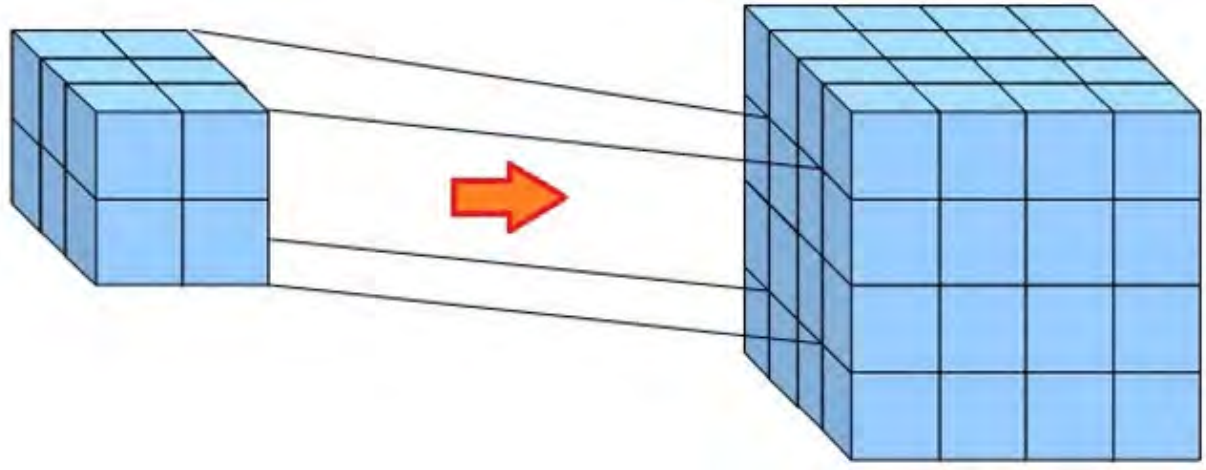
При свертке одно или несколько подчиненных значений измерений заменяются теми значениями, которым они подчинены. При этом уровень обобщения данных уменьшается.

В соответствии с уровнями иерархии вычисляются агрегатные значения.



Операция «детализация»

Детализация — это процедура, обратная свертке; уровень обобщения данных уменьшается. При этом значения измерений более высокого иерархического уровня заменяются одним или несколькими значениями более низкого уровня, например, вместо наименований групп товаров отображаются наименования отдельных товаров.



Понятие Data Mining

Понятие Data Mining (добыча данных), появилось в 1978 году.

Data Mining - это процесс поддержки принятия решений, основанный на поиске в «сырых» больших объемах данных скрытых (неочевидных), объективных и полезных на практике закономерностей, необходимых для принятия решений.

- **Неочевидных** - найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем.
- **Объективных** - найденные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным.
- **Практически полезных** - выводы имеют конкретное значение, которому можно найти практическое применение.
- **Знания** - совокупность сведений, которая образует целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т.д.

Применения Data Mining

- **Банки:** Кредитование, Прогнозирование остатка на счетах клиентов.
- **Маркетинг:** Сегментация клиентов, формирование адресных предложений (таргетинг), RTB-реклама, повышение лояльности
- **Мерчендайзинг, улучшения дизайна и удобства использования продукта.**
- **Исследования для правительства:** предсказание поведения населения (анализ соц. сетей), иностранцев (таможня); кибербезопасность, отслеживание пассажироперевозок и трафика (автомобильные номера, авиабилеты), предсказание стихийных бедствий.
- **Производство:** минимизация времени «простоя» производства, оптимизация производственной цепочки.
- **Медицина:** медицинская диагностика, предсказание эпидемий, назначение лекарств.

Yandex Data Factory: поиск, машинный перевод, фильтрация спама, рекламный таргетинг, рекомендации, распознавание образов и речи, предсказание пробок

Применения Data Mining

Web Mining

- ***Web Content Mining*** - автоматический поиск и извлечение информации из разнообразных источников Интернета, перегруженных "информационным шумом".
- ***Web Usage Mining*** - обнаружение закономерностей в действиях пользователя Web-узла или их группы. В результате система может рекомендовать ему определенные наборы товаров или услуг.

Text Mining

Разработанный на основе статистического и лингвистического анализа предназначен для выполнения семантического (смыслового) анализа неструктурированных текстов.

Social Mining

Типовые задачи: анализ информационных потоков, персонификация предложений, поиск аномалий, компьютерных ботов и мошенников

Call Mining

Объединяет в себя распознавание речи, ее анализ и Data Mining. Цель - упрощение поиска в аудио-архивах, содержащих записи переговоров между операторами и клиентами. Два подхода - на основе преобразования речи в текст и на базе фонетического анализа.

Этапы процесса Data Mining

1) Анализ предметной области

Предметная область – это мысленно ограниченная область реальной действительности, подлежащая описанию или моделированию и исследованию. Она состоит из объектов, различаемых по свойствам и находящихся в определенных отношениях между собой.

В процессе изучения предметной области должна быть создана ее модель. Модель предметной области описывает процессы, происходящие в предметной области, и данные, которые в этих процессах используются.

Этапы процесса Data Mining

2) Постановка задачи

- формулировка задачи;
- формализация задачи.

Постановка задачи включает также описание статического и динамического поведения исследуемых объектов.

Этапы процесса Data Mining

3) Подготовка данных

- Определение и анализ требований к данным
- Сбор данных
- Предобработка данных

Данные высокого качества – это полные, точные, своевременные данные, которые поддаются интерпретации. Данные низкого качества называют «**грязные**» данные.

«Грязные» данные и их очистка

«Грязные» данные – это отсутствующие, неточные или бесполезные данные с точки зрения практического применения. Они могут появиться из-за ошибок при вводе данных, использовании иных форматов представления или единиц измерения, отсутствия своевременного или неудачного обновления, и т.д.

Первый этап очистки данных происходит еще в системе ETL. Однако специальные средства очистки могут справиться не со всеми видами грязных данных.

Наличие «грязных» данных не обязательно означает необходимость их очистки или же предотвращения появления.

Наиболее распространенные виды «грязных» данных:

- **пропущенные значения;**
- **дубликаты данных;**
- **шумы и выбросы.**

«Грязные» данные и их очистка

Пропущенные значения.
Причины:
- данные вообще не были собраны (например, при анкетировании скрыт возраст);
- некоторые атрибуты могут быть неприменимы для некоторых объектов (например, атрибут «годовой доход» неприменим к ребенку).

Решение:
- исключить объекты с пропущенными значениями из обработки;
- рассчитать новые значения для пропущенных данных;
- игнорировать пропущенные значения в процессе анализа;
- заменить пропущенные значения на возможные значения.

Дублирование данных.
Дубликатами называются записи с одинаковыми значениями всех атрибутов.
Причина: результаты ошибок при подготовке данных.
Решение:
- удалить всю группу записей, содержащая дубликаты (если наличие дубликатов вызывает недоверие к информации, полностью ее обесценивает).
- заменить группу дубликатов на одну уникальную запись.

Шумы и выбросы.
Шумы – сильные отклонения от среднего значения в наборе данных. Шум в данных не несет никакой полезной скрытой информации, поэтому его стараются минимизировать.
Выбросы – резко отличающиеся объекты или наблюдения в наборе данных.

Задача аналитика – не только их обнаружить, но и оценить степень их влияния на результаты дальнейшего анализа. Если выбросы являются информативной частью набора данных, то проводится анализ с выбросами и с их отсутствием, и сравнение полученных результатов.

Этапы процесса Data Mining

4) Построение модели

Построение моделей Data Mining осуществляется с целью исследования или изучения моделируемого объекта, процесса, явления и получения новых знаний, необходимых для принятия решений.

Классификация типов моделей в зависимости от характерных свойств, присущих изучаемому объекту или системе:

- динамические (изменяющиеся во времени) и статические;
- стохастические и детерминированные;
- непрерывные и дискретные;
- линейные и нелинейные;
- статистические; экспертные; модели, основанные на методах Data Mining;
- Предикативные (прогностические) и дескриптивные (описательные).

Этапы процесса Data Mining

5) Проверка и оценка моделей

Проверка достоверности или адекватности модели. Она заключается в определении степени соответствия модели реальности. Адекватность модели проверяется путем тестирования.

6) Применение модели

Модель используется применительно к новым данным с целью решения поставленных задач.

7) Коррекция и обновление модели

По прошествии определенного промежутка времени модель следует проанализировать, определить, действительно ли она эффективна.

При появлении новых данных требуется повторное обучение модели. Этот процесс называют *обновлением модели.*

Классификация задач Data Mining

Согласно *классификации по стратегиям*, задачи Data Mining подразделяются на:

- обучение с учителем;
- обучение без учителя.

«Обучение с учителем» – система принудительно обучается с помощью примеров «стимул-реакция». Между входами и эталонными выходами может существовать некоторая зависимость, но она не известна. Известна только конечная совокупность прецедентов – **обучающая выборка**. Чтобы проверить способность модели к обобщению, всю обучающую выборку разделяют на два множества – обучающее и тестовое.

- **Обучающее множество** включает данные, использующиеся для конструирования модели. Оно содержит входные и выходные значения примеров. На основе этих данных требуется построить алгоритм, способный для любого объекта выдать достаточно точный ответ.
- **Тестовое множество** также содержит входные и выходные значения примеров. Выходные значения используются для проверки работоспособности модели.

«Обучение без учителя» – система спонтанно обучается выполнять поставленную задачу, без вмешательства со стороны экспериментатора. Как правило, это пригодно только для задач, в которых известны описания множества объектов (обучающего множества), и требуется обнаружить внутренние взаимосвязи, существующие между объектами.

Классификация задач Data Mining

Ошибка обучения – ошибка, допущенная моделью на обучающем множестве. На каждой итерации обучения для непрерывной входной переменной она рассчитывается как среднеквадратическая ошибка:

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - y_d^{(i)}\right)^2,$$

где N – число обучающих примеров,

$y^{(i)}$ – значение на выходе для i-го примера,

$y_d^{(i)}$ – целевое значение.

Ошибка обобщения – это ошибка, полученная на тестовых примерах, т.е. вычисляемая по тем же формулам, но для тестового множества.

Классификация данных

Классификация – системное распределение изучаемых предметов, процессов по каким-либо существенным признакам для удобства их исследования.

Сегодня используют для:

- Спам-фильтры
- Определение языка
- Поиск похожих документов
- Анализ тональности
- Распознавание рукописных букв и цифр
- Определение подозрительных транзакций

Популярные алгоритмы: Наивный Байес, Деревья Решений, Логистическая Регрессия, К-ближайших соседей, Машины Опорных Векторов

Классификация данных

Классификация – системное распределение изучаемых предметов, процессов по каким-либо существенным признакам для удобства их исследования.

Бинарная классификация - зависимая переменная может принимать только два значения (например, потенциальный покупатель или нет).

Другой вариант классификации возникает, если зависимая переменная может принимать значения из некоторого множества предопределенных классов. Например, когда необходимо предсказать, какую марку автомобиля захочет купить клиент. В этих случаях рассматривается множество классов для зависимой переменной.

Классификация может быть **одномерной** (по одному признаку) и **многомерной** (по двум и более признакам).

Оценка точности классификации может проводиться при помощи **кросс-проверки**, при которой точность классификации тестового множества сравнивается с точностью классификации обучающего множества.

Пример классификации

Пример классификации

База данных о клиентах турагентства с информацией о возрасте и доходе за месяц.

Есть рекламный материал двух видов: более дорогой и комфортный отдых и более дешевый, молодежный отдых. Соответственно, определены два класса клиентов: класс 1 и класс 2.

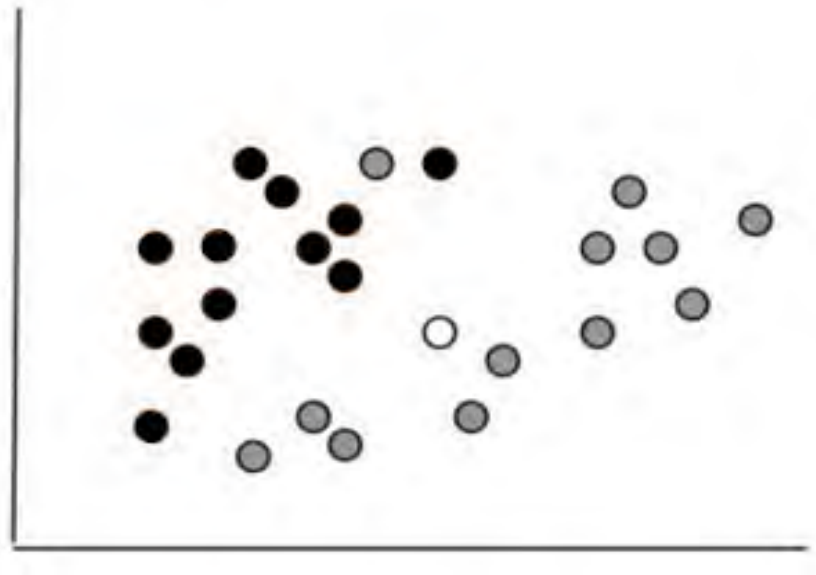
Необходимо определить, к какому классу принадлежит новый клиент, и какой из двух видов рекламных материалов ему стоит отсылать.

Исходные данные для классификации

Код клиента	Возраст	Доход	Класс
1	18	25	1
2	22	100	1
3	30	70	1
4	32	120	1
5	24	15	2
6	25	22	1
7	32	50	2
8	19	45	2
9	22	75	1
10	40	90	2

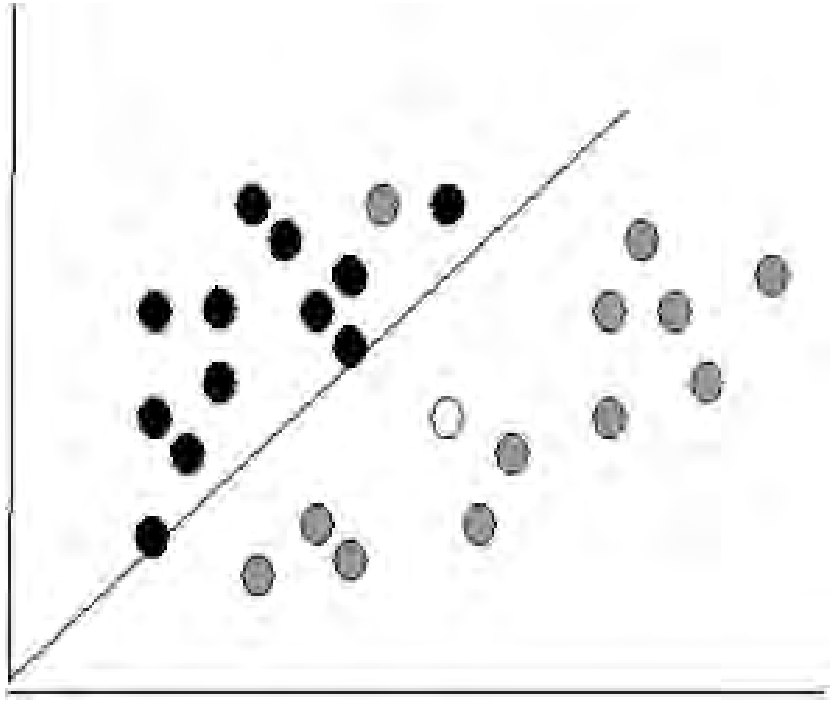
Множество объектов базы данных в двухмерном измерении

Для наглядности представим базу данных в двухмерном измерении (возраст и доход), в виде множества объектов, принадлежащих классам 1 (черная метка) и 2 (серая метка).



Решение задачи будет состоять в том, чтобы определить, к какому классу относится новый клиент, на рисунке обозначенный белой меткой.

Решение задачи классификации методом линейной регрессии



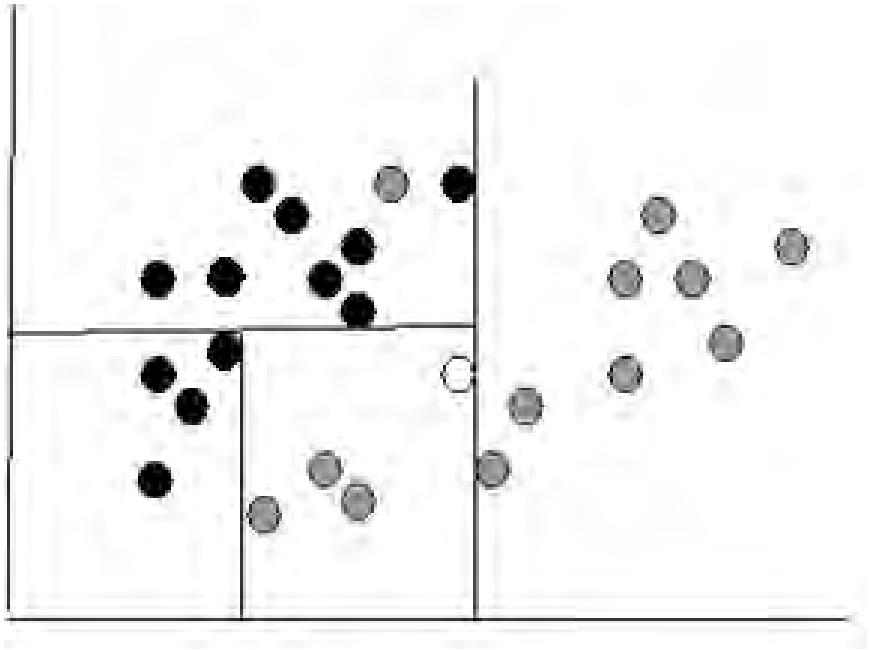
Решение задачи классификации методом деревьев решений

ЕСЛИ X > 5 ТО grey

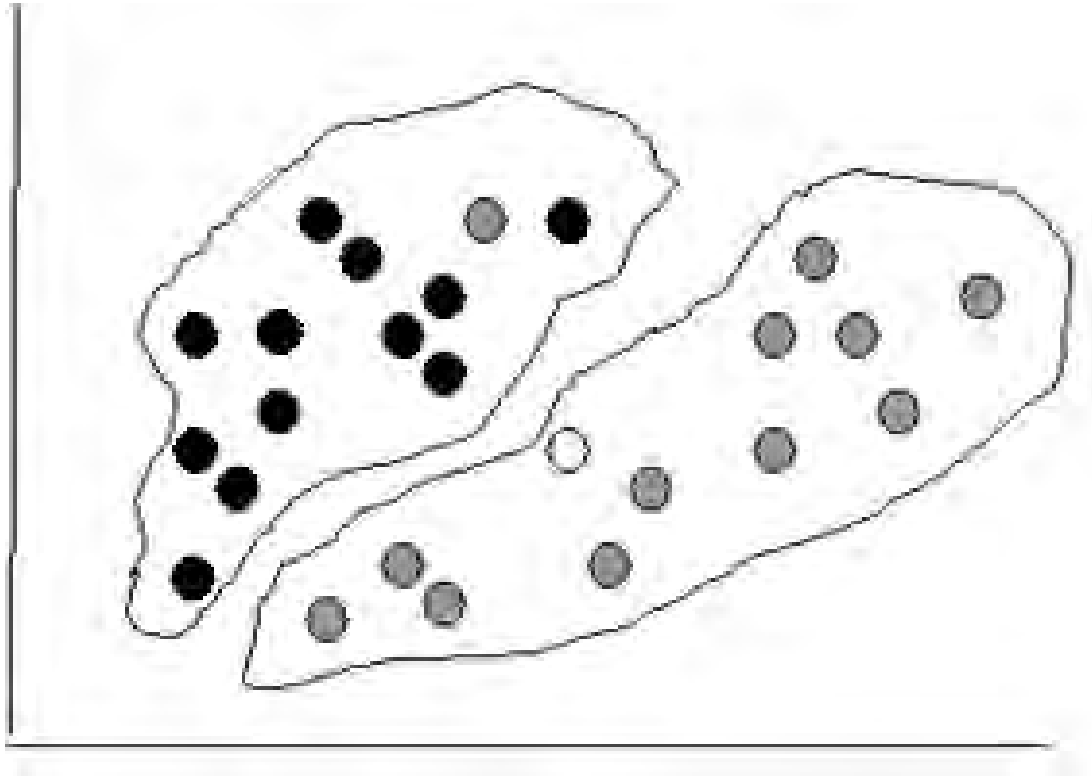
ИНАЧЕ ЕСЛИ Y > 3 ТО black

ИНАЧЕ ЕСЛИ X > 2 ТО grey

ИНАЧЕ black



Решение задачи классификации методом нейронных сетей



Кластеризация данных

Кластеризация предназначена для разбиения совокупности объектов на однородные группы (кластеры), имеющие общие свойства (сегментация).

Сегодня используют для:

- Сегментация рынка (типов покупателей, лояльности)
- Объединение близких точек на карте
- Сжатие изображений
- Анализ и разметки новых данных
- Детекторы аномального поведения

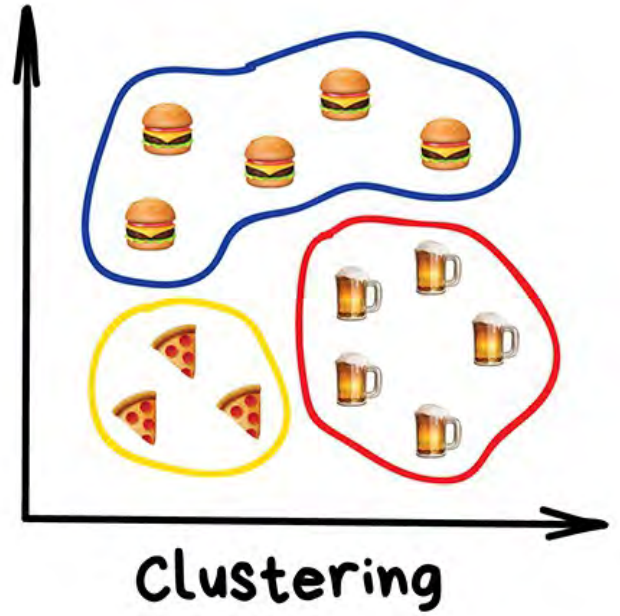
Популярные алгоритмы: Метод K-средних, Mean-Shift, DBSCAN

Кластеризация данных

В отличие от задачи классификации здесь классы объектов изначально не предопределены.

Кластеризация относится к стратегии «обучения без учителя».

В результате применения различных методов кластеризации могут быть получены неодинаковые результаты.



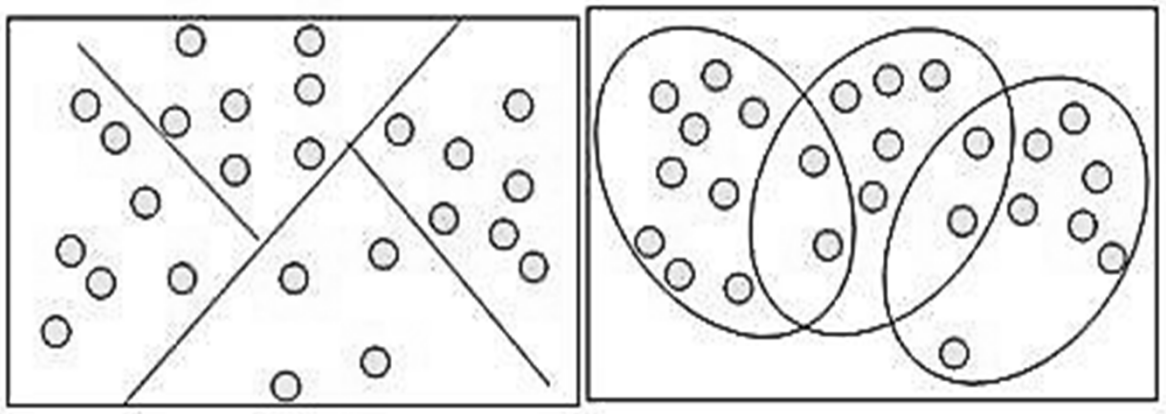
Понятие кластера

Кластер имеет следующие математические характеристики: центр, радиус, среднеквадратическое отклонение, размер кластера.

Центр кластера – это среднее геометрическое место точек в пространстве переменных.

Радиус кластера – максимальное расстояние точек от центра кластера.

Кластеры могут быть непересекающимися и пересекающимися.



В случае пересекающихся кластеров невозможно однозначно отнести объект к одному из двух кластеров. Такие объекты называют **спорными**. Объект относится к кластеру, если расстояние от объекта до центра кластера меньше радиуса кластера. Если это условие выполняется для двух и более кластеров, объект является спорным.

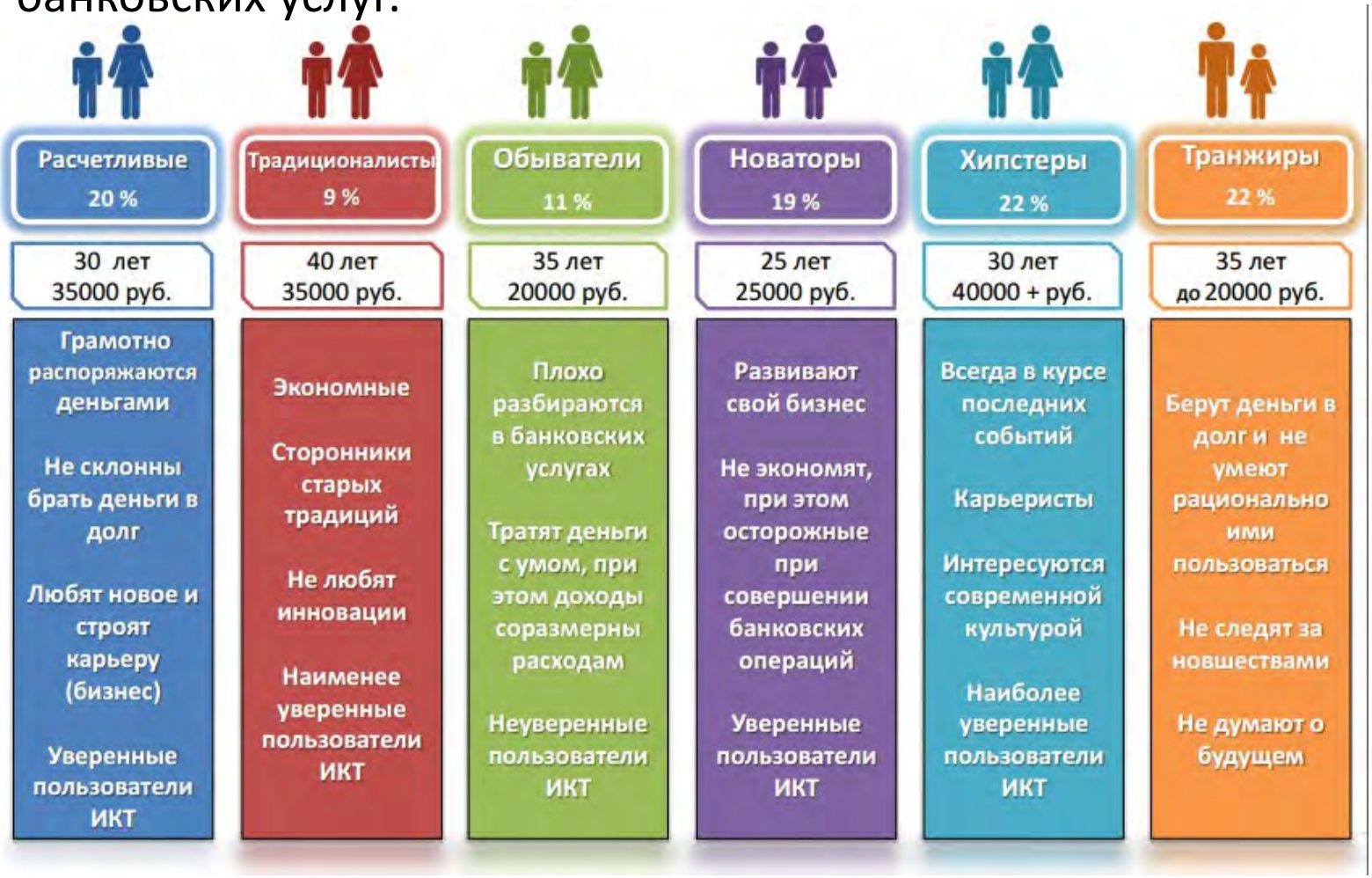
Понятие кластера

Наиболее распространенный способ определения меры расстояния между кластерами (меры близости) – вычисление евклидова расстояния между двумя точками *i* и *j* на плоскости, когда известны их координаты *X* и *Y*.

$$D_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}$$

Пример кластеризации

Анализ статистики использования клиентами различных банковских услуг.



Прогнозирование данных

Прогнозирование направлено на определение тенденций динамики конкретного объекта или события на основе ретроспективных данных.

Различие задач классификации и прогнозирования состоит в том, что в первой задаче предсказывается класс зависимой переменной, а во второй – числовые значения зависимой переменной (интерполяция или экстраполяция).

Основой для прогнозирования являются временные ряды. **Временной ряд** – последовательность наблюдаемых значений какого-либо признака, упорядоченных в неслучайные моменты времени.

Прогноз может быть краткосрочным (не более чем на 3% от объема наблюдений), среднесрочным (на 3-5%) и долгосрочным (более чем на 5%).

Для построения краткосрочных и среднесрочных прогнозов вполне подходят статистические методы.

Основные понятия прогнозирования

- **Тренд** – неслучайная функция, которая формируется под действием тенденций, влияющих на временной ряд (например, фактор роста рынка).
- **Циклическая составляющая** является периодически повторяющейся компонентой временного ряда (важно при определении количества используемых ретроспективных данных).
- **Период прогнозирования** – основная единица времени, на которую делается прогноз. Например, если необходимо узнать доход компании через месяц, то период прогнозирования – месяц.
- **Горизонт прогнозирования** – это число периодов в будущем, которые покрывает прогноз. Например, если необходимо узнать прогноз на 12 месяцев вперед, с данными по каждому месяцу, то горизонт прогнозирования – 12 месяцев. С увеличением горизонта точность прогноза снижается.
- **Интервал прогнозирования** – частота, с которой делается новый прогноз. Может совпадать с периодом прогнозирования. При длительном интервале возникает риск не идентифицировать изменения, произошедшие в процессе, при коротком – возрастают издержки на прогнозирование.

Точность прогноза характеризуется **ошибкой прогноза** (обычно среднеквадратическая ошибка).

Поиск ассоциативных правил

Ассоциация – высокая вероятность связи событий друг с другом. В ходе решения задачи поиска ассоциативных правил отыскиваются закономерности между связанными событиями в наборе данных.

Примеры:

- определение товаров, которые стоит продвигать совместно, выбор местоположения товара;
- перекрестные продажи: если есть информация о том, что клиенты приобрели продукты А, Б и В, то какие из них вероятнее всего купят продукт Г?
- определение последовательностей покупок клиентов: какая покупка последует за покупкой товара А?

Популярные алгоритмы: Apriori, Euclat, FP-growth

Поиск ассоциативных правил (основные понятия)

- **Транзакция** – множество событий, которые произошли одновременно.
- **Поддержка правила** – количество или процент транзакций, содержащих определенный набор данных.
- **Достоверность правила** показывает, какова вероятность того, что из события A следует событие B.

Например, есть правило: «3% покупателей, приобретающих «хлеб», приобретают и «молоко» с вероятностью 75%». 75% – это достоверность правила, 3% – это поддержка правила.

Если значение поддержки слишком велико, то в результате работы алгоритма будут найдены очевидные правила. Слишком низкое значение поддержки приведет к нахождению очень большого количества правил, которые будут в большей части необоснованными, но неизвестными для аналитика.

Если уровень достоверности слишком мал, то ценность правила вызывает серьезные сомнения.

Основные методы Data Mining

- Корреляционно-регрессионный анализ
- Деревья решений
- Нейронные сети
- Метод k-means
- Алгоритм Apriori

Корреляционно-регрессионный анализ

Применяется для исследования интенсивности, вида и формы зависимостей, является методическим инструментарием при решении задач прогнозирования.

Корреляционный анализ – количественный метод определения тесноты и направления взаимосвязи между выборочными переменными величинами.

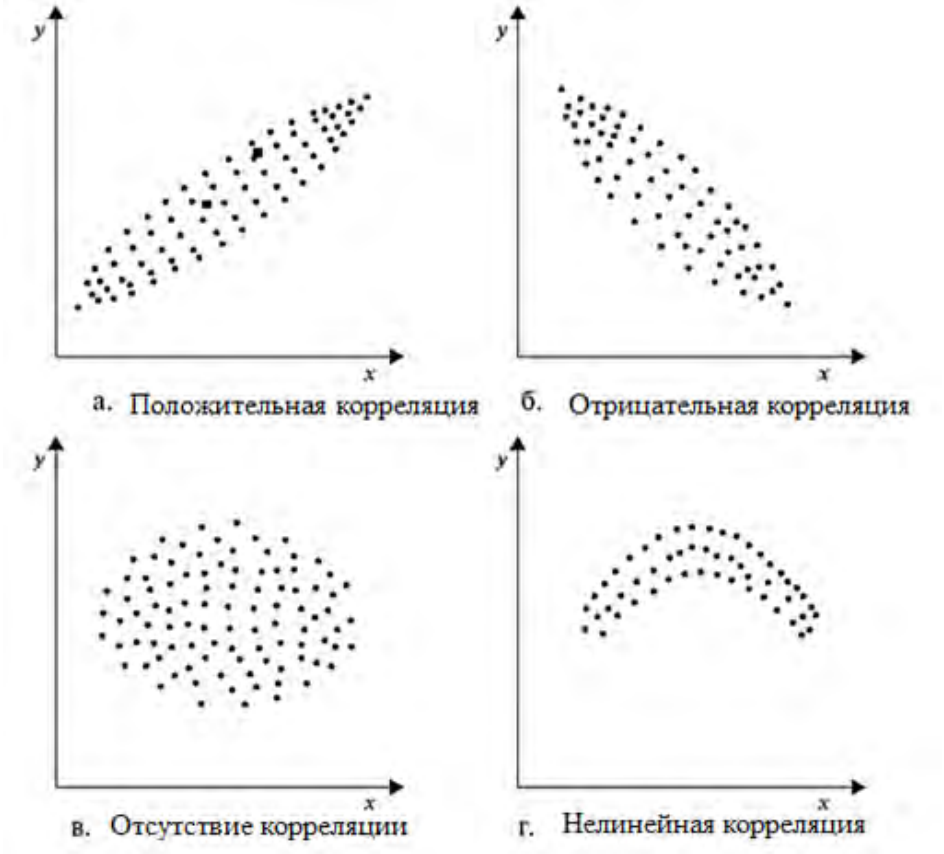
Применяется для оценки зависимости выходных полей данных от входных факторов и устранения незначащих факторов.

Критерием принятия решения об исключении является **порог значимости**. Если корреляция между входным и выходным факторами меньше порога значимости, то соответствующий фактор отбрасывается как незначащий.

Коэффициент корреляции Пирсона - безразмерный индекс в интервале от -1,0 до 1,0 включительно, отражает степень линейной зависимости между двумя множествами данных.

Корреляционный анализ

Для графического представления связи двух переменных используется система координат с осями, соответствующими переменным *x* и *y*. Построенный график, называемый **диаграммой рассеивания**. Обычно, значения независимого параметра откладывается по горизонтальной оси, а значения зависимого – по вертикальной.



Регрессионный анализ

Количественный метод определения вида математической функции в причинно-следственной зависимости между переменными величинами.

Задачи:

1) Установление формы зависимости.

Относительно формы зависимостей выделяют:

- линейная регрессия – выражается линейной функцией (связь между количеством тренировок и количеством правильно решаемых задач в сессии);
- нелинейная регрессия – выражается нелинейной функцией (связь между уровнем мотивации и эффективностью выполнения задачи).

Относительно числа переменных выделяют:

- парная регрессия – регрессия между двумя переменными;
- множественная регрессия – регрессия между зависимой переменной и несколькими факторами.

В зависимости от характера регрессии различают:

- положительную регрессию. С увеличением (уменьшением) объясняющей переменной значения зависимой переменной также соответственно увеличиваются (уменьшаются);
- отрицательную регрессию. С увеличением или уменьшением объясняющей переменной зависимая переменная уменьшается или увеличивается.

Регрессионный анализ

2) Определение функции регрессии.

Задача сводится к выяснению действия на зависимую переменную главных факторов, при неизменных прочих равных условиях, и при условии исключения воздействия на зависимую переменную случайных элементов. Функция регрессии определяется в виде математического уравнения того или иного типа.

3) Оценка неизвестных значений зависимой переменной.

Решение этой задачи сводится к решению задачи одного из типов:

- оценка значений зависимой переменной внутри рассматриваемого интервала исходных данных, т.е. пропущенных значений; при этом решается задача **интерполяции**;

- оценка будущих значений зависимой переменной, т.е. нахождение значений вне заданного интервала исходных данных; при этом решается задача **экстраполяции**.

Регрессионный анализ

Модель линейной регрессии является часто используемой. Уравнение линейной регрессии: $Y=b_0+b_1*x_1+ b_2*x_2+…+ b_n*x_n$,
где Y – выходная (зависимая) переменная модели;
$x_1, x_2,…x_n$ – входные (независимые) переменные;
b_0 – константа (свободный член);
b_i – коэффициенты линейной регрессии (параметры модели).
Коэффициент регрессии показывает, на сколько (в абсолютном выражении) изменяется значение результативного признака при изменении факторного признака на единицу.
Задача линейной регрессии заключается в подборе коэффициентов b_i таким образом, чтобы на заданный входной вектор $X=(x_1, x_2…x_n)$ регрессионная модель формировала желаемое выходное значение Y.
Вычисляемая с помощью **метода наименьших квадратов** линия называется **линией регрессии**. Она характеризуется тем, что сумма квадратов расстояний от точек на диаграмме до этой линии минимальна.

Пример линейной регрессии

Пример. Зависимость между количеством работников и объемом производства.

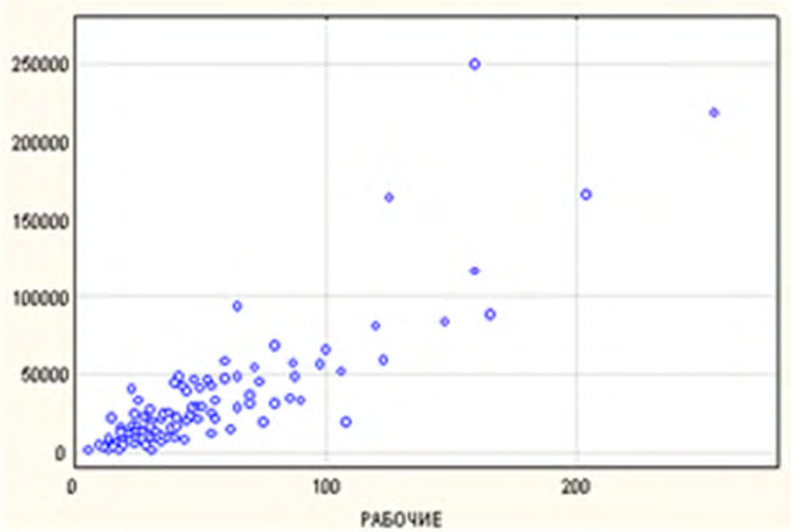
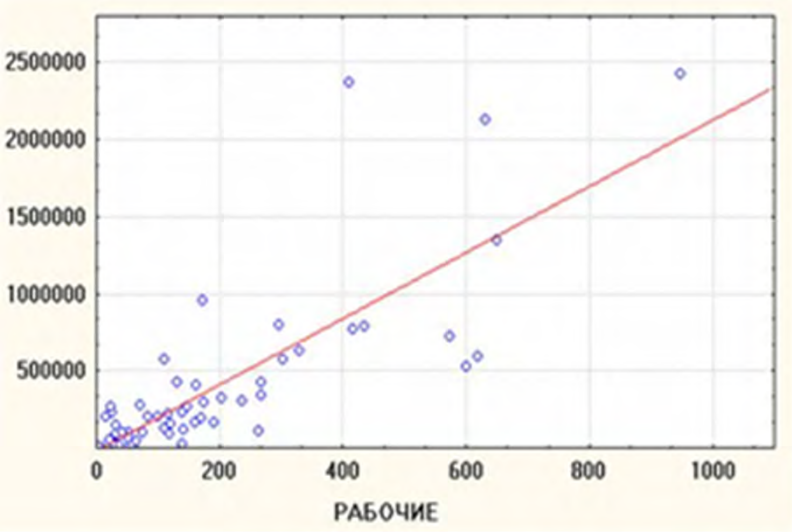


Диаграмма рассеяния

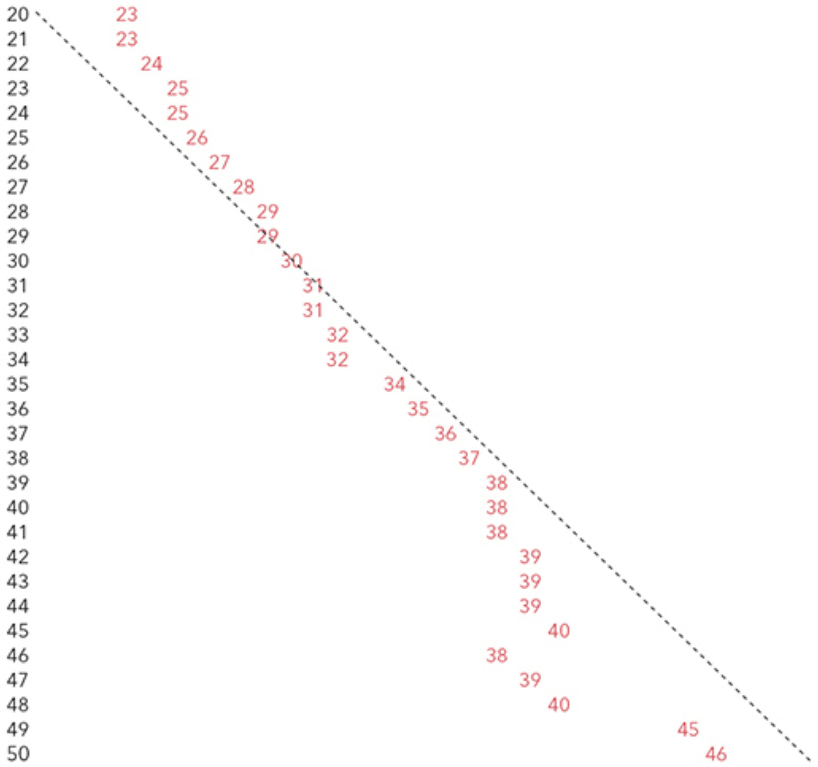


Линия регрессии

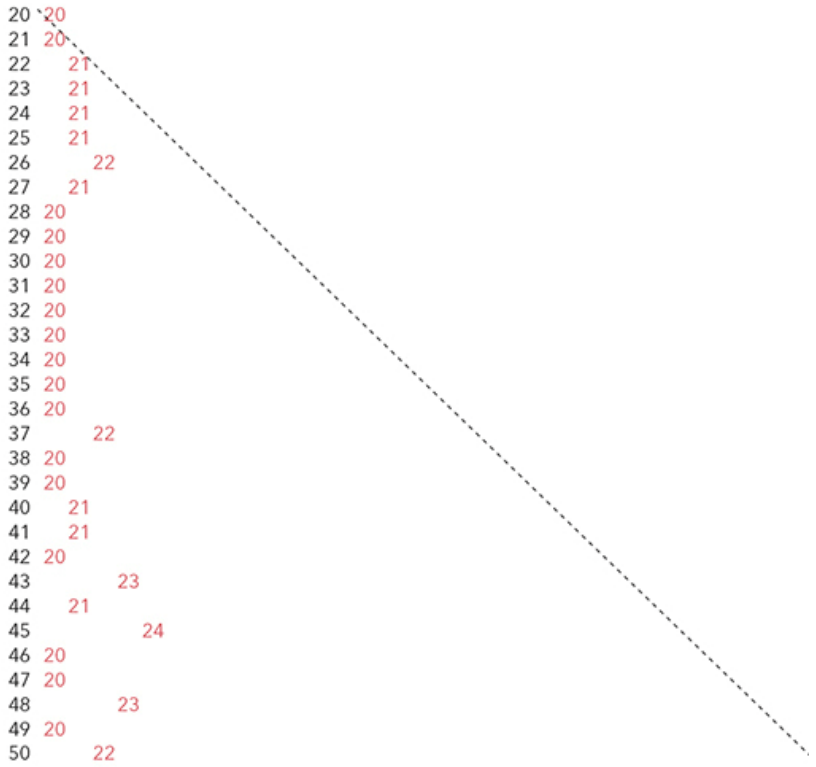
В большинстве случаев наблюдается определенный разброс наблюдений относительно линии регрессии. **Остаток** – это отклонение отдельной точки (наблюдения) от линии регрессии (предсказанного значения).

Пример линейной регрессии

a woman's age *vs. the age of the men who look best to her*



a man's age *vs. the age of the women who look best to him*

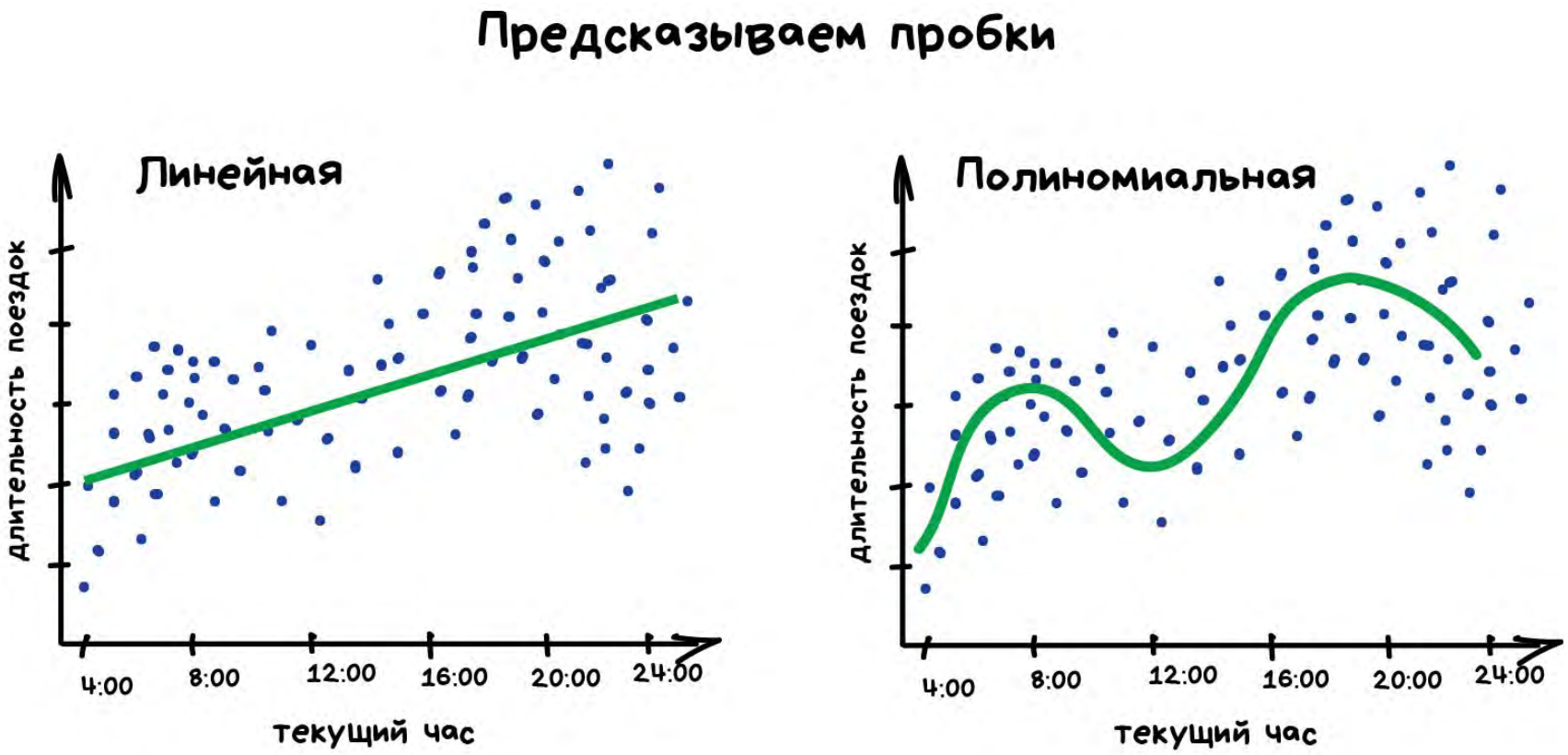


Нелинейная регрессия

Часто встречаются полиномиальная, параболическая, гиперболическая, степенная, показательная и экспоненциальная зависимости.

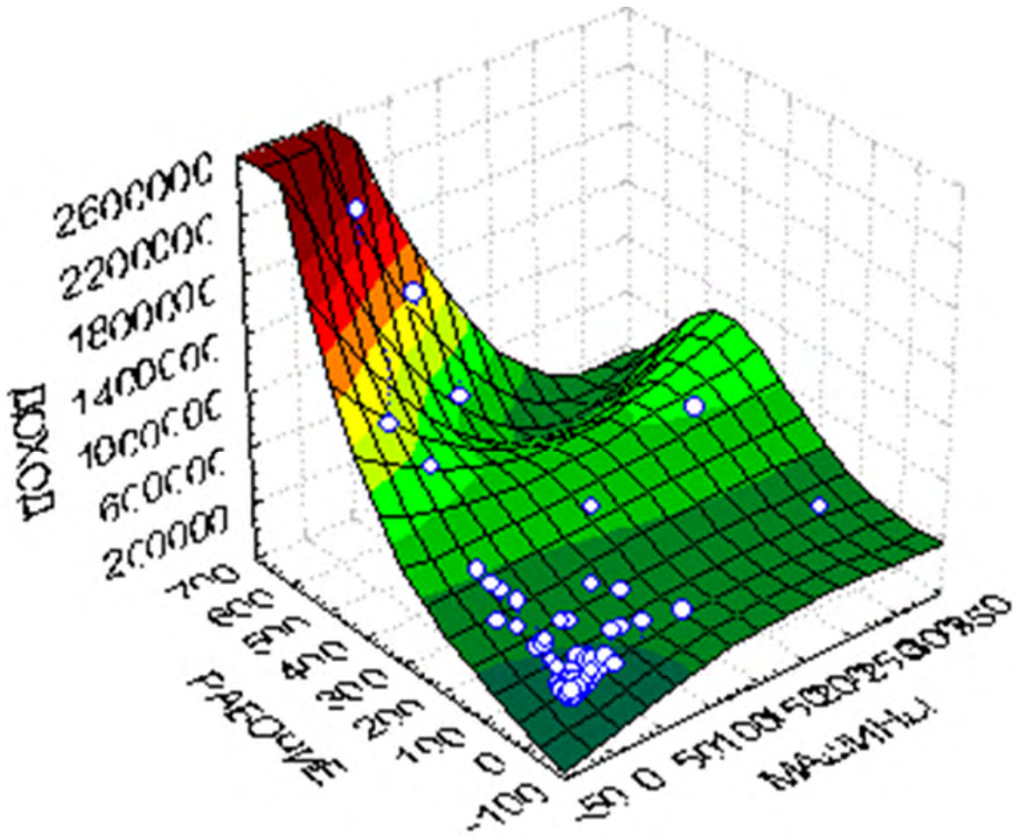
Название	Уравнение
парабола	$y = \alpha + \beta x + \gamma x^2 + u$
гипербола	$y = \alpha + \frac{\beta}{x} + u$
показательная	$y = \alpha \cdot \beta^x \cdot u$
степенная	$y = \alpha \cdot x^{\beta} \cdot u$
экспоненциальная	$y = e^{\alpha + \beta x} \cdot u$

Пример линейной регрессии



Пример нелинейной регрессии

Пример множественной нелинейной регрессии: На объем производства влияют несколько факторов (например, количество работников и энерговооруженность).



Деревья решений

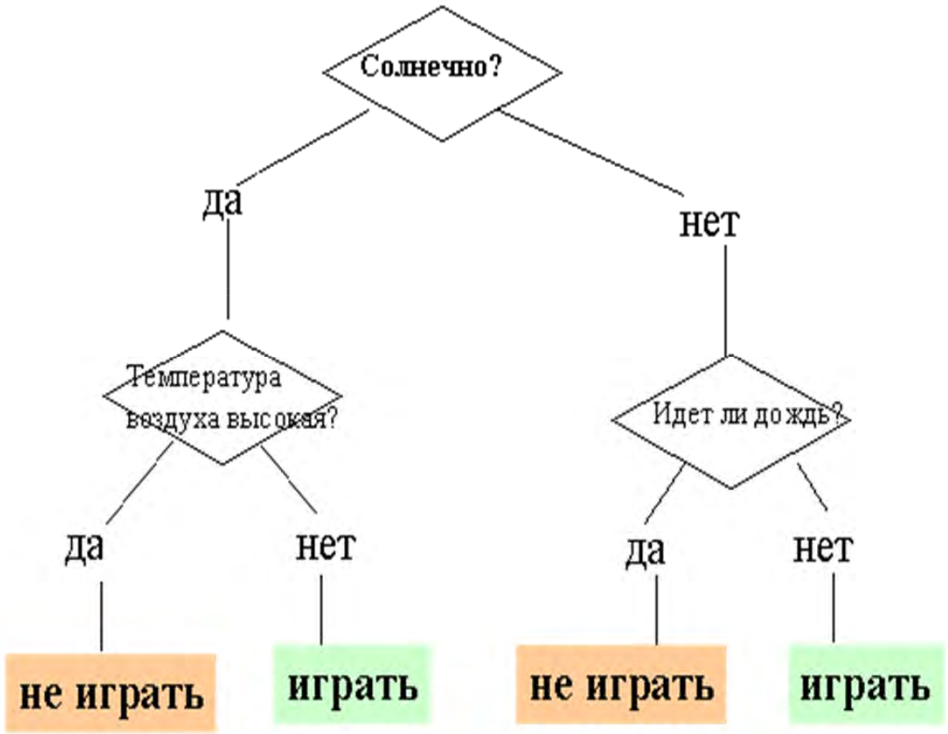
Метод является одним из наиболее популярных методов решения задач классификации и прогнозирования.

Дерево решений представляет собой иерархическую структуру, базирующуюся на наборе вопросов, подразумевающих ответ «да» или «нет».

Если целевая переменная принимает дискретные значения, решается задача классификации. Если же зависимая переменная принимает непрерывные значения, то решается задачу численного прогнозирования.

Пример дерева решений

Пример: «Играть ли в гольф?». Для решения требуется ответить на ряд вопросов, которые находятся в узлах дерева, начиная с его корня.



Пример дерева решений

Основные понятия деревьев решений

- **Атрибуты ветвления** – признаки, описывающие классифицируемые объекты, по которым будет производиться проверка правила («температура воздуха»).
- **Узлы** – содержат правила, с помощью которых производится проверка атрибутов, и множество объектов в данном узле разбивается на подмножества.
- **Листья** – конечные узлы дерева, в которых содержатся подмножества, ассоциированные с классами («Играть», «Не играть»).
- **Корень дерева (корневой узел**) – начальный (входной) узел дерева («Солнечно?»)
- **Внутренний узел** (**узел проверки)**: «Температура воздуха высокая?», «Идет ли дождь?»
- **Ветвь дерева** (случаи ответа): «Да», «Нет».
- **Алгоритм построения дерева решений** – метод, в соответствии с которым осуществляется выбор атрибута ветвления на каждом шаге.

Деревья решений

Бинарные деревья являются самым простым случаем деревьев решений. В остальных случаях, ответов и, соответственно, ветвей дерева, выходящих из его внутреннего узла, может быть больше двух.

Достоинства деревьев решений:

- Просты в понимании и интерпретации.
- Не требуют подготовки данных.
- Быстрый процесс обучения.
- Используется модель «белого ящика». Если определенная ситуация наблюдается в модели, то ее можно объяснить при помощи булевой логики.
- Метод является надежным.
- Позволяют работать с большим объемом информации без специальных подготовительных процедур.

Качество деревьев решений

Качество классификационной модели, построенной при помощи дерева решений, характеризуется двумя основными признаками: точностью распознавания и ошибкой.

- **Точность распознавания** рассчитывается как отношение объектов, правильно классифицированных в процессе обучения, к общему количеству объектов набора данных, которые принимали участие в обучении.
- **Ошибка** рассчитывается как отношение объектов, неправильно классифицированных в процессе обучения, к общему количеству объектов набора данных, которые принимали участие в обучении.

Качество деревьев решений

В результате построения могут быть созданы слишком сложные конструкции (большая глубина дерева). Такие «ветвистые» деревья очень трудно понять. Ценность правила, справедливого для 2-3 объектов, крайне низка. Лучше иметь дерево, состоящее из малого количества узлов, которым бы соответствовало большое количество объектов из обучающей выборки.

Остановка – такой момент в процессе построения дерева, когда следует прекратить дальнейшие ветвления.

Решением проблемы слишком ветвистого дерева является его сокращение путем **отсечения** некоторых ветвей.

Отсечение ветвей или замену некоторых ветвей поддеревом следует проводить там, где эта процедура не приводит к возрастанию ошибки. Процесс проходит снизу вверх, т.е. является восходящим. Это более популярная процедура, чем использование правил остановки.

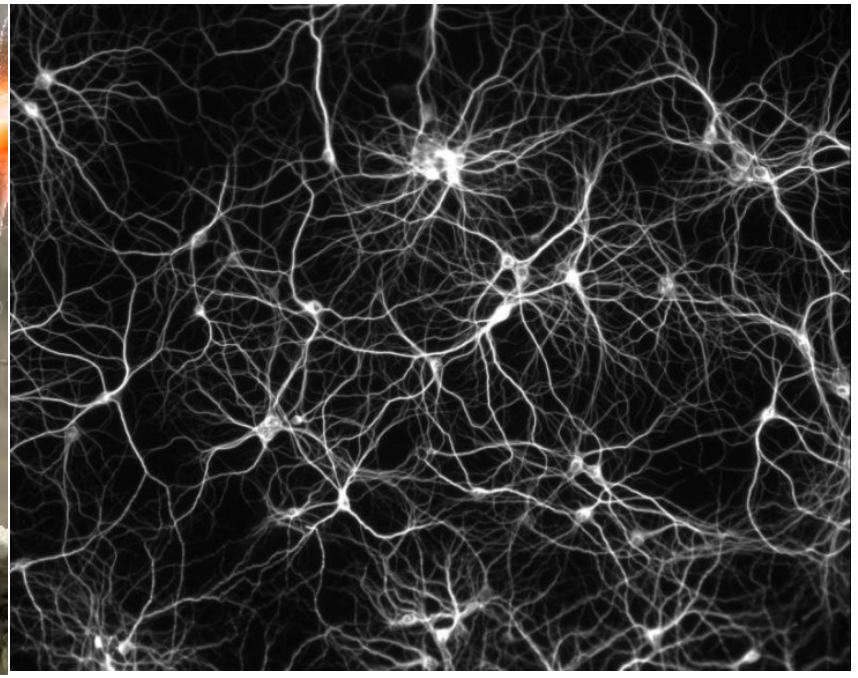
Нейронные сети

Искусственные нейронные сети – это модели биологических нейронных сетей мозга, в которых нейроны имитируются относительно простыми элементами (искусственными нейронами).

Идея нейронных сетей основана на аналогии с функционированием нервной ткани и заключается в том, что исходные параметры рассматриваются как сигналы, преобразующиеся в соответствии с имеющимися связями между «нейронами», а в качестве ответа (результата анализа) рассматривается отклик всей сети на исходные данные.

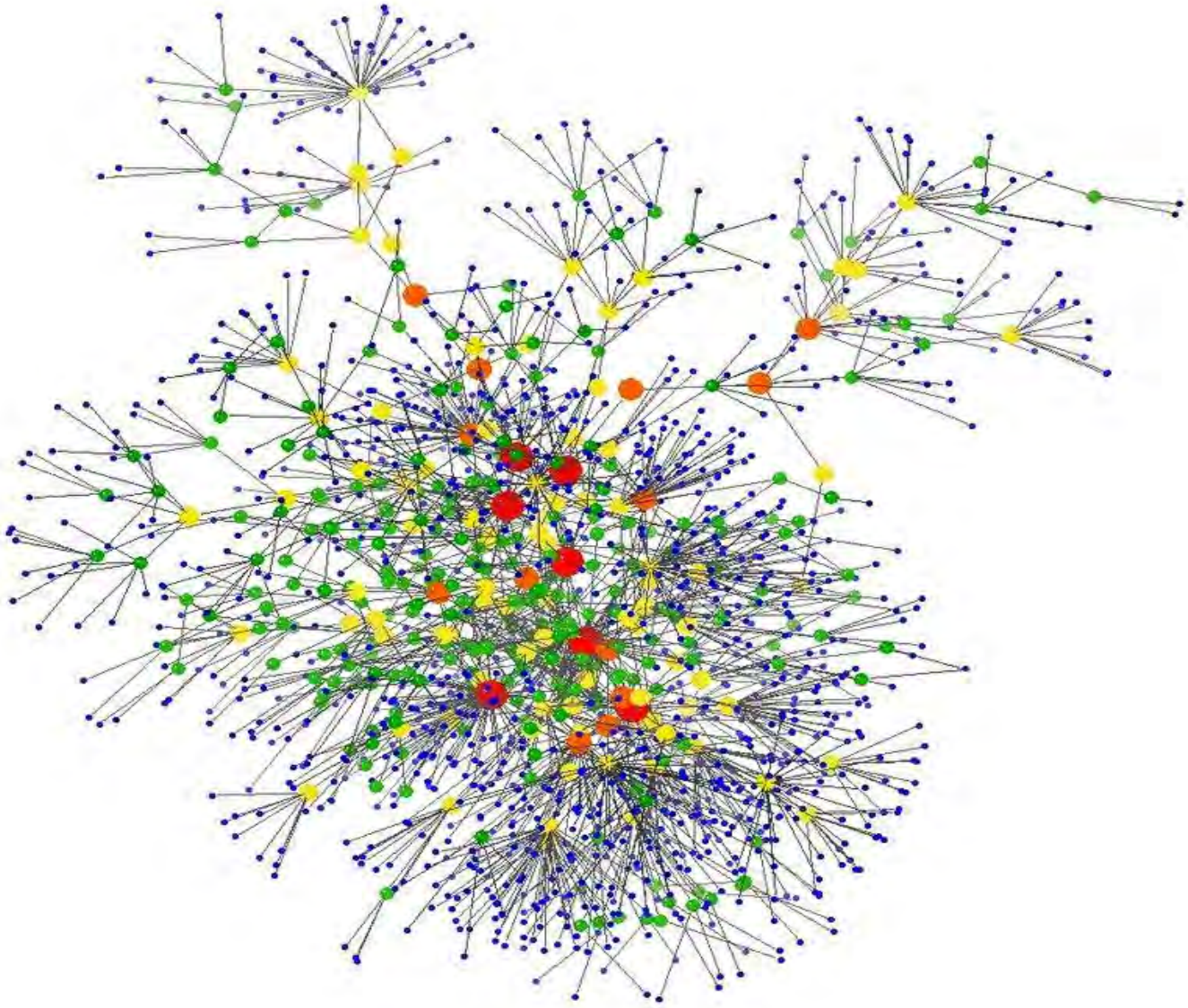
Среди задач Data Mining, решаемых с помощью нейронных сетей, можно выделить классификацию, прогнозирование и кластеризацию.

Нейронные сети



В среднем наш мозг насчитывает около 86 млрд нейронов.

Искусственная нейронная сеть

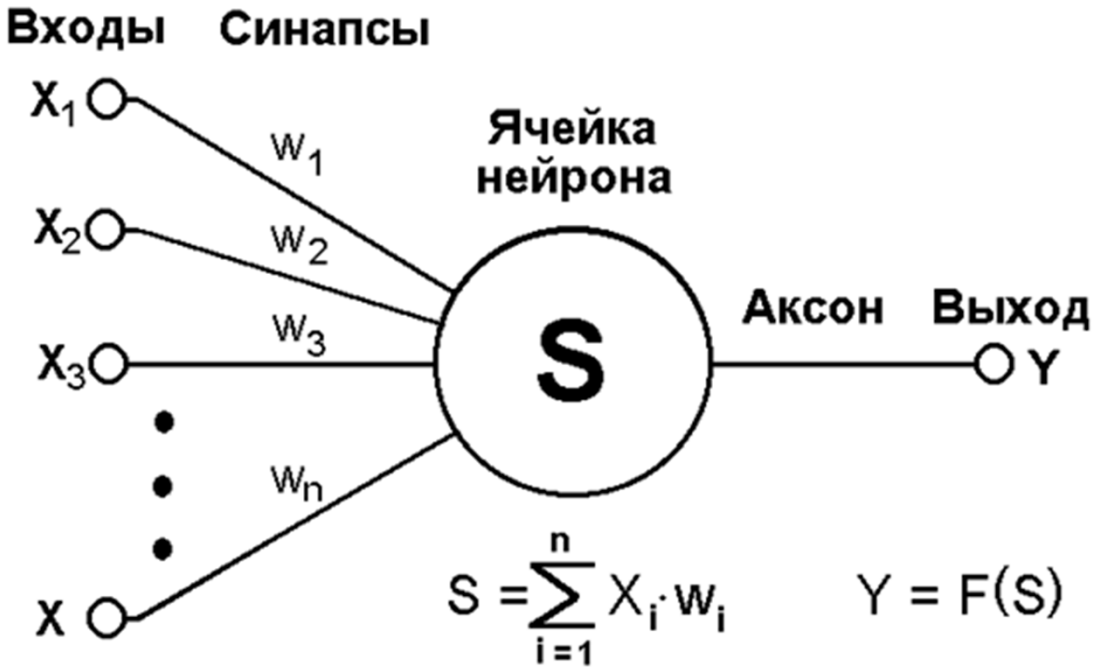


Примеры использования нейронных сетей

1) Удаление лиц прохожих с Google Street View.
2) Рисование картин – Ostagram, Prisma, Deepart.io
3) Распознавание изображений – Algorithmia, Quickdraw.withgoogle.
4) Написание стихов – Яндекс Автопоэт.
5) Интеллектуальные игры – Google AlphaGo.
6) Поиск по фотографии – FindFace.
7) Видеочаты – MSQRD, Snapchat
8) Распознавание музыка – Shazam, SoundHound
9) Голосовые помощники – Алиса, Siri

Искусственный нейрон

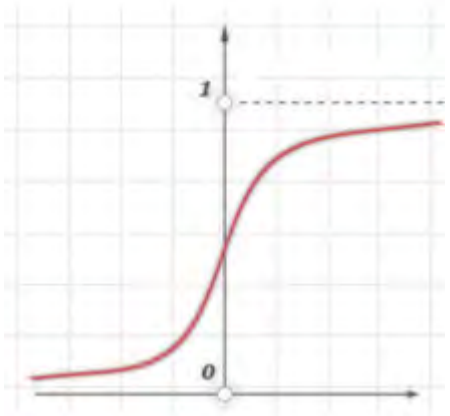
Искусственный нейрон – элемент искусственных нейронных сетей, моделирующий некоторые функции биологического нейрона.
Его главная функция – формировать выходной сигнал в зависимости от сигналов, поступающих на его входы.



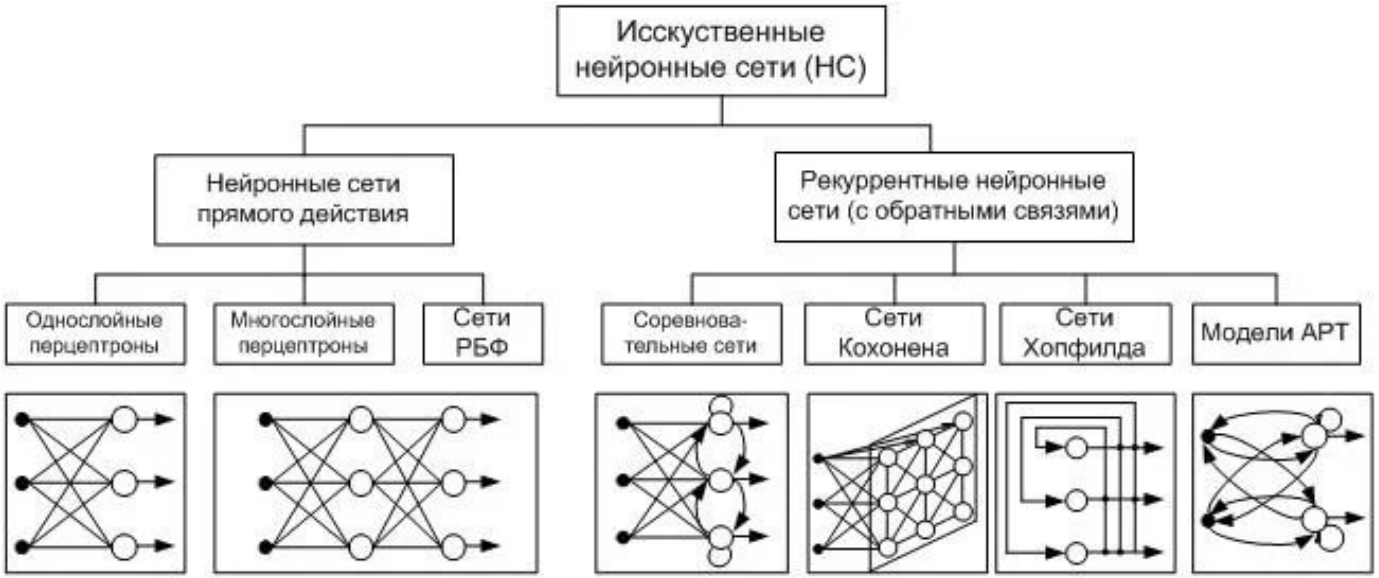
Элементы нейронных сетей

- **Синапсы** – однонаправленные входные связи, соединенные с выходами других нейронов.
  Каждый синапс характеризуется величиной **синаптической связи** (ее весом w_i). Текущее состояние нейрона определяется как взвешенная сумма его входов.
- **Аксон** – выходная связь данного нейрона, с которой сигнал поступает на синапсы следующих нейронов.
- **Точка ветвления (выход)** – это элемент нейрона, посылающий его выходной сигнал по нескольким адресам и имеющий один вход и несколько выходов. На вход точки ветвления обычно подается выходной сигнал нелинейного преобразователя, который затем посылается на входы других нейронов.
- **Функция активации** – функция, вычисляющая выходной сигнал искусственного нейрона. Обычно сигмоидальная (логистическая) функция является наиболее распространенной.

$$f(s) = \frac{1}{1 + e^{-s}}.$$



Классификация нейронных сетей



Сети прямого распространения являются статическими, т.е. на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети.

Рекуррентные сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети.

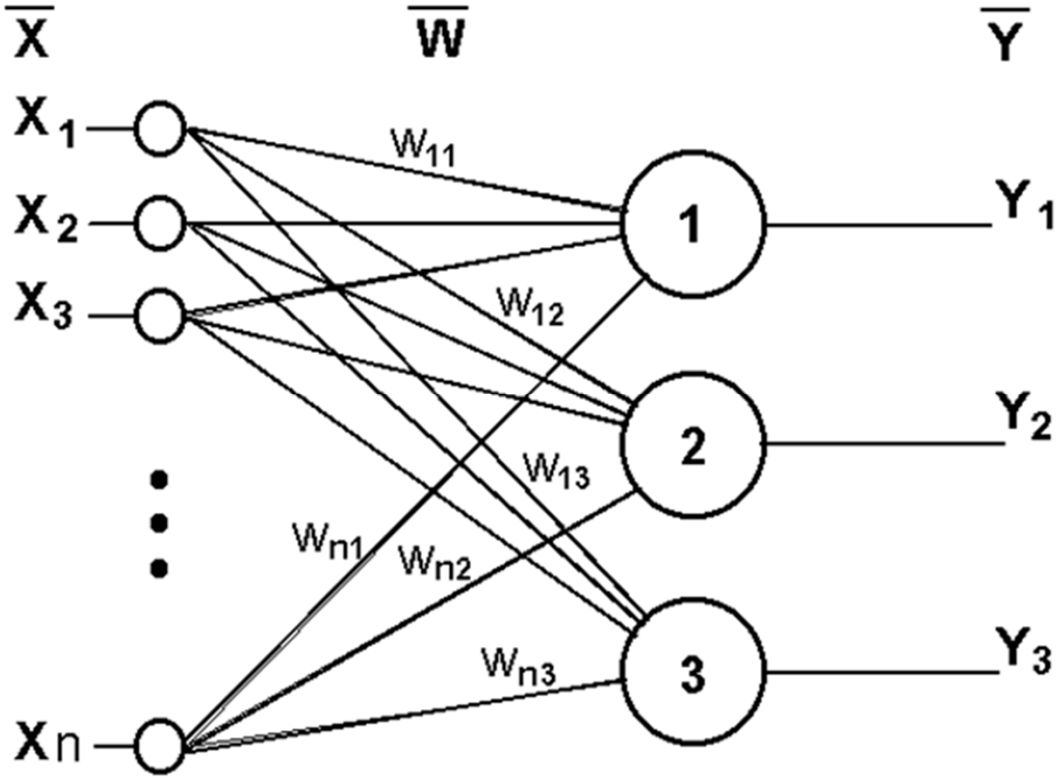
Слоистые сети

Слой – один или несколько нейронов, на входы которых подается один и тот же общий сигнал.

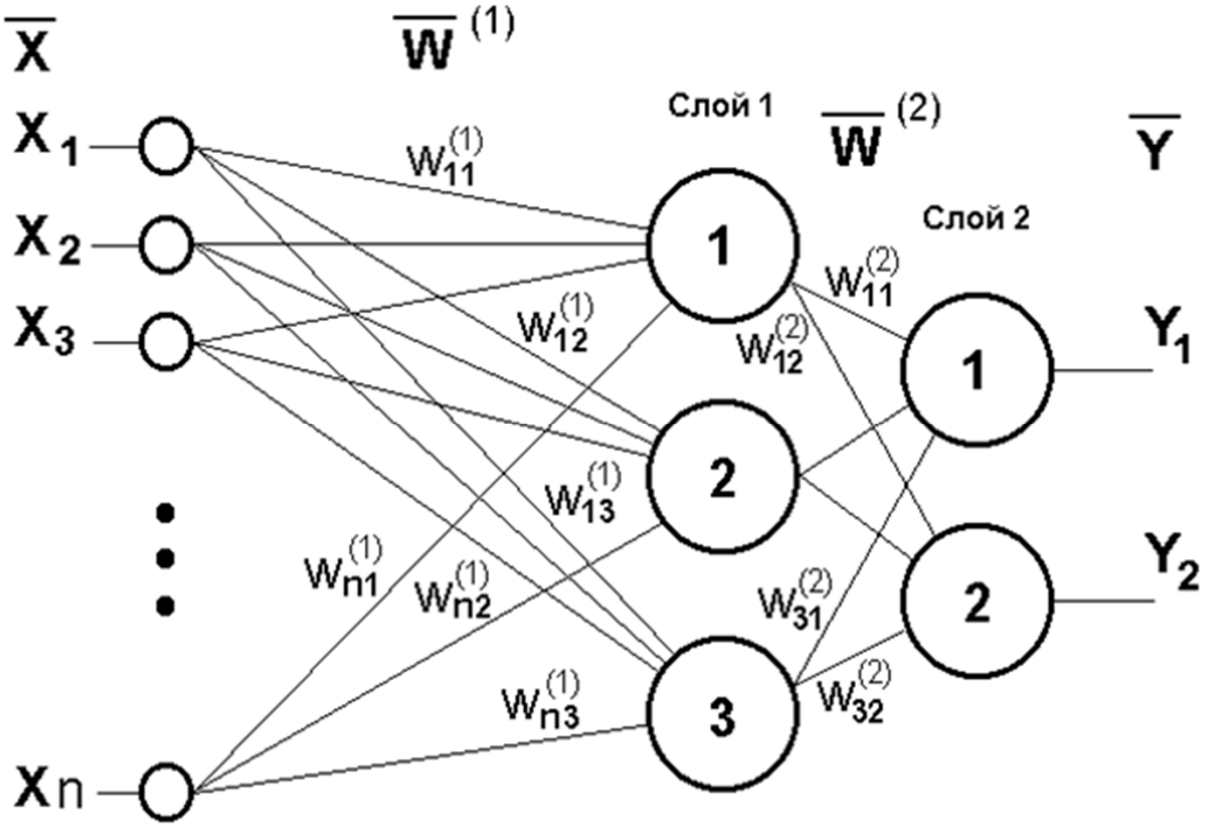
Слоистые нейронные сети – нейронные сети, в которых нейроны разбиты на отдельные группы (слои) так, что обработка информации осуществляется послойно.

- входные нейроны (input nodes), на которые подаются входные сигналы. Такие нейроны имеют, как правило, один вход с единичным весом, а значение выхода нейрона равно входному сигналу;
- выходные нейроны (output nodes), значения которых представляют результирующие выходные сигналы нейронной сети;
- скрытые нейроны (hidden nodes), не имеющие прямых связей с входными сигналами, при этом значения выходных сигналов скрытых нейронов не являются выходными сигналами сети.

Однослойная трехнейронная сеть



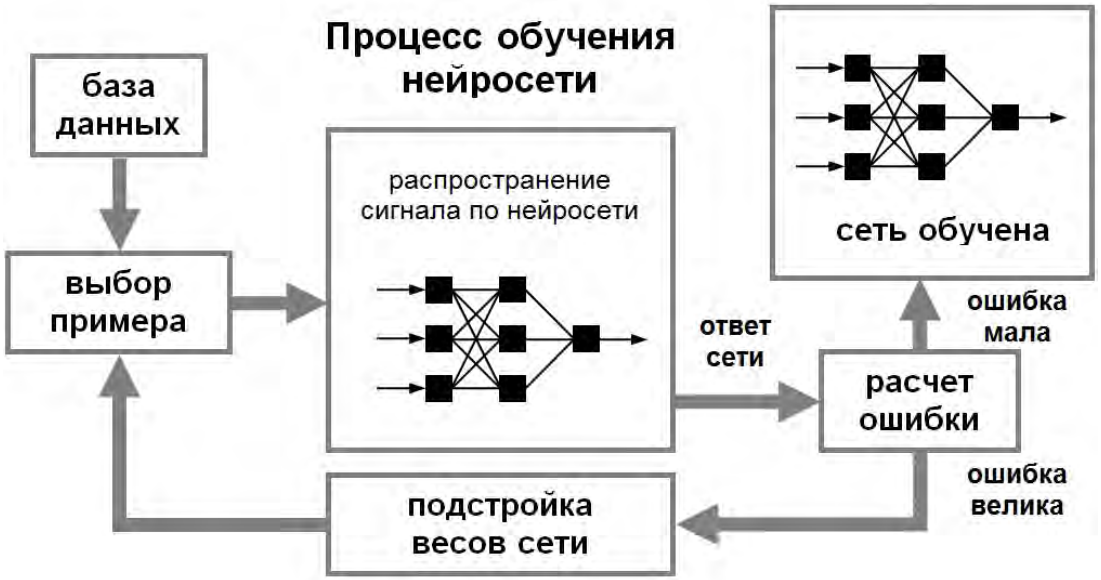
Многослойный персептрон (двухслойная сеть)



Пример «Ехать ли в отпуск?»

Из-за субъективности коэффициентов сеть обучена только для 1 человека.
Обучение нейронной сети — это математические методы, которые позволяют расставлять коэффициенты внутри нейронной сети.
Для сложной задачи нужно несколько слоев.

Обучение нейронных сетей



При **обучении с учителем** для каждого обучающего входного примера требуется знание правильного ответа. Нейронной сети предъявляются значения входных и выходных сигналов, а она по определенному алгоритму подстраивает веса синаптических связей. В процессе обучения производится корректировка весов сети.

При **обучении без учителя** выходы нейронной сети формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы. В результате такого обучения объекты или примеры распределяются по категориям, сами категории и их количество могут быть заранее не известны.

Обучение нейронных сетей

Эпоха – одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества и, возможно, проверку качества обучения на тестовом множестве.

Ошибка обучения для построенной нейронной сети вычисляется путем сравнения выходных и целевых значений. Из полученных разностей формируется **функция ошибок** – целевая функция, требующая минимизации в процессе управляемого обучения нейронной сети.

Проблема переобучения сети. Обычно обучение останавливают, когда ошибка достигает значения 0,01 – 0,001.

Нейронная сеть работает только с дискретными входными данными!

Для приведения исходных данных к одной размерности используют нормирование

$$\tilde{x} = \frac{x - Min}{Max - Min}$$

Алгоритм обратного распространения ошибки

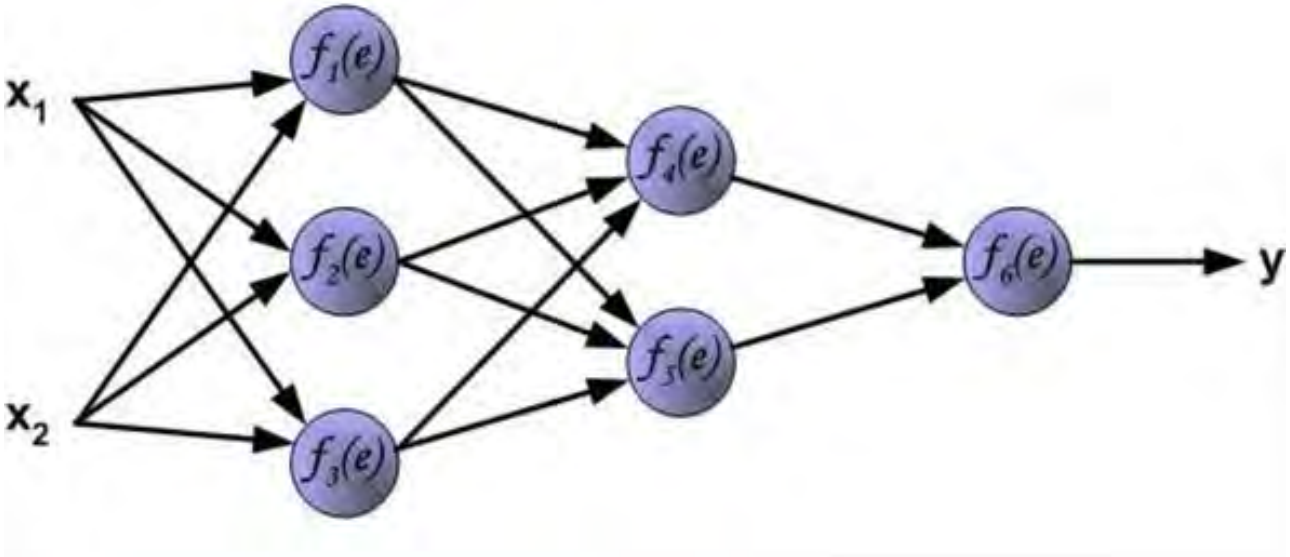
Используется для минимизации отклонения реальных значений выходных сигналов нейронной сети от требуемых.

Сигнал ошибки распространяется обратно на все нейроны, чьи выходные сигналы были входящими для последнего нейрона. Весовые коэффициенты не меняются (только поток данных). Процесс повторяется для всех слоев. Если ошибка пришла от нескольких слоев, она суммируется. Когда вычислена величина ошибки для каждого нейрона, корректируются весовые коэффициенты.

При вычислении изменений для весовых коэффициентов необходимо вычислить производную от функции активации нейрона.

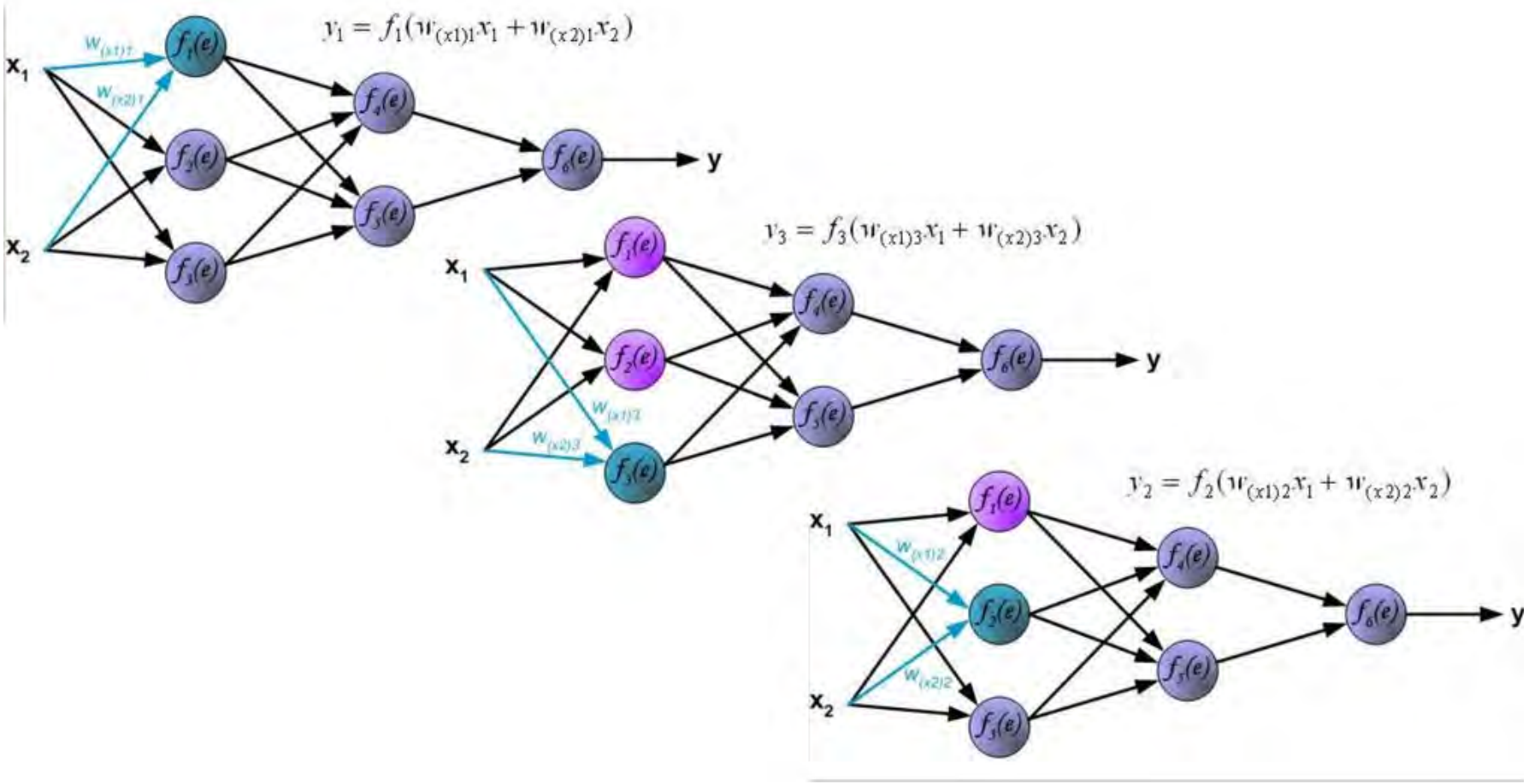
Для сигмоида: $S'(x) = S(x)*(1-S(x))$

Алгоритм обратного распространения ошибки



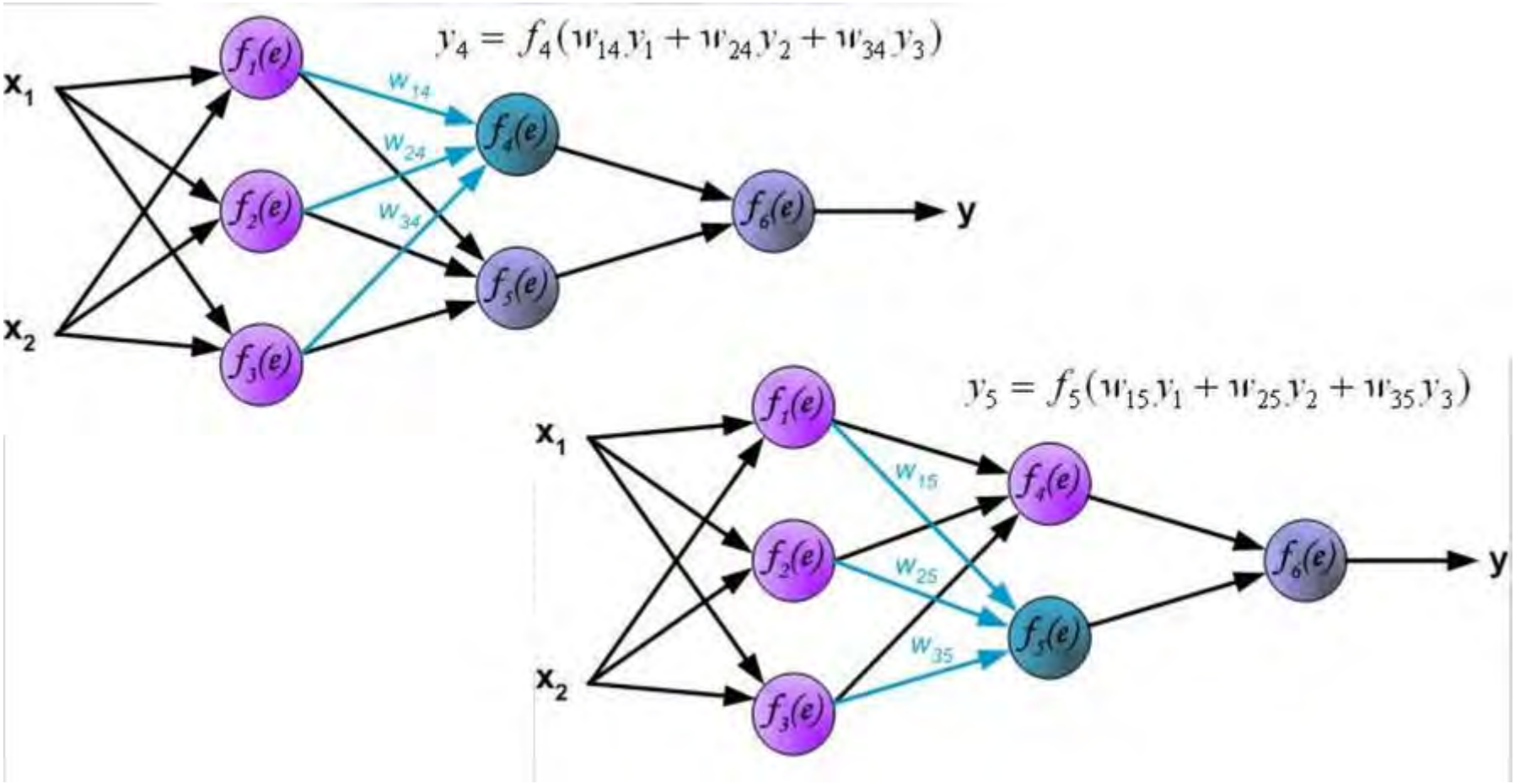
Исходная нейронная сеть (2 входа, 1 выход, 2 скрытых слоя)

Алгоритм обратного распространения ошибки



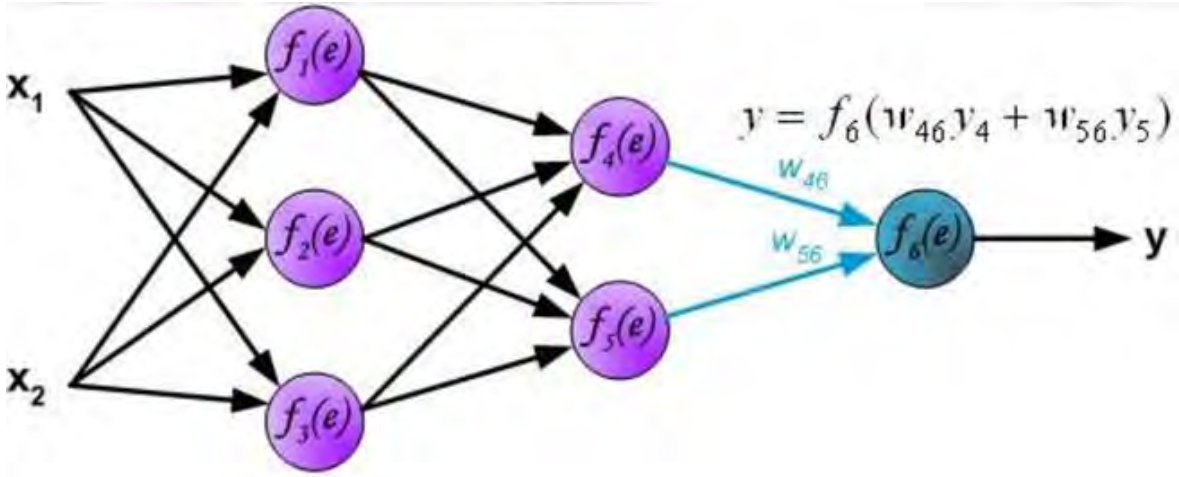
Вычисляются значения выходного вектора для первого слоя

Алгоритм обратного распространения ошибки

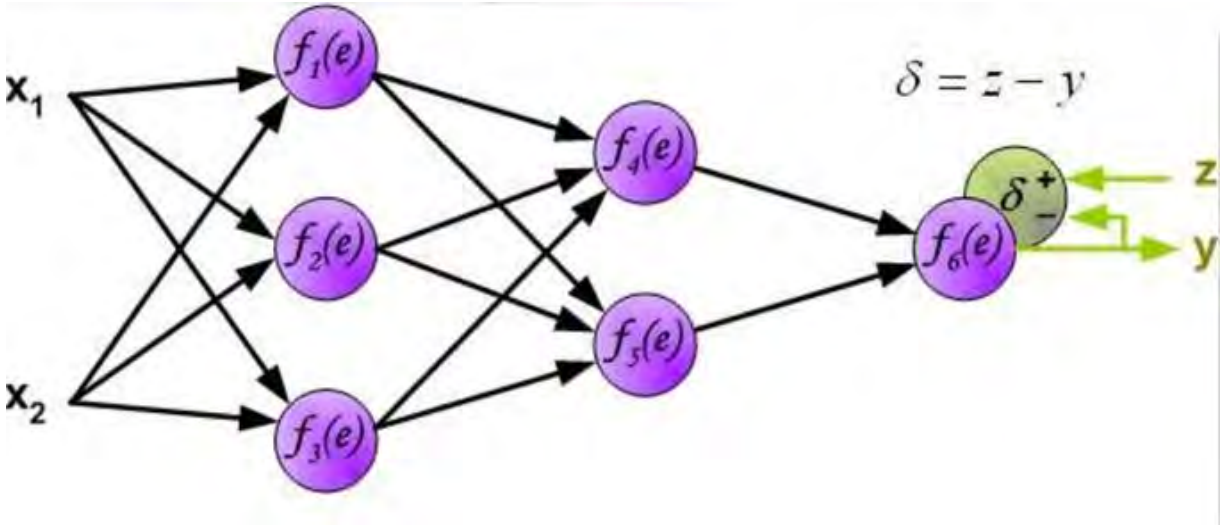


Вычисляются значения выходного вектора для второго слоя

Алгоритм обратного распространения ошибки

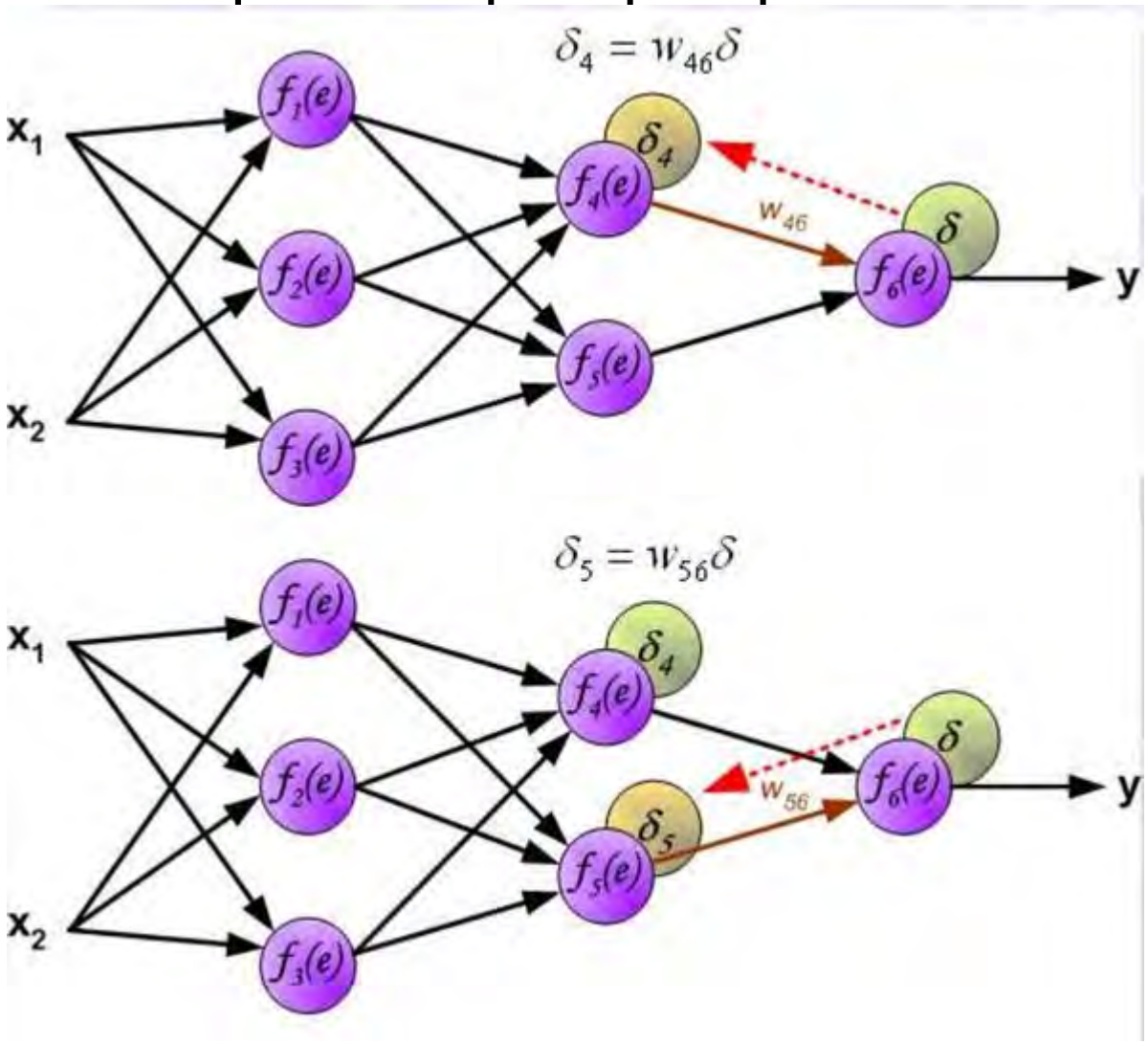


Рассчитывается выходное значение для всей сети



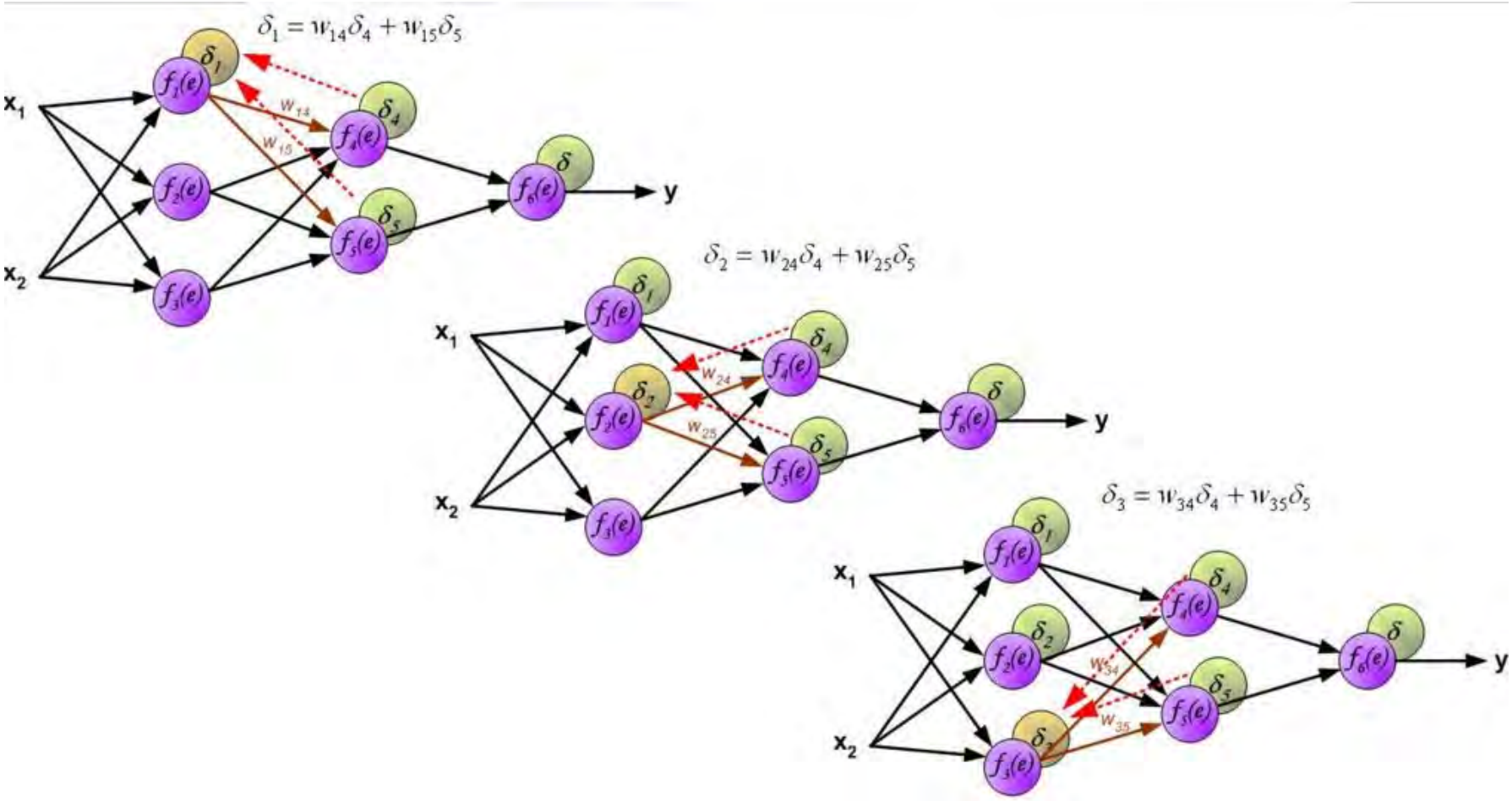
Сравнивается сигнал Y и желаемый сигнал Z. Разница – ошибка.

Алгоритм обратного распространения ошибки



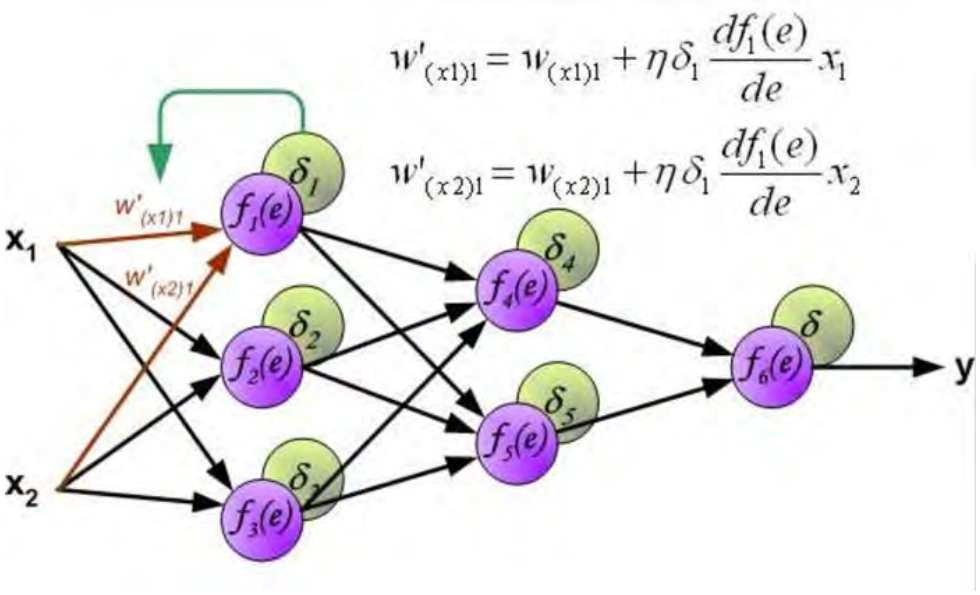
Обратное распространение ошибки

Алгоритм обратного распространения ошибки

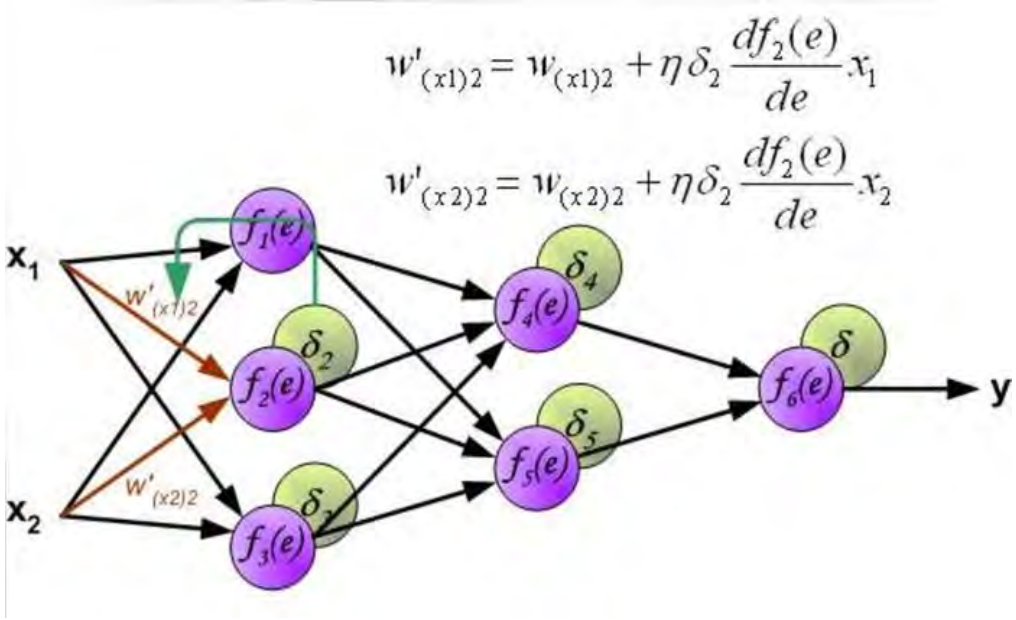


Обратное распространение ошибки

Алгоритм обратного распространения ошибки



Коррективы весовых коэффициентов

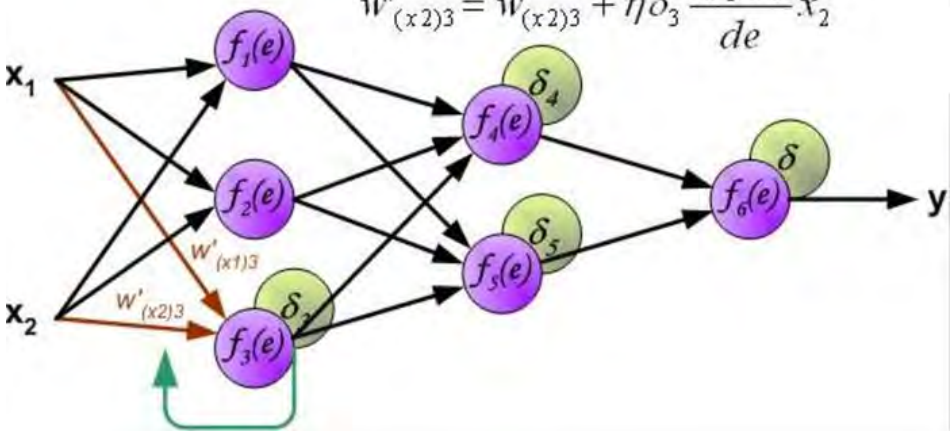


Коэффициент η влияет на скорость обучения сети (от 0 до 1)

Алгоритм обратного распространения ошибки

$$w'_{(x1)3} = w_{(x1)3} + \eta\delta_3 \frac{df_3(e)}{de} x_1$$

$$w'_{(x2)3} = w_{(x2)3} + \eta\delta_3 \frac{df_3(e)}{de} x_2$$

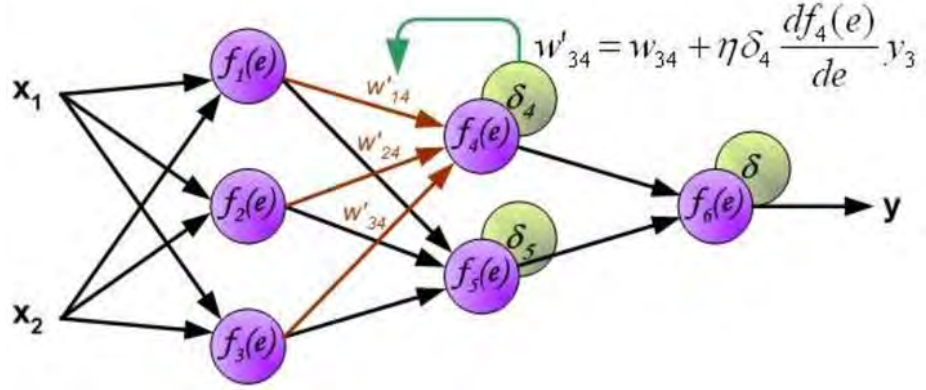


Коррективы весовых коэффициентов

$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$

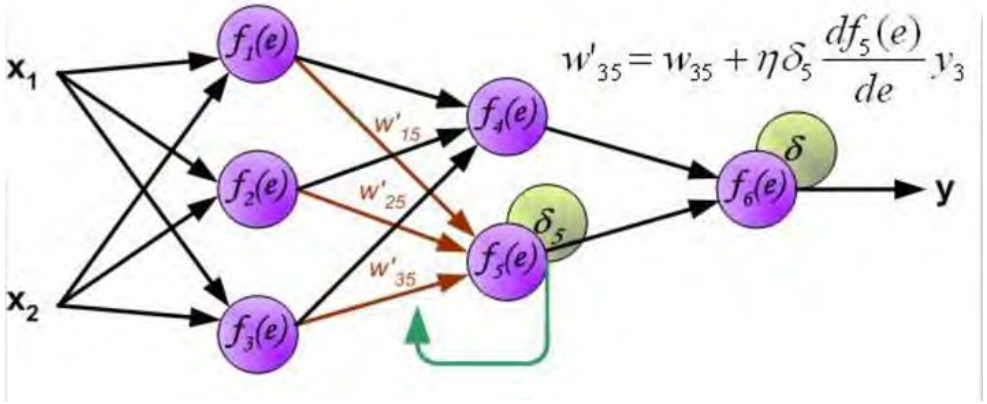


Алгоритм обратного распространения ошибки

$$w'_{15} = w_{15} + \eta\delta_5 \frac{df_5(e)}{de} y_1$$

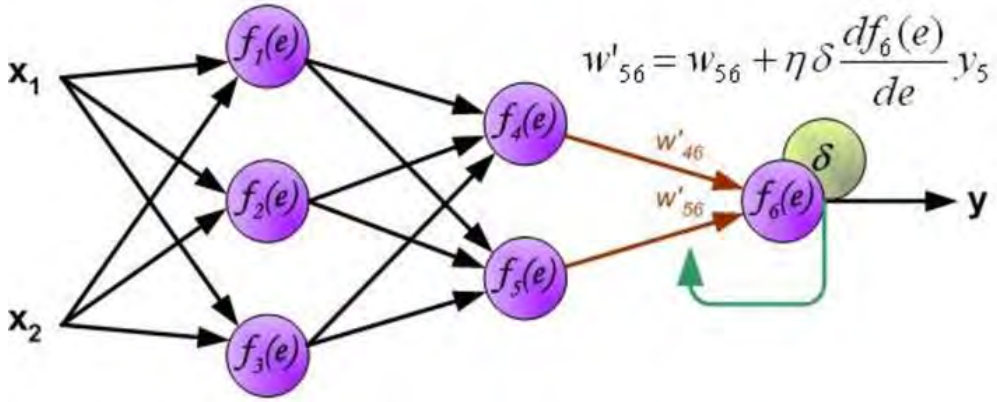
$$w'_{25} = w_{25} + \eta\delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta\delta_5 \frac{df_5(e)}{de} y_3$$



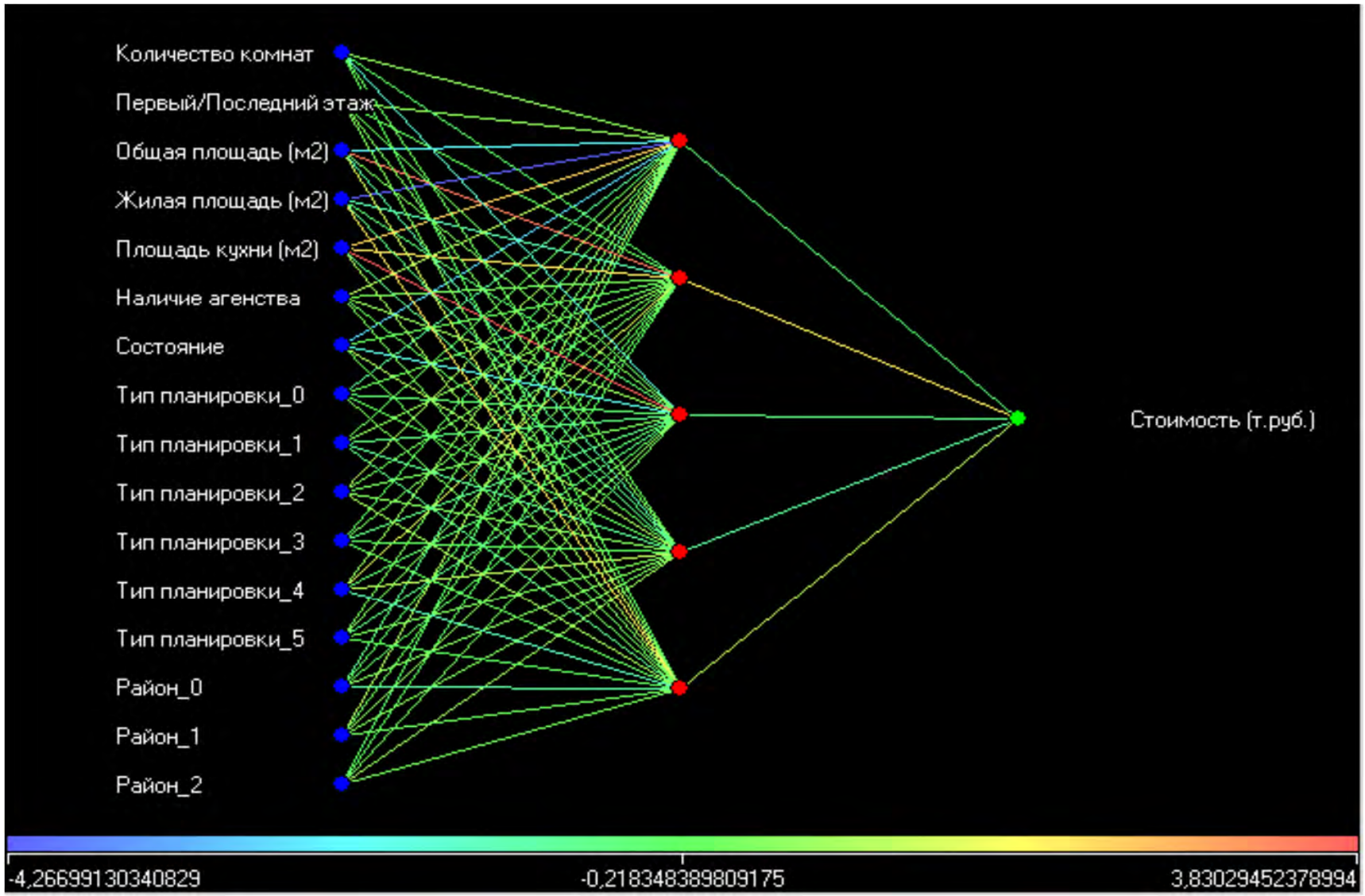
$$w'_{46} = w_{46} + \eta\delta \frac{df_6(e)}{de} y_4$$

$$w'_{56} = w_{56} + \eta\delta \frac{df_6(e)}{de} y_5$$



Коррективы весовых коэффициентов

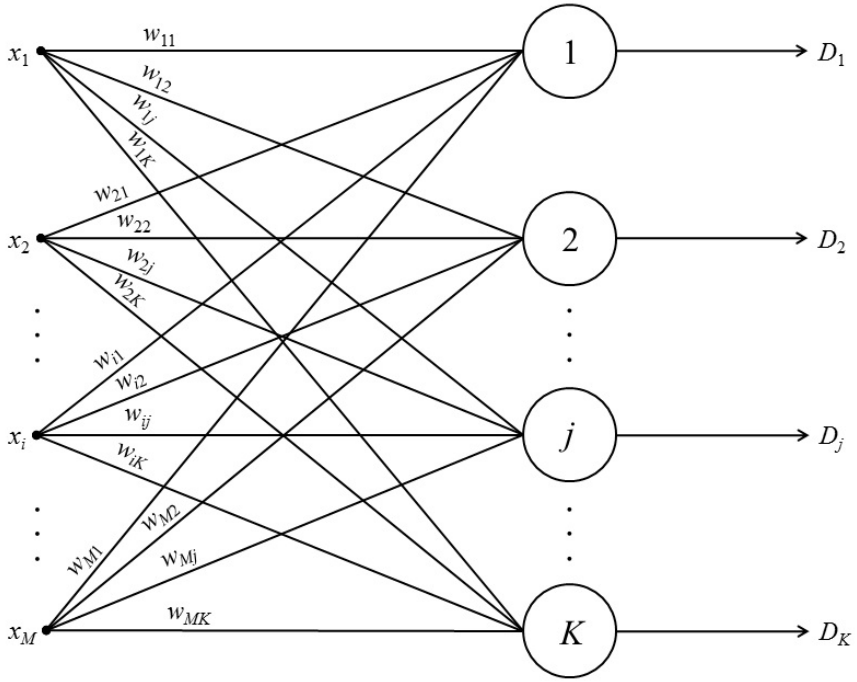
Граф нейросети

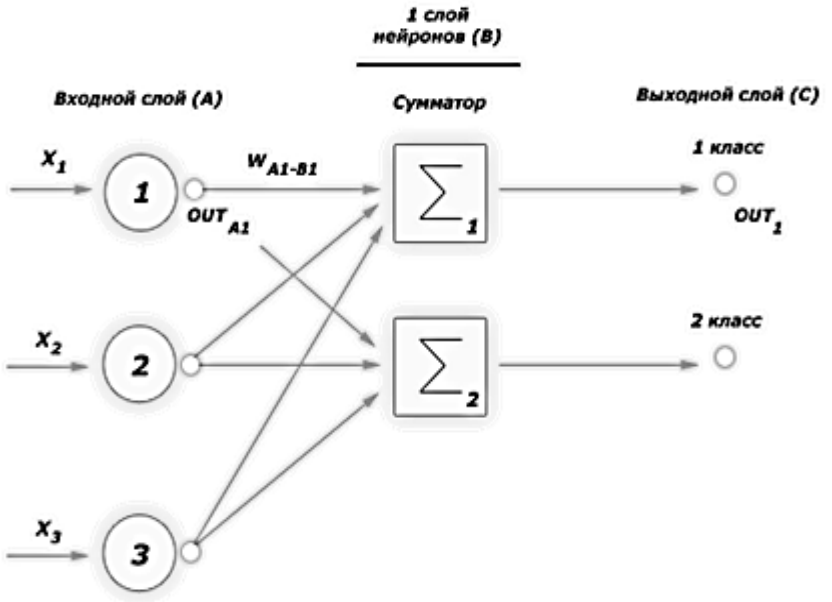


Самоорганизующаяся карта Кохонена

Карты Кохонена – это соревновательная нейронная сеть с обучением без учителя. Наиболее распространенное применение – решение задачи кластеризации.

Сеть Кохонена представляет собой два слоя: входной и выходной.





Обучение карты Кохонена

На входы подаются данные, но сеть подстраивается не под эталонное значение выхода, а под закономерности во входных данных.

Сначала нужно указать оптимальное число классов (выходных нейронов).

На входы сети подается случайный обучающий пример текущей эпохи обучения и рассчитываются евклидовые расстояния от входного вектора до центров всех кластеров.

$$R_j = \sqrt{\sum_{i=1}^{M} (\tilde{x}_i - w_{ij})^2}$$

По наименьшему из значений R_j выбирается нейрон-победитель j, в наибольшей степени близкий по значениям с входным вектором. Для выбранного нейрона (и только для него) выполняется коррекция весовых коэффициентов:

$$w_{ij}^{(q-1)} = w_{ij}^{(q)} + v\left(\tilde{x}_i - w_{ij}^{(q)}\right)$$

где v – коэффициент скорости обучения

Коэффициент скорости обучения может задаваться постоянным из пределов (0, 1] или переменным значением, постепенно уменьшающимся от эпохи к эпохе.

Обучение карты Кохонена

Цикл повторяется до выполнения одного или нескольких условий окончания:

- исчерпано заданное предельное количество эпох обучения;
- не произошло значимого изменения весовых коэффициентов в пределах заданной точности на протяжении последней эпохи обучения;
- исчерпано заданное предельное физическое время обучения.

После предъявления достаточного числа входных векторов синаптические веса сети становятся способны определить кластеры. Веса организуются так, что топологически близкие узлы чувствительны к похожим входным сигналам.

Пример: кластеризация данных об успеваемости студентов

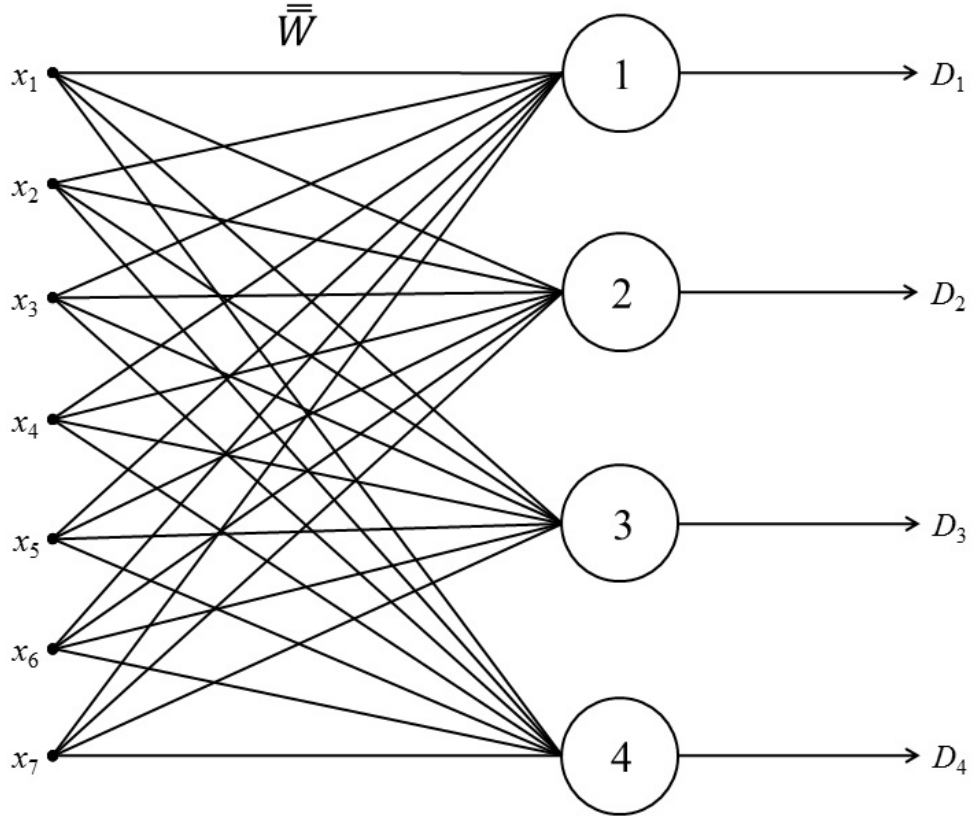
Таблица 1. Исходная выборка данных об успеваемости студентов

№	Фамилия	Пол x_1	Получ. все зачеты x_2	Рейтинг по дисциплинам:					Коэфф. стипендии x_8
				история x_3	инж. графика x_4	матем. x_5	орг. химия x_6	физика x_7	
1	Варданян	М	Да	60	79	60	72	63	1,00
2	Горбунов	М	Нет	60	61	30	5	17	0,00
3	Гуменюк	Ж	Нет	60	61	30	66	58	0,00
4	Егоров	М	Да	85	78	72	70	85	1,25
…									
10	Нетреба	М	Нет	60	56	30	16	17	0,00
…									
20	Шевченко	М	Нет	55	60	30	8	60	0,00

Распределение примеров должно осуществляться строго по 4 кластерам. В качестве входных переменных используем x_1 – x_7, переменная x_8 не будет использоваться для обучения, однако информация о ее значениях будет задействована в ходе кластерного анализа.

Пример: кластеризация данных об успеваемости студентов

Таким образом, структурно сеть будет состоять из единственного слоя нейронов, имеющего 7 входов и 4 выхода.



Пример: кластеризация данных об успеваемости студентов

Выполним нормализацию значений входных переменных выборки в пределах [0, 1]. Дискретные значения опишем следующим образом:

– пол студента: 0 – женский, 1 – мужской;

– наличие всех зачетов: 0 – нет, 1 – да.

Таблица 2. Нормализованная выборка данных об успеваемости студентов

№ примера							
1	1,00	1,00	0,17	0,78	0,70	0,77	0,68
2	1,00	0,00	0,17	0,58	0,35	0,00	0,00
3	0,00	0,00	0,17	0,58	0,35	0,70	0,60
4	1,00	1,00	1,00	0,77	0,84	0,75	1,00
…							
10	1,00	0,00	0,17	0,52	0,35	0,13	0,00
…							
20	1,00	0,00	0,00	0,57	0,35	0,03	0,63

Пример: кластеризация данных об успеваемости студентов

Проинициализируем все 28 весовых коэффициентов нейронной сети значениями, представленными в табл. 3, с учетом ограничения

$$0{,}5 - \frac{1}{\sqrt{M}} \le w_{ij} \le 0{,}5 + \frac{1}{\sqrt{M}}$$

где M – количество входных переменных сети

Таблица 3. Начальные значения весовых коэффициентов сети Кохонена

№ кластера j	Весовые коэффициенты w_{ij}						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,20	0,20	0,30	0,40	0,40	0,20	0,50
2	0,20	0,80	0,70	0,80	0,70	0,70	0,80
3	0,80	0,20	0,50	0,50	0,40	0,40	0,40
4	0,80	0,80	0,60	0,70	0,70	0,60	0,70

Пример: кластеризация данных об успеваемости студентов

Выберем начальный коэффициент скорости обучения, равный 0,30, уменьшающийся с каждой эпохой на 0,05. Таким образом, будет выполнено 6 эпох обучения с различным коэффициентом скорости, на каждой из которых будет 20 корректировок весов одного из нейронов.

Подадим на вход нейронной сети случайно выбранный нормализованный пример № 10 и рассчитаем расстояния до текущих центров четырех кластеров. Они соответственно будут равны 0,98, 1,65, 0,65 и 1,32.

Наименьшее расстояние соответствует третьему кластеру, из чего делаем вывод, что третий нейрон – нейрон-победитель и именно его веса должны быть скорректированы по соотношению.

Новые значения весовых коэффициентов составят: $w_{13} = 0{,}86$, $w_{23} = 0{,}14$, $w_{33} = 0{,}40$, $w_{43} = 0{,}51$, $w_{53} = 0{,}39$, $w_{63} = 0{,}32$, $w_{73} = 0{,}28$.

Веса остальных нейронов при этом не изменяются.

Далее аналогичным образом на входы сети предъявляются остальные примеры выборки в случайной последовательности.

Пример: кластеризация данных об успеваемости студентов

После предъявления всех 20 примеров начинается следующая эпоха обучения, при этом коэффициент скорости уменьшаем на 0,05. В результате полного цикла обучения сети Кохонена получаем итоговые значения весов.

Таблица 4. Итоговые значения весовых коэффициентов нейронной сети Кохонена

№ кластера j	Весовые коэффициенты w_{ij}						
	w_{1j}	w_{2j}	w_{3j}	w_{4j}	w_{5j}	w_{6j}	w_{7j}
1	0,06	0,06	0,21	0,52	0,36	0,55	0,57
2	0,00	1,00	0,50	0,80	0,80	0,80	0,73
3	1,00	0,00	0,04	0,48	0,26	0,22	0,42
4	1,00	0,99	0,69	0,77	0,79	0,78	0,81

Пример: кластеризация данных об успеваемости студентов

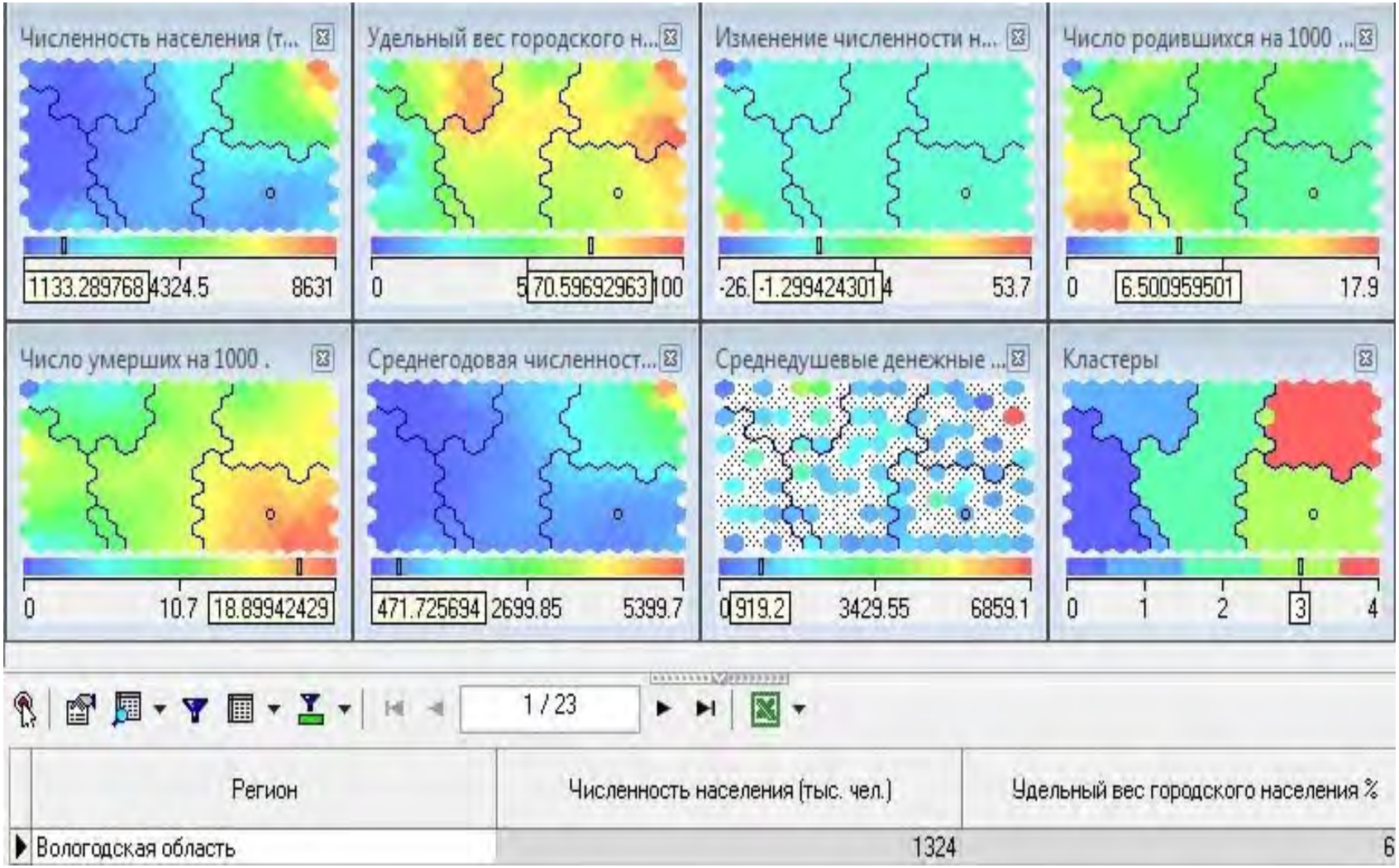
Для обученной нейронной сети выполним процедуру кластерного анализа. Все четыре кластера оказались заполнены. Однако количественный состав их разный. В 1-й кластер попал единственный пример – № 3. Во 2-м, самом объемном кластере оказались примеры №№ 5–7, 11, 12, 14, 16, 18. В 3-й кластер вошли примеры №№ 2, 8–10, 13, 15, 20. В 4-й – №№ 1, 4, 17, 19.

Выводы о факте получения стипендии в описаниях кластеров сделаны на основе анализа значений переменной x_8, не участвовавшей в процессе обучения.

Таблица 5. Результаты кластерного анализа

№ кластера	Размер кластера	Пол	Получ. все зачеты	Средний рейтинг	Коэфф. стипендии	Описание
1	1	Ж	Нет	55	0,00	Удовлетворительно успевающие студенты-девушки, не имеющие одного или нескольких зачетов и не получающие стипендию
2	8	Ж	Да	72	0,97	Хорошо успевающие студенты-девушки, имеющие все зачеты и в большинстве своем получающие стипендию
3	7	М	Нет	40	0,00	Неуспевающие студенты-юноши, не имеющие одного или нескольких зачетов и не получающие стипендию
4	4	М	Да	73	0,94	Хорошо успевающие студенты-юноши, имеющие все зачеты и в большинстве своем получающие стипендию

Визуализация карты Кохонена



Метод k-means (метод k-средних)

Общая идея алгоритма: разбить множество элементов на заранее известное число кластеров.

Описание алгоритма

1) Первоначальное распределение объектов по кластерам. Выбирается число *k*, и на первом шаге эти точки считаются «центрами» кластеров. Каждому кластеру соответствует один центр. В результате каждый объект назначен определенному кластеру.

2) Итеративный процесс.

   Вычисляются центры кластеров (центроиды), которыми затем и далее считаются покоординатные средние кластеров. Объекты опять перераспределяются.

Таким образом, необходимо минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров $V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$

где k — число кластеров, S_i— полученные кластеры, i=1…k и μ_i и — центры масс векторов $x_i \in S_i$.

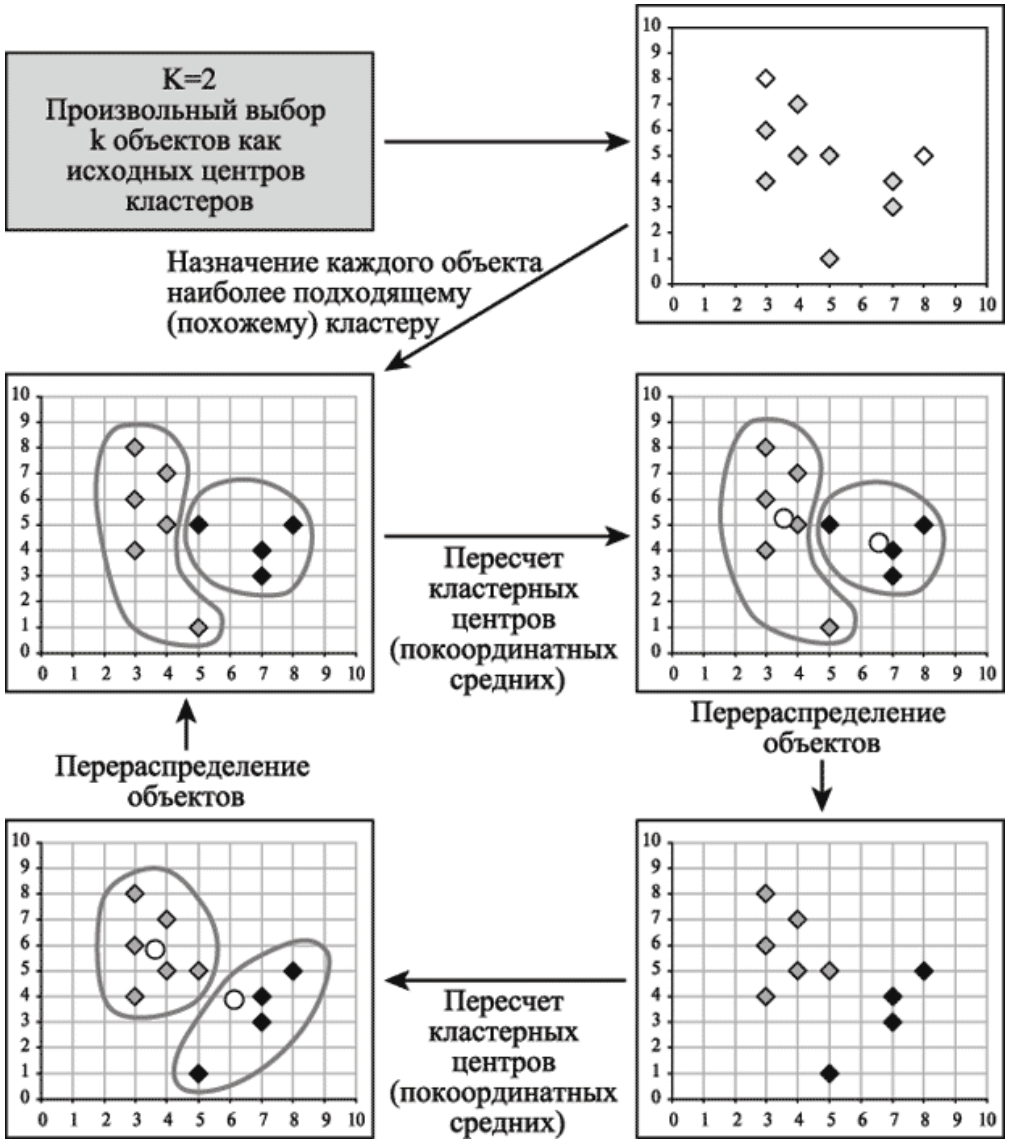
Метод k-means (метод k-средних)

Процесс вычисления центров и перераспределения объектов продолжается до тех пор, пока не выполнено одно из условий:

- кластерные центры стабилизировались, т.е. все наблюдения принадлежат кластеру, которому принадлежали до текущей итерации;
- число итераций равно максимальному числу итераций.

После получений результатов следует проверить правильность кластеризации (т.е. оценить, насколько кластеры отличаются друг от друга). Для этого рассчитываются средние значения для каждого кластера. При хорошей кластеризации должны быть получены сильно отличающиеся средние для всех измерений.

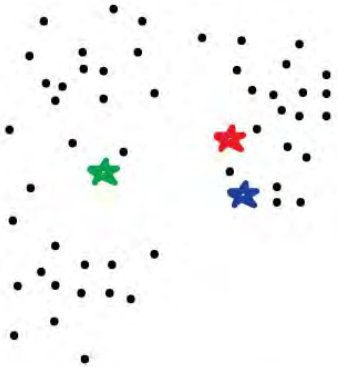
Метод k-means (k=2)



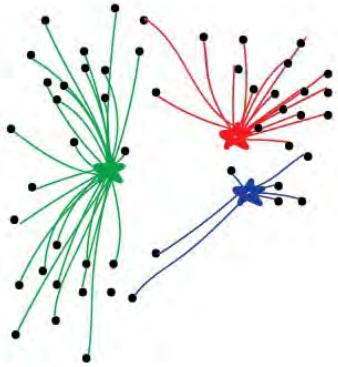
Метод k-means

Ставим три ларька с шаурмой оптимальным образом

(иллюстрируя метод K-средних)

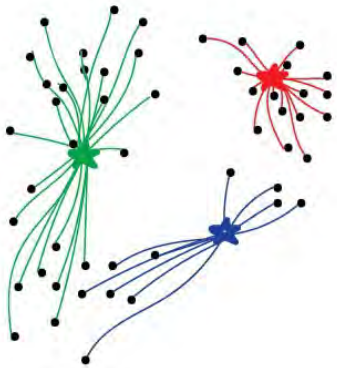


1. Ставим ларьки с шаурмой в случайных местах

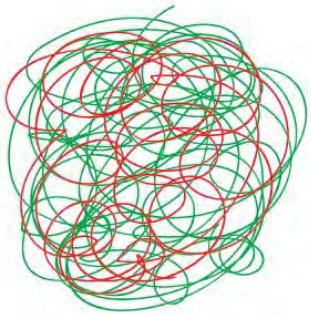


2. Смотрим в какой кому ближе идти

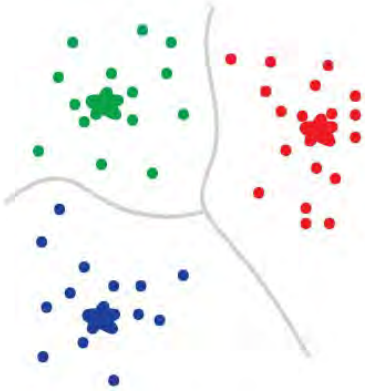
3. Двигаем ларьки ближе к центрам их популярности



4. Снова смотрим и двигаем



5. Повторяем много раз



6. Готово, вы великолепны!

Метод k-means (метод k-средних)

Достоинства:

- простота использования;
- быстрота использования;
- понятность и прозрачность алгоритма.
- алгоритм может медленно работать на больших базах данных.

Недостатки:

- алгоритм слишком чувствителен к выбросам, которые могут искажать среднее;
- число кластеров надо знать заранее.

Алгоритм Apriori

Является наиболее популярным методом поиска ассоциативных правил. Работа данного алгоритма состоит из нескольких этапов:

- формирование кандидатов;
- подсчет кандидатов.

Формирование кандидатов – этап, на котором алгоритм, сканируя базу данных, создает множество i-элементных кандидатов (i – номер этапа).

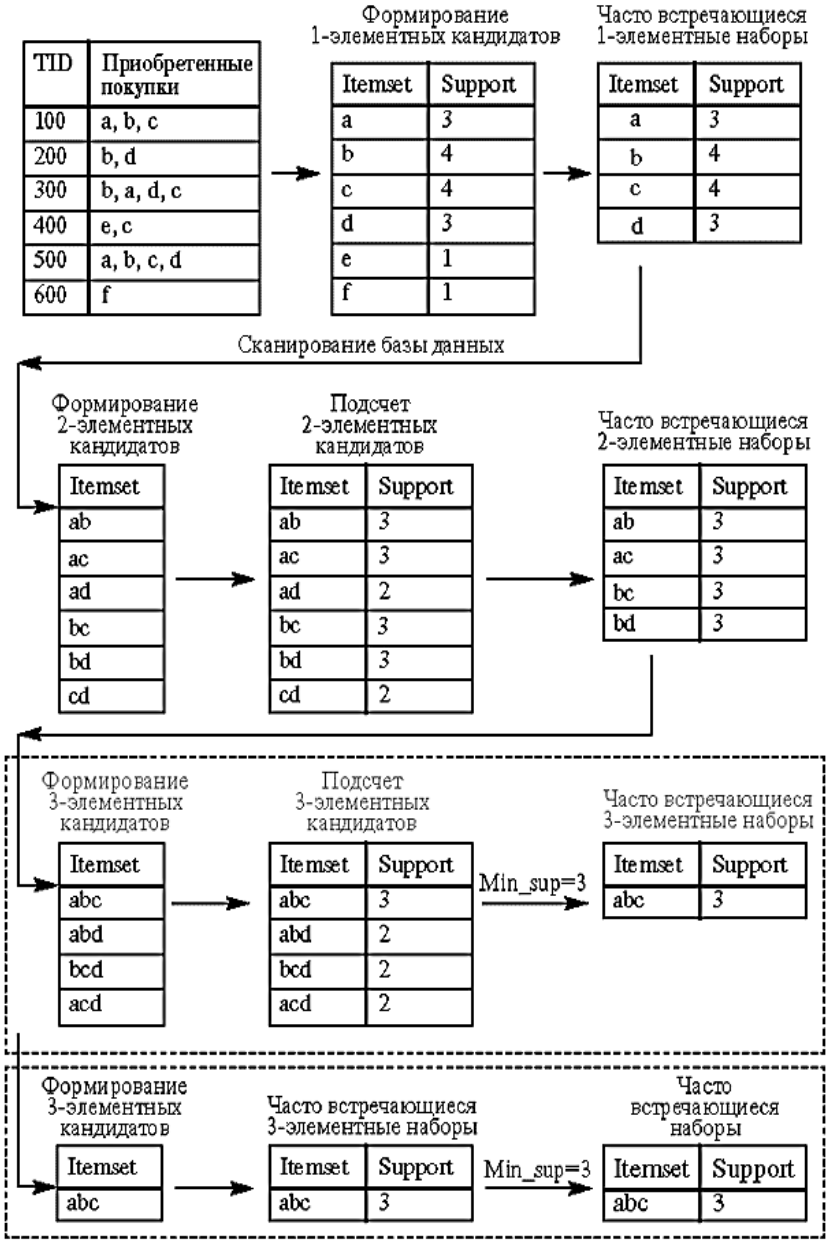
Подсчет кандидатов – этап, на котором вычисляется поддержка каждого i-элементного кандидата. Здесь же осуществляется отсечение кандидатов, поддержка которых меньше минимума, установленного пользователем.

Алгоритм Apriori (пример)

ID	Приобретенные покупки	Присвоенные переменные
100	Хлеб, молоко, печенье	a, b, c
200	Молоко, сметана	b, d
300	Молоко, хлеб, сметана, печенье	b, a, d, c
400	Колбаса, печенье	e, c
500	Хлеб, молоко, печенье, сметана	a, b, c, d
600	Конфеты	f

Каждому товару для удобства присвоена переменная. Необходимо найти закономерности между покупками. Минимальный уровень поддержки (min_sup) равен 3.

Алгоритм Apriori (пример)



Отсечение кандидатов происходит на основе предположения о том, что у часто встречающегося набора товаров все подмножества должны быть часто встречающимися.

Так как наборы товаров ad и cd были отброшены как нечасто встречающиеся, алгоритм не рассматривал набор товаров abd, bcd, acd.

Визуализация данных

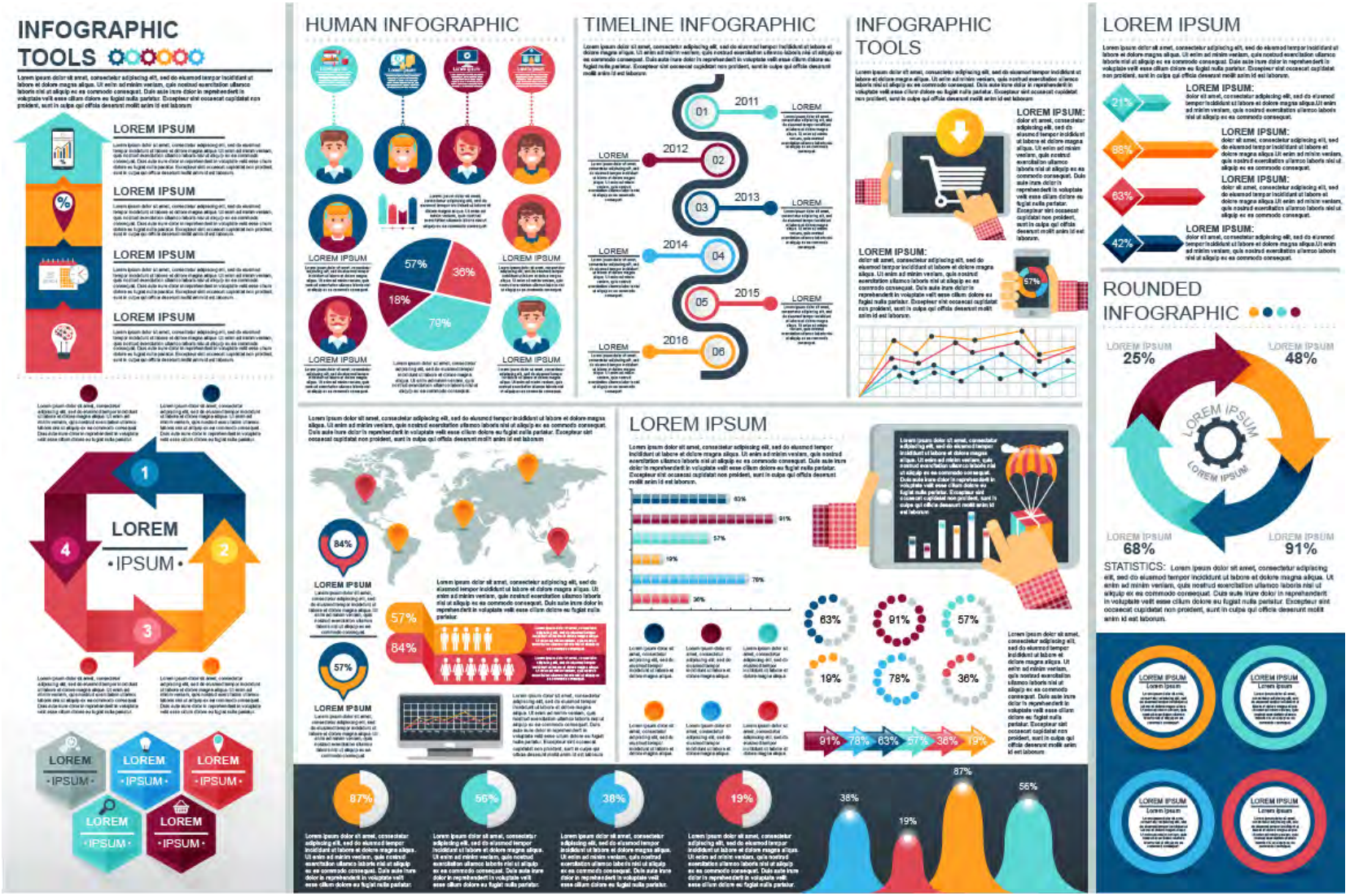
Визуализация – это инструментарий, который позволяет увидеть конечный результат вычислений. В результате создается графический образ анализируемых данных. Применение визуализации помогает увидеть аномалии, структуры, тренды.

90% информации человек воспринимает через зрение, около 50% нейронов мозга задействованы в обработке визуальной информации, 80% информации человек запоминает из увиденного, на 323% лучше человек выполняет инструкцию, если она содержит иллюстрации.

Желязны Д. Говори на языке диаграмм: Пособие по визуальным коммуникациям для руководителей / Пер. с англ. – М.: Институт комплексных стратегических исследований, 2004. – 220 с.

Инфографика

Графический способ подачи информации, данных и знаний, целью которого является быстро и чётко преподносить сложную информацию.

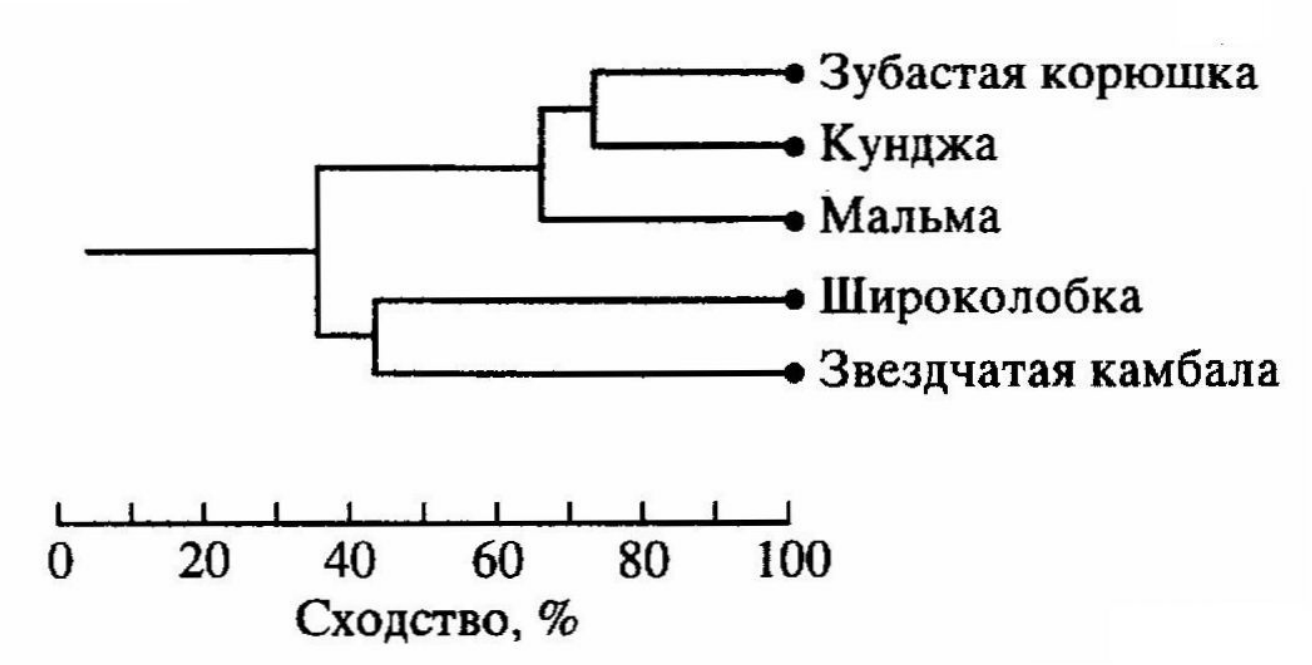


Примеры визуализации

- Радиальная (лепестковая) диаграмма

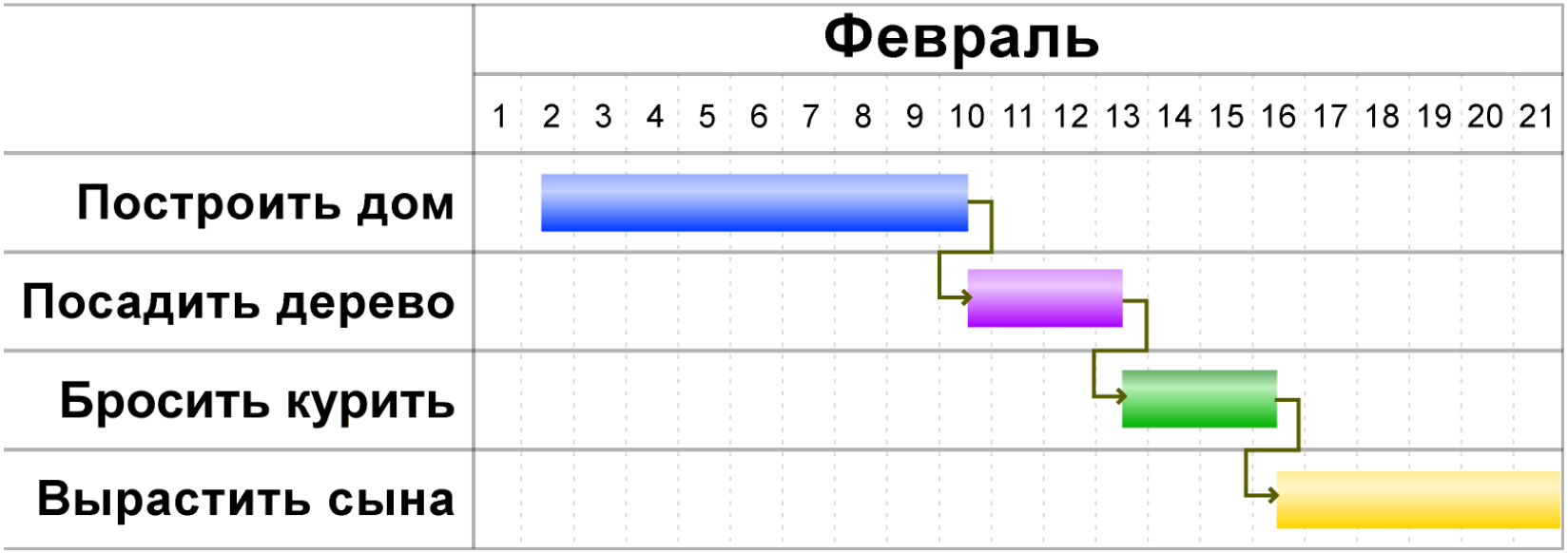
Примеры визуализации

- Дендрограмма



Примеры визуализации

- Диаграмма Ганта

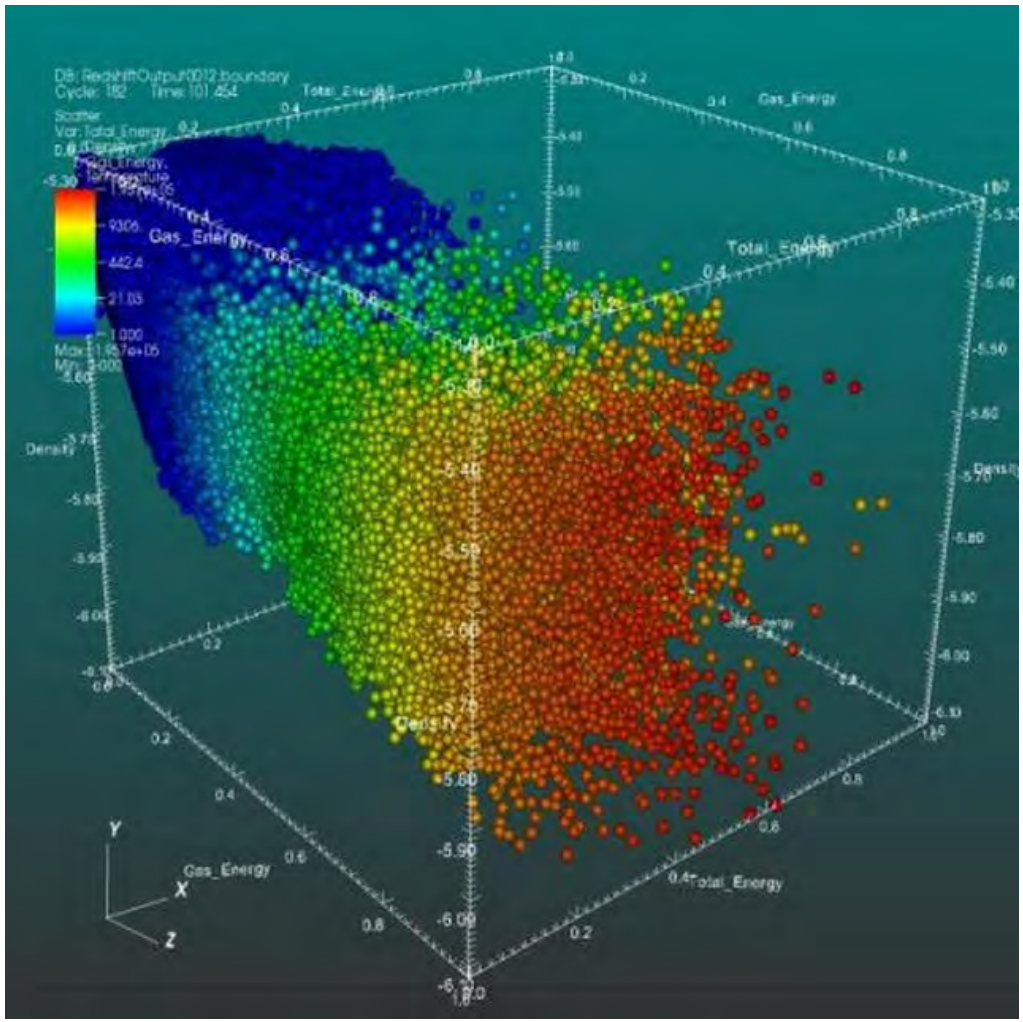


Примеры визуализации

- Многофакторный анализ

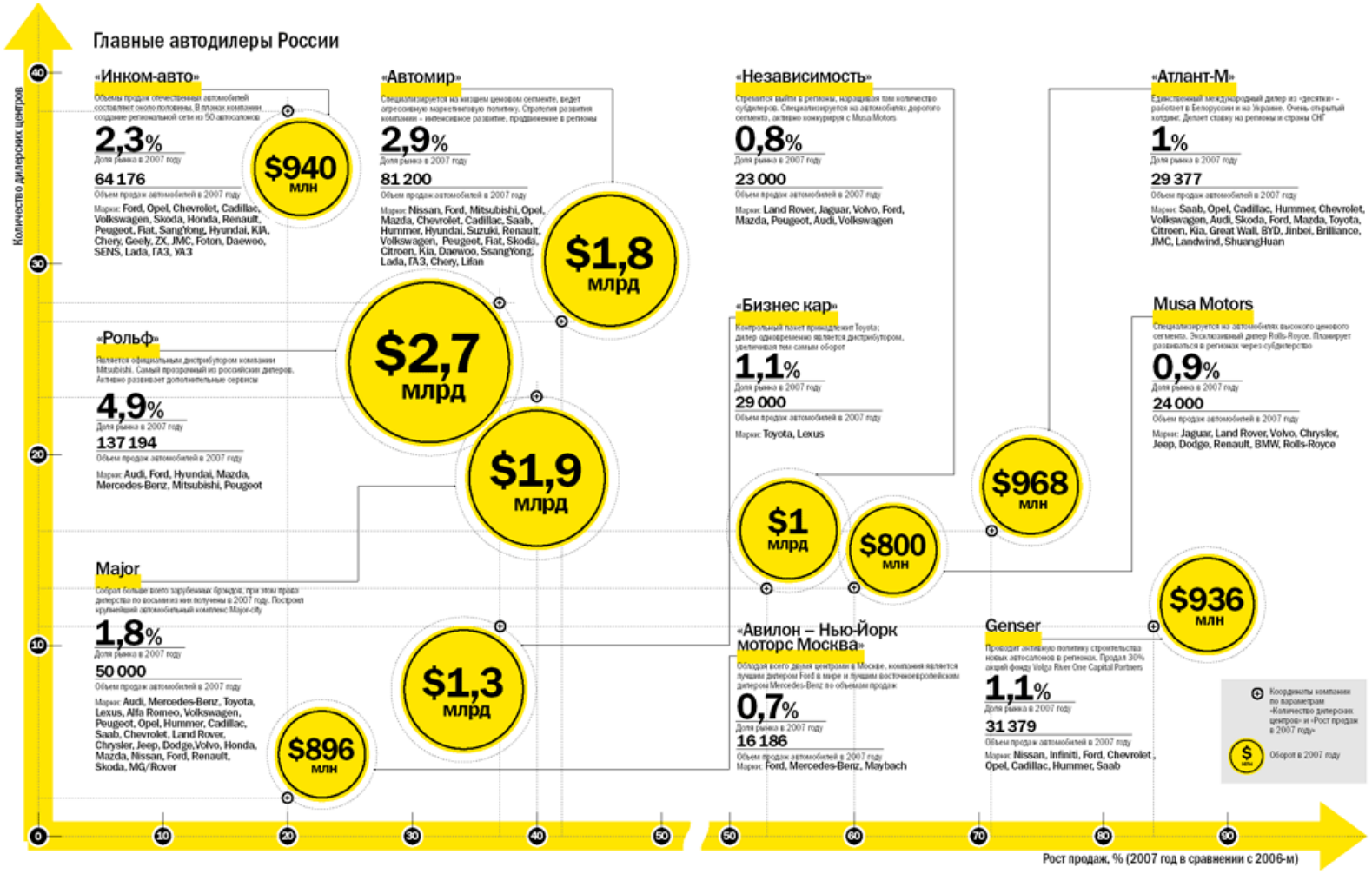
Примеры визуализации

- Многофакторный анализ



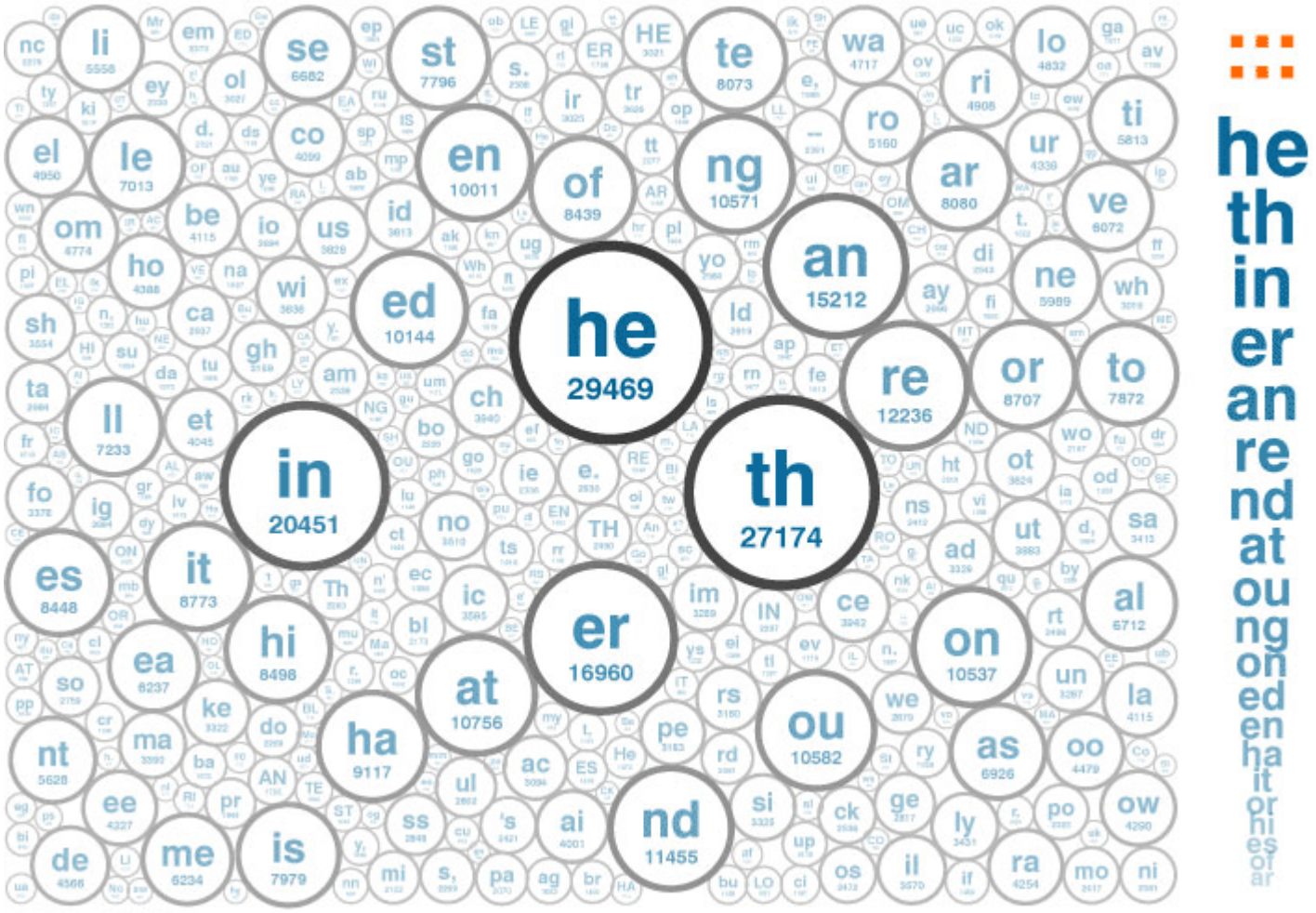
Примеры визуализации

- Площадная диаграмма



Примеры визуализации

- Облако тегов



Примеры визуализации

- Тепловая диаграмма

Примеры визуализации

- Ментальная карта (mind map)

Примеры визуализации

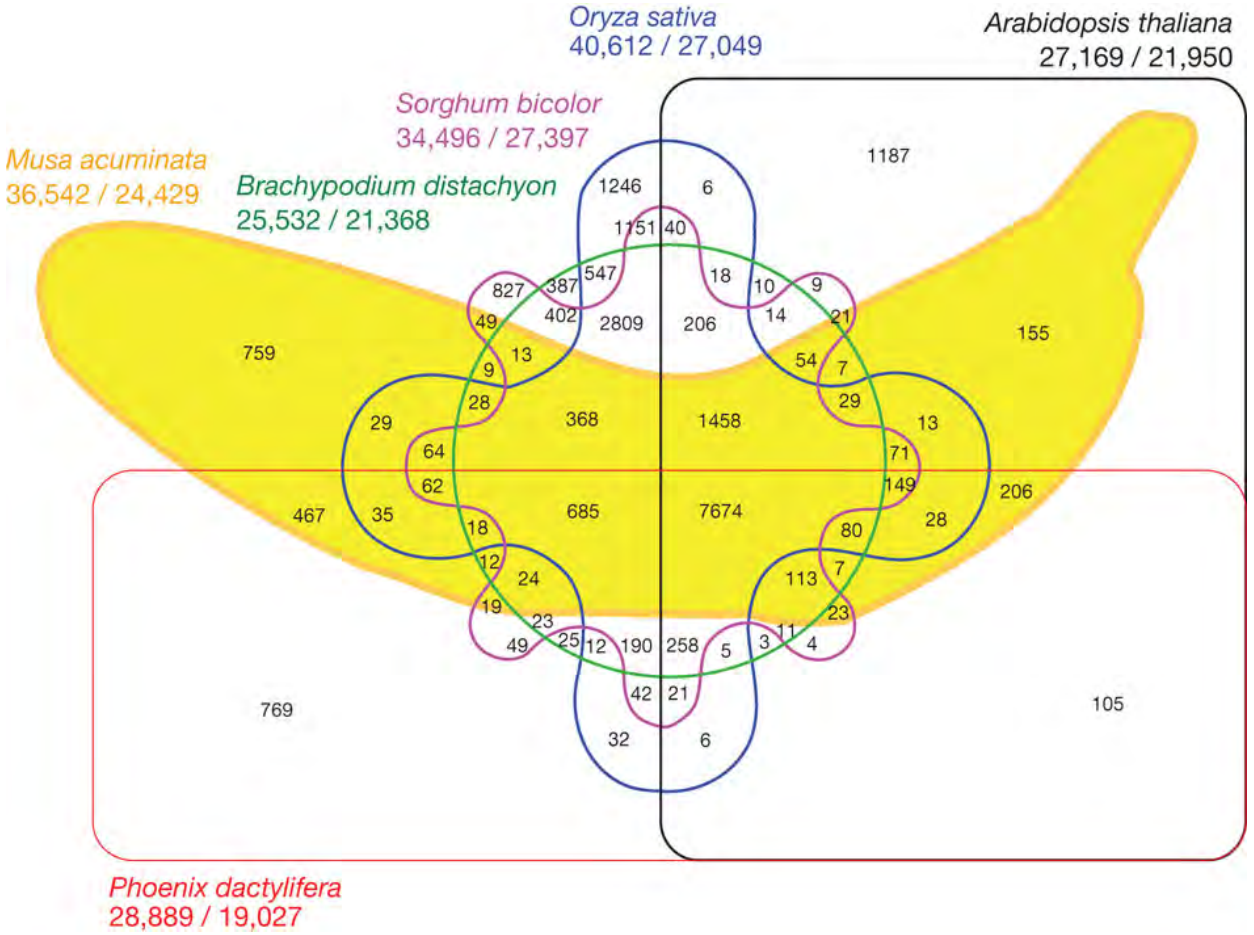
- Диаграмма Исикавы («рыбьей кости»)

Примеры визуализации

- Диаграмма Венна-Эйлера

Примеры визуализации

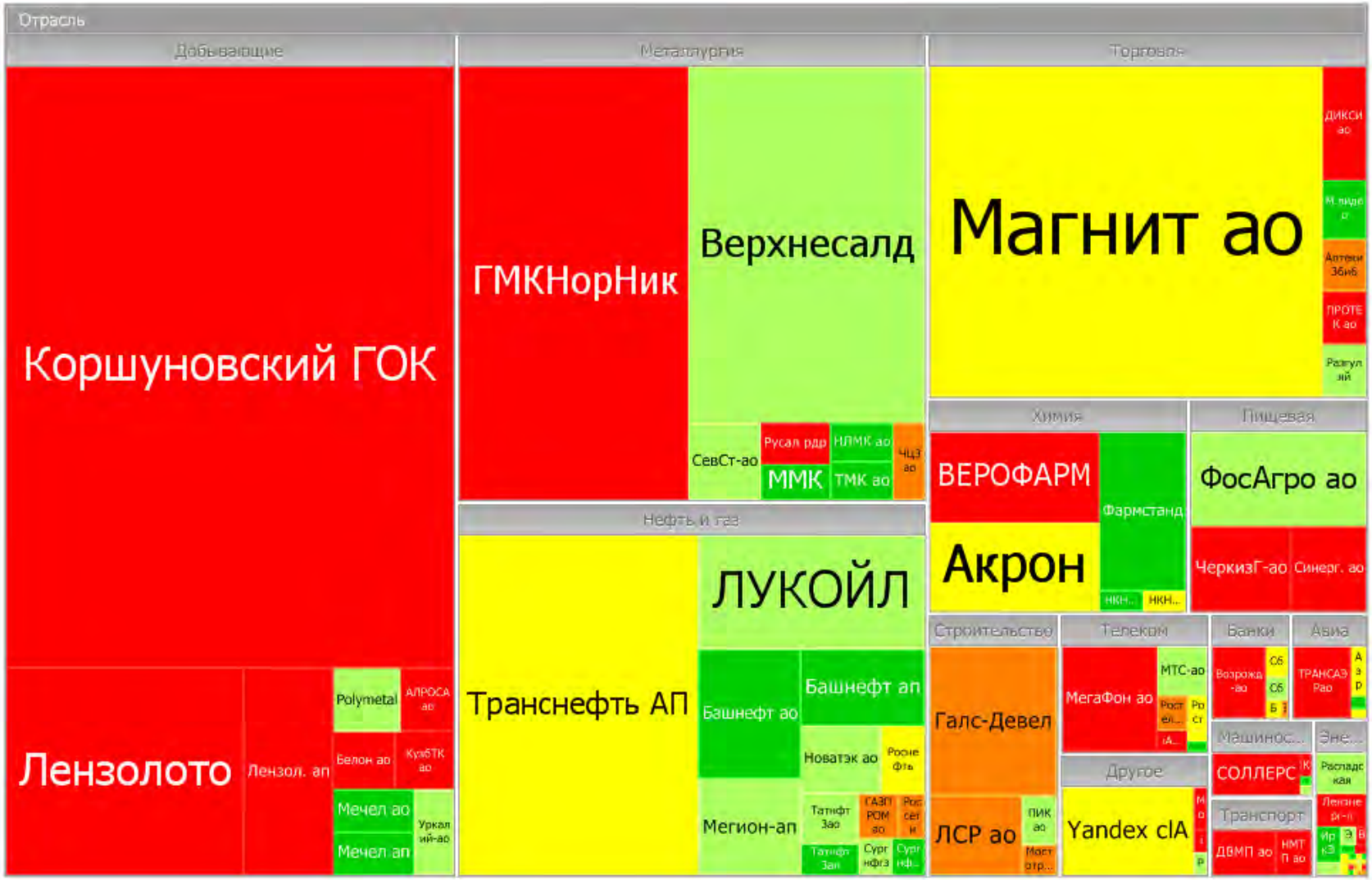
- Диаграмма Венна-Эйлера



Геном банана

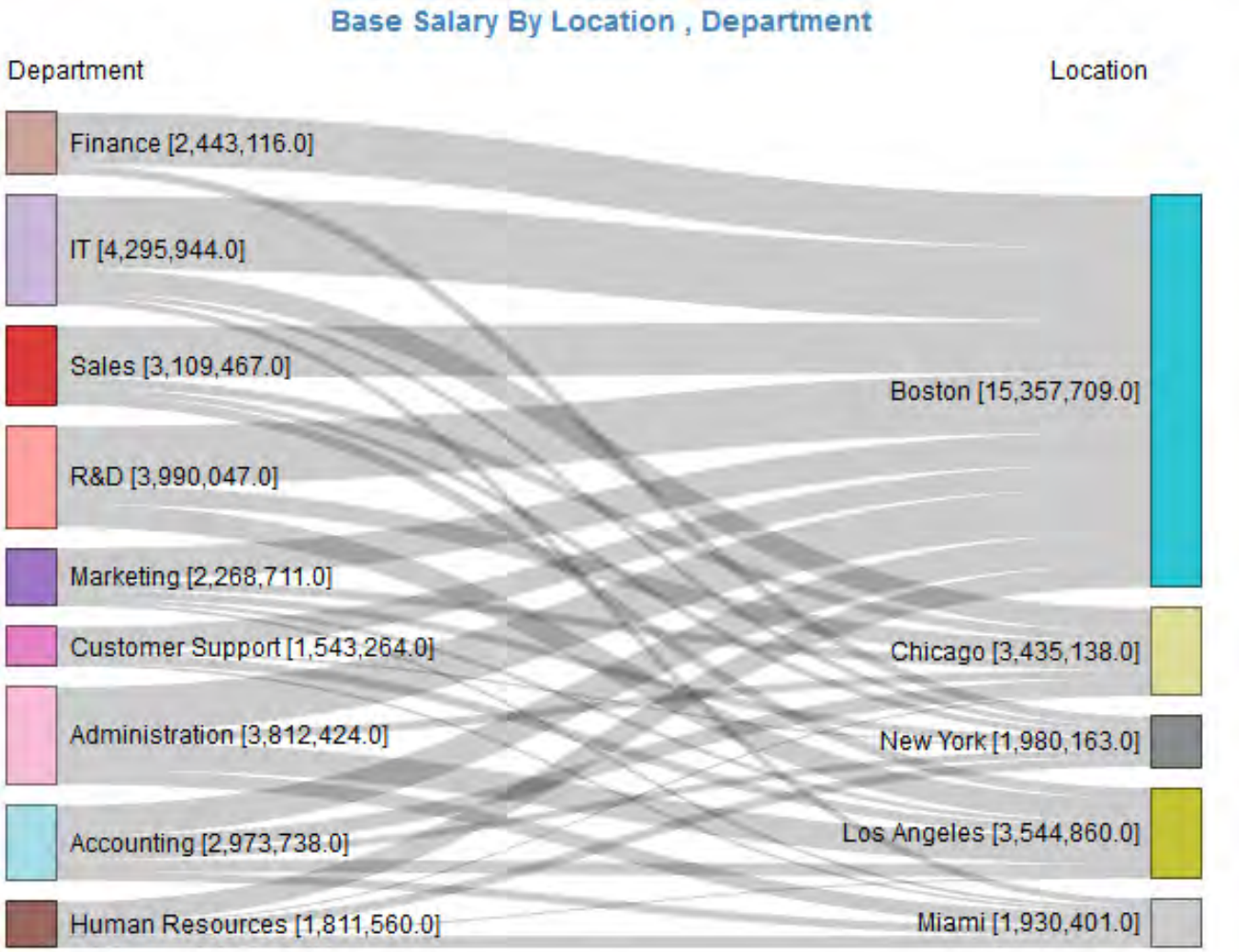
Примеры визуализации

- Плоское дерево



Примеры визуализации

- Диаграмма Санкей

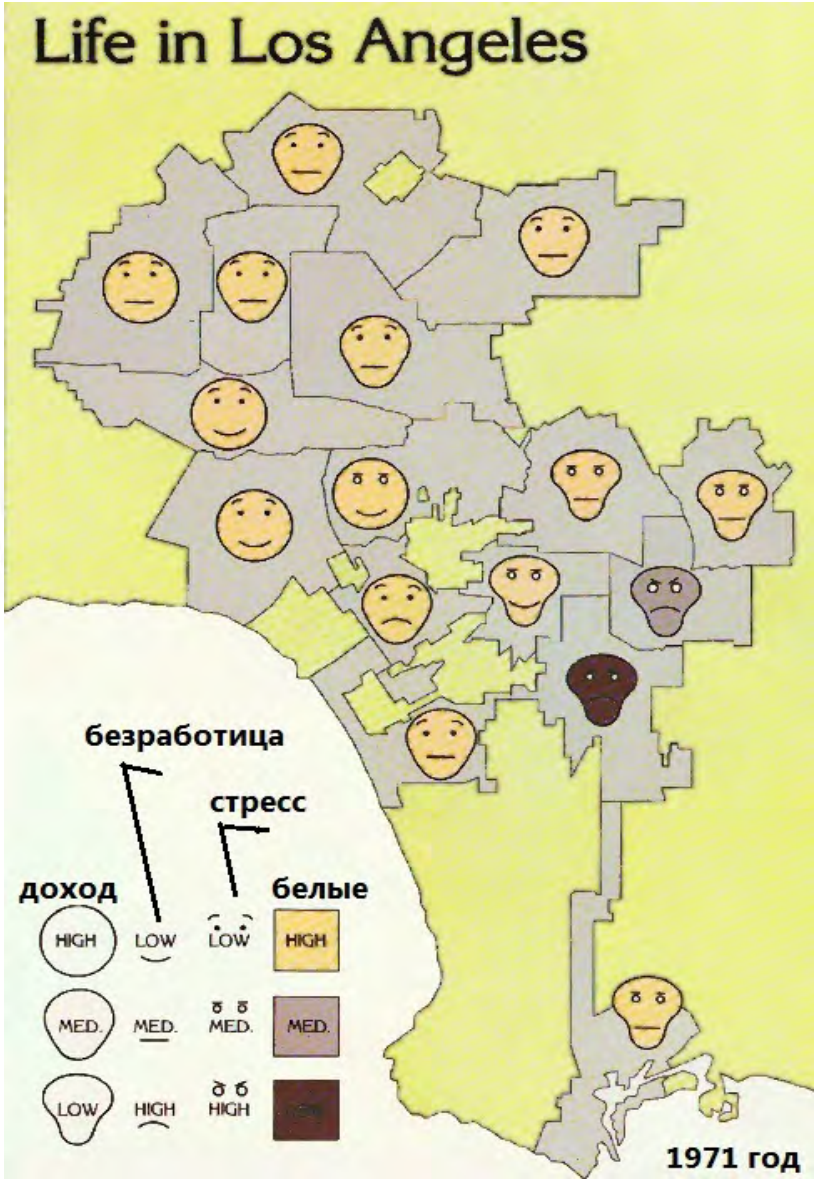


Примеры визуализации

- Картограмма

Примеры визуализации

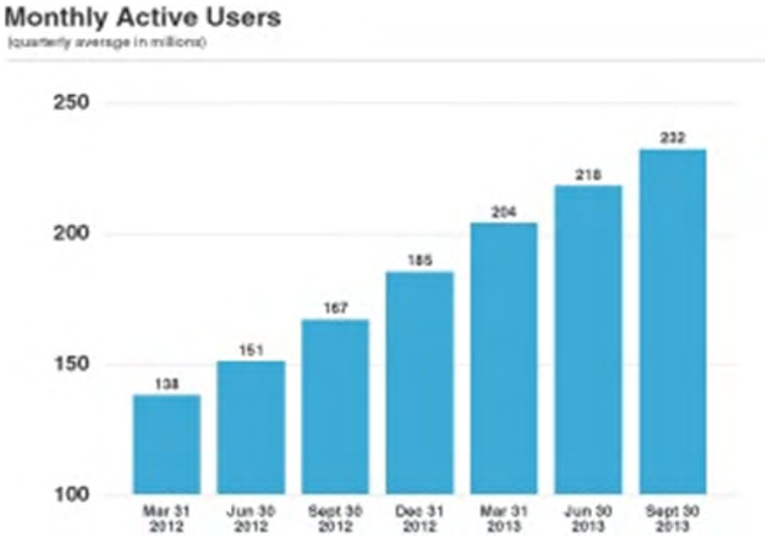
- Лица Чернова

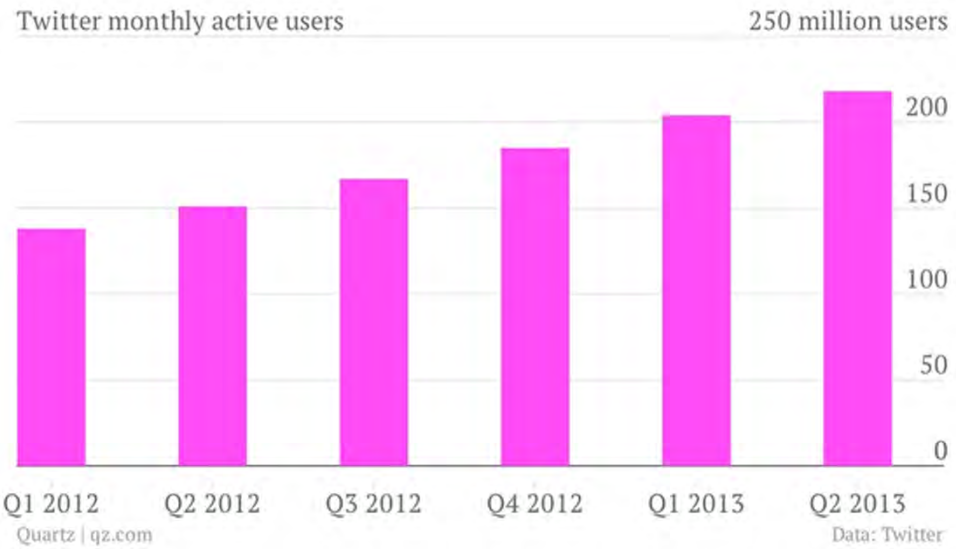


Ошибки визуализации

- **Начинать ось *y* не с нуля**

График пользователей твиттера



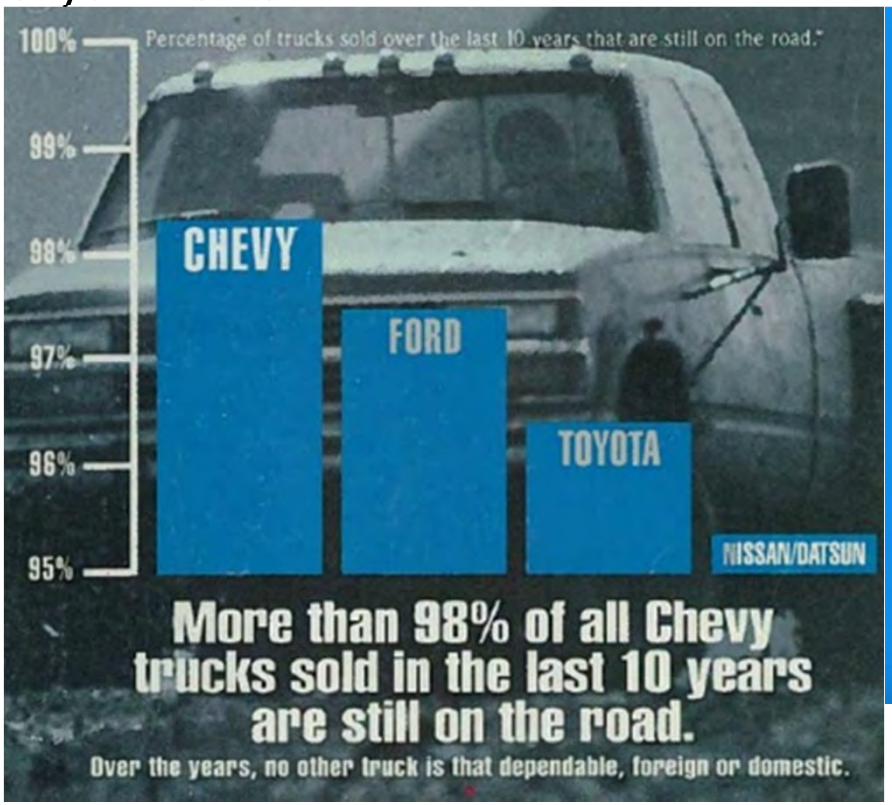


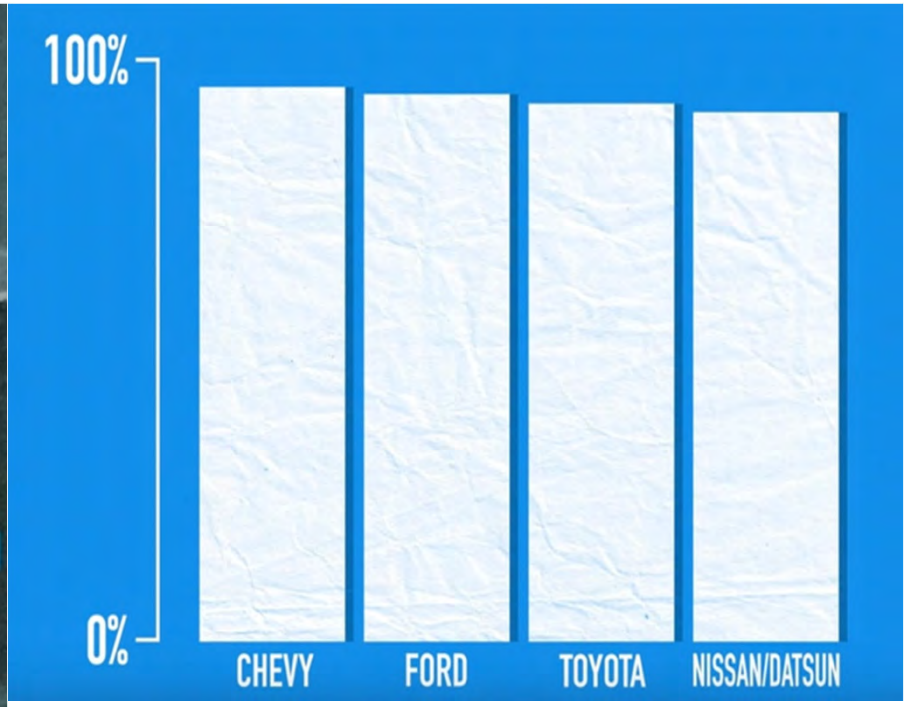
Последний столбик выше первого в 3 раза!

Ошибки визуализации

- **Начинать ось *y* не с нуля**

Chevrolet хвалится тем, сколько их внедорожников всё ещё работают, спустя 10 лет



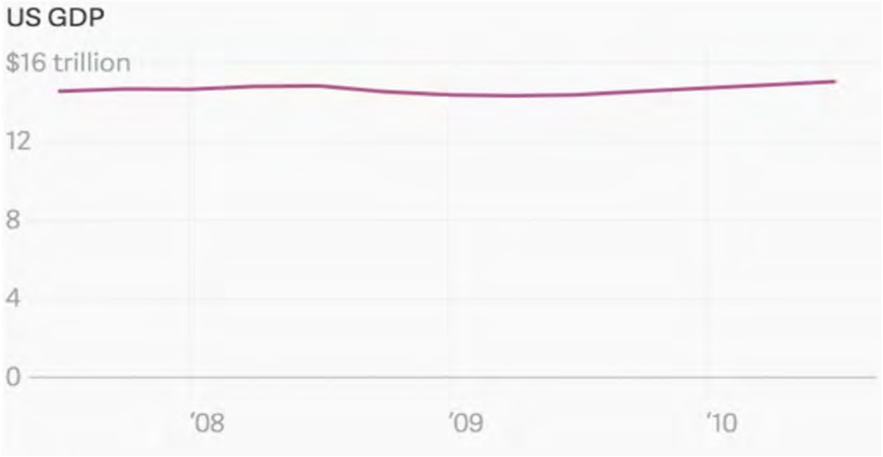


Она начинается с 95%!

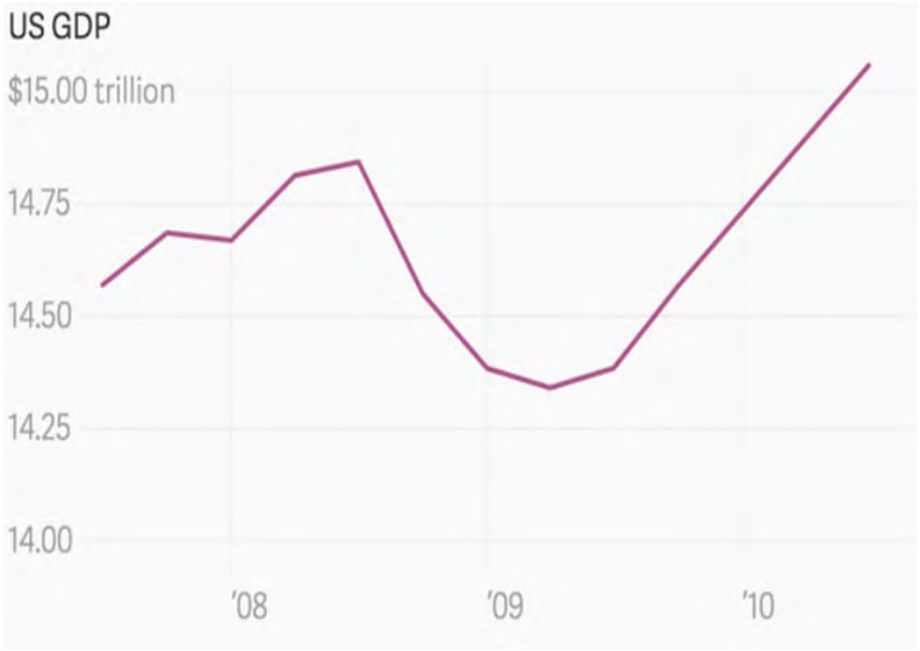
Ошибки визуализации

- **Иногда начинать отсчёт не с нуля — нормально**

Если в графике есть какая-то *временная зависимость*, то есть мы хотим посмотреть изменение параметра за какой-то срок, гораздо информативнее будет начать отсчёт не с нуля! Иначе мы можем вообще не увидеть изменений. Так, например, выглядит мировой финансовый кризис, если показать полный график:



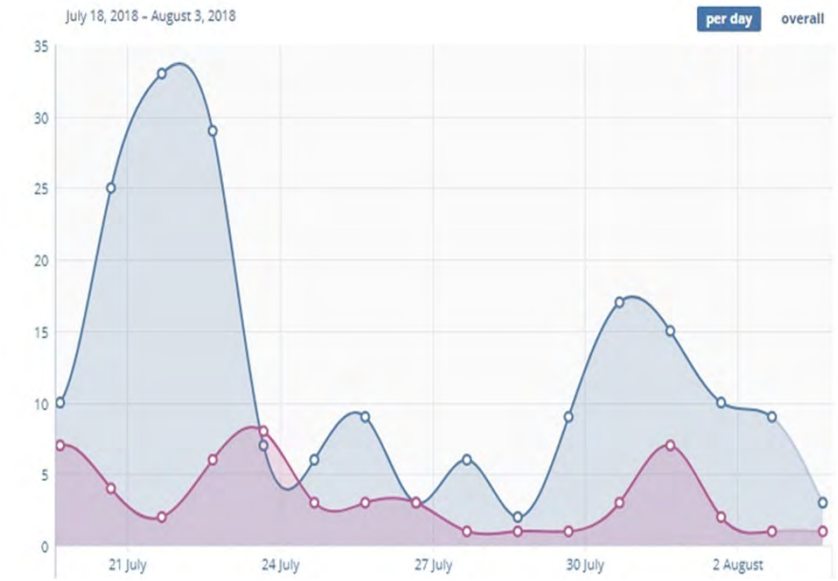
Кризиса – не видно

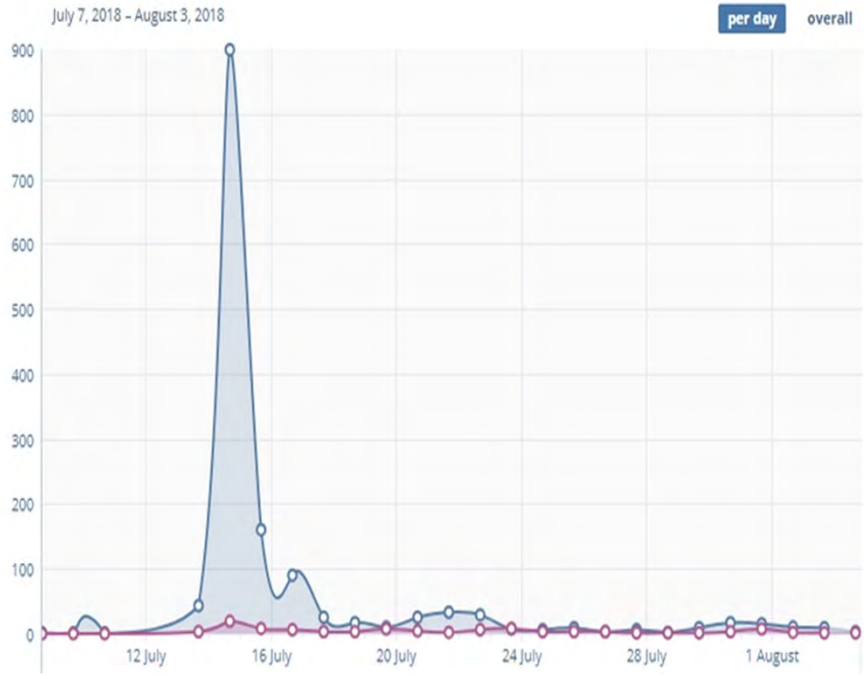


Ошибки визуализации

- **Имеет смысл брать не весь доступный временной отрезок, а лишь его актуальную часть**

Если смотреть на график подписчиков, явно видно, когда выходили посты и насколько они были успешны:



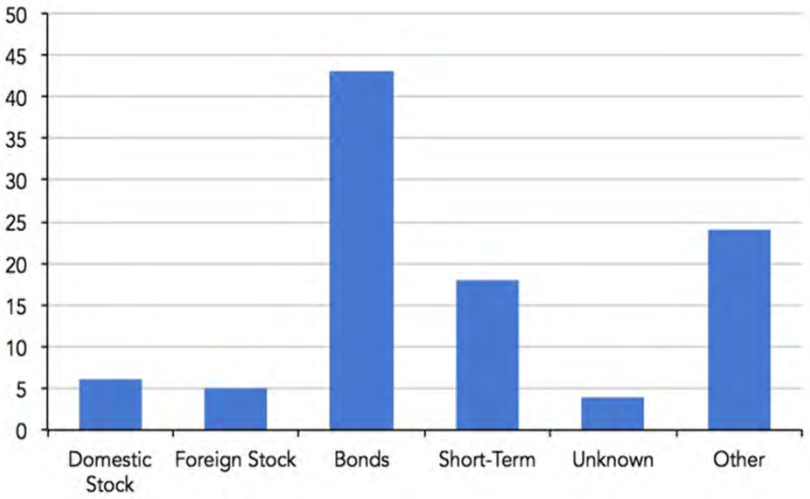


Но если включить во временной отрезок самый успешный пост, его величина сведёт эти колебания на нет!

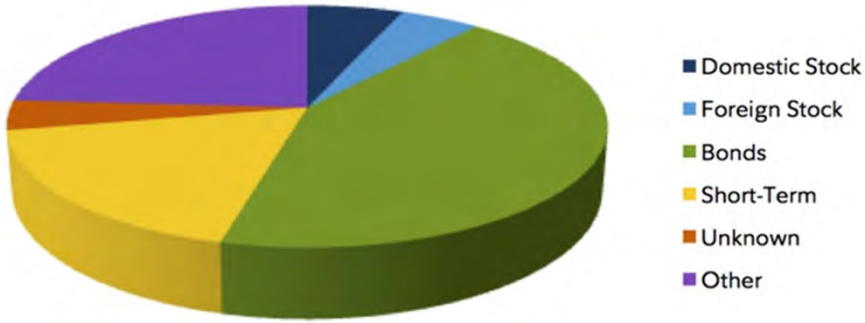
Ошибки визуализации

- **Неподходящие графики**

Такой график позволит легко понять соотношение данных



Из-за наклона соотношения площадей искажаются. Информация воспринимается гораздо хуже



Ошибки визуализации

Используйте графики правильно: