# 02 - Управление данными в ИИ

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


## Описание:

CRISP-DM — наиболее распространённая методология исследования данных, применяемая в проектах машинного обучения и анализа данных. Процесс ориентирован на итеративный подход и состоит из нескольких этапов

## Этапы процесса:

1. **Business Understanding (Понимание бизнеса):**
   - Определение целей и задач проекта с точки зрения бизнеса
   - Определение ключевых метрик успеха
2. **Data Understanding (Понимание данных):**
   - Сбор данных и их предварительный анализ
   - Оценка качества данных, выявление пропусков и аномалий
3. **Data Preparation (Подготовка данных):**
   - Очистка данных, преобразование форматов, создание новых признаков
   - Этот этап занимает **40–80% времени проекта**
4. **Modeling (Моделирование):**
   - Выбор алгоритмов машинного обучения.
   - Обучение и настройка моделей на подготовленных данных
5. **Evaluation (Оценка):**
   - Проверка качества модели
   - Сравнение результатов с бизнес-целями
6. **Deployment (Внедрение):**
   - Интеграция модели в рабочую среду
   - Использование модели для реального принятия решений

## Датасентричный взгляд на моделирование:

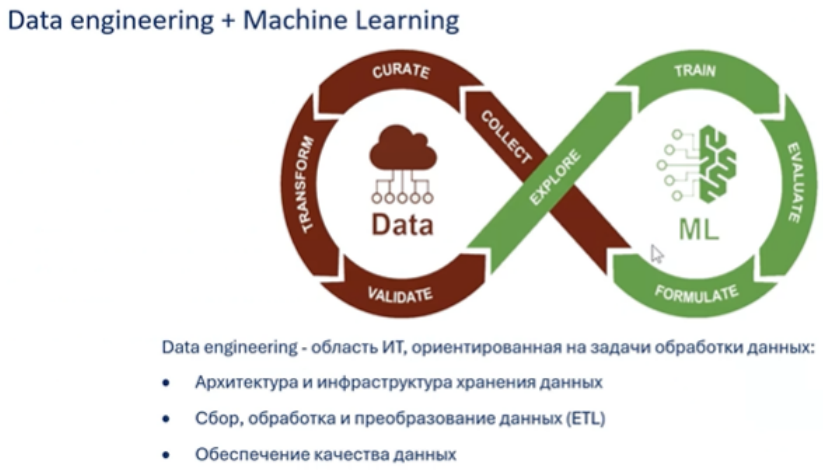
- **Данные — ключевой ресурс в ML:** Их часто называют "новой нефтью".

- Наличие данных необходимого качества, объема и разрезов определяет успех проекта.
- Большая часть времени уходит на сбор, понимание и подготовку данных.

---

**Итог:** CRISP-DM помогает структурировать процесс работы с данными и моделями, фокусируясь на достижении бизнес-целей через понимание, подготовку и анализ данных. Это универсальный подход для успешной реализации проектов машинного обучения

# Data Engineering + Machine Learning




## Data Engineering:

Data engineering — это область ИТ, ориентированная на задачи обработки данных. Включает:

1. **Архитектуру и инфраструктуру хранения данных:**
   - Создание устойчивых систем для обработки больших объемов данных
   - Использование хранилищ данных (Data Warehouses) и озер данных (Data Lakes)
2. **Сбор, обработку и преобразование данных (ETL):**
   - Экстракция данных из различных источников (Extract)
   - Преобразование данных в нужный формат (Transform)
   - Загрузка в хранилища или базы данных (Load)
3. **Обеспечение качества данных:**
   - Проверка и валидация данных для устранения ошибок и пропусков
   - Удаление дубликатов и аномалий

## Machine Learning:

Machine learning включает этапы:

1. **Train (Обучение):**
   - Использование данных для построения модели
2. **Formulate (Формулирование задачи):**
   - Определение бизнес-проблемы и преобразование её в ML-задачу
3. **Evaluate (Оценка):**
   - Проверка производительности модели с использованием метрик (например, точность, F1-скор)

## Связь Data Engineering и ML:

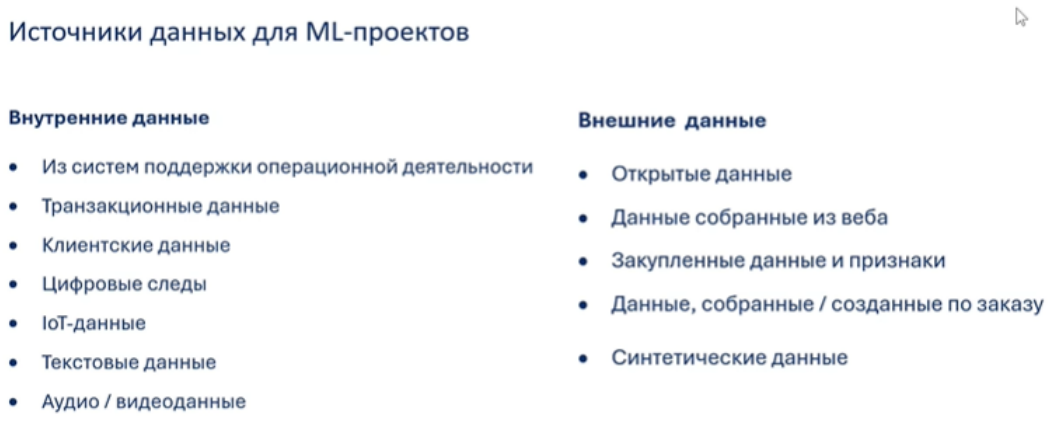
- **Pipeline данных:**
  - Data Engineering отвечает за подготовку и структурирование данных, которые затем используются для обучения ML-моделей
- **Подготовка данных:**
  - Сбор, валидация, очистка и анализ данных перед подачей их в модель
- **Обратная связь:**
  - Результаты ML-модели могут повлиять на улучшение процессов Data Engineering (например, выявление проблем качества данных)

## Ключевые этапы на диаграмме:

1. **Data (обработка данных):**
   - Transform → Validate → Curate → Collect → Explore
2. **ML (машинное обучение):**
   - Explore → Formulate → Train → Evaluate

**Итог:** эффективная работа Data Engineering обеспечивает успех ML-моделей, так как качество входных данных определяет результативность и точность предсказаний. Эта синергия становится ключом к решению сложных бизнес-задач

# Источники данных для ML-проектов




## Внутренние данные:

1. **Из систем поддержки операционной деятельности:**
   - ERP-системы (SAP, Oracle), CRM-системы (Salesforce, HubSpot)
   - Данные о бизнес-процессах
2. **Транзакционные данные:**
   - Финансовые операции, чеки, продажи
3. **Клиентские данные:**
   - Личные данные клиентов, профили, истории покупок
4. **Цифровые следы:**
   - Логи веб-сайтов, события приложений, активность пользователей
5. **IoT-данные:**
   - Сведения с датчиков, умных устройств, оборудования
6. **Текстовые данные:**
   - Электронные письма, сообщения в чатах, документы

7. **Аудио/видеоданные:**
    - Звонки в кол-центры, видеозаписи для анализа

## Внешние данные:

1. **Открытые данные:**
    - Госстатистика, отчеты, данные научных исследований
    - Примеры: данные Росстата, Kaggle, UCI Machine Learning Repository
2. **Данные, собранные из веба:**
    - Парсинг сайтов, API публичных сервисов
    - Примеры: отзывы на продукцию, данные социальных сетей
3. **Закупленные данные и признаки:**
    - Данные от партнёров или специализированных компаний
    - Например, маркетинговая информация, демографические данные
4. **Данные, собранные/созданные по заказу:**
    - Пользовательские опросы, исследования
5. **Синтетические данные:**
    - Искусственно созданные данные для тестирования или обучения моделей
    - Используются, когда реальные данные недоступны или их недостаточно

**Итог:** для ML-проектов важно комбинировать **внутренние данные** (бизнес и операционные процессы) и **внешние данные** (контекст и дополнительная информация). Это позволяет создать более точные и полезные модели для реального применения

# Принцип GIGO: Garbage In — Garbage Out


## Суть принципа GIGO:

- Если на вход системы анализа или модели подаются **некачественные или нерелевантные данные** (garbage in), то на выходе результат также будет низкого качества (garbage out)
- Качество входных данных определяет точность и полезность результата

## Ключевые элементы:

1. **Garbage Data In (Мусорные данные на входе):**
    - Проблемы:
        - Ошибки в данных (например, пропуски, выбросы, некорректные форматы)
        - Неверные источники данных.

    - Отсутствие релевантности данных для задачи
2. **Analysis Pipeline (Процесс анализа):**
    - Анализ или обработка данных
    - Даже с идеально настроенным пайплайном, если входные данные плохие, результат будет некорректным
3. **Garbage Data Out (Мусорные данные на выходе):**
    - Низкая точность модели или некорректные аналитические выводы.
    - Ошибочные бизнес-решения, основанные на неверных данных

## Почему качество данных важно:

1. **Снижение ошибок:**
    - Гарантия, что данные соответствуют задачам анализа
2. **Повышение производительности моделей:**
    - Чистые и релевантные данные позволяют ML-моделям обучаться эффективно
3. **Оптимизация ресурсов:**
    - Предотвращает затраты времени и ресурсов на переработку некорректных данных

## Решения для обеспечения качества данных:

1. **Очистка данных:**
    - Удаление пропусков, дубликатов, аномалий
2. **Валидация данных:**
    - Проверка корректности, полноты и актуальности
3. **Автоматизация обработки:**
    - Использование ETL-инструментов (например, Apache NiFi, Talend) для стандартизации данных
4. **Мониторинг данных:**
    - Постоянный контроль качества поступающих данных

**Итог:** принцип GIGO подчеркивает важность работы с качественными данными на этапе подготовки. Это основной фактор успеха как аналитических систем, так и ML-проектов

# Корпоративная функция управления данными (Data Governance)


## Что такое Data Governance?

**Управление данными (Data Governance)** — это функция, направленная на эффективное использование данных как стратегического актива организации

## Цели управления данными:

1. **Улучшение доступности данных:**
   - Гарантия, что данные доступны для нужных людей в нужное время
2. **Обеспечение качества данных:**
   - Работа над корректностью, полнотой, актуальностью и консистентностью данных
3. **Снижение рисков при использовании данных:**
   - Управление доступом, защита конфиденциальных данных.
   - Соблюдение нормативных требований
4. **Снижение издержек при использовании данных:**
   - Оптимизация процессов хранения и обработки данных
   - Исключение дублирования и избыточности

## Структура Data Governance:

1. **Стратегия:**
   - Определение роли данных в достижении бизнес-целей
   - Управление и реализация корпоративной политики по данным
2. **Операционные процессы:**
   - Включают:
     - Роли и ответственность (например, CDO — Chief Data Officer).
     - Процессы очистки, интеграции, управления метаданными.
     - Мониторинг и контроль использования данных
3. **Технологии:**
   - Использование инструментов для:
     - Хранения данных (Data Warehouses, Data Lakes)
     - Управления метаданными (например, Collibra, Alation)
     - Обеспечения безопасности данных.
     - Автоматизации процессов управления
4. **Методология Data Governance:**
   - Поддержка стандартов и процедур управления данными

## Роль офиса CDO:

**Chief Data Officer (CDO)**:

- Ответственный за внедрение и реализацию управления данными
- Обеспечивает согласование стратегий и технологий для эффективного использования данных

## Значение для ML-проектов:

Реализованная функция управления данными создает основу для успешной реализации ML-проектов, обеспечивая:

- Качественные и доступные данные для анализа
- Стандартизированные процессы и технологии

**Итог:** Data Governance обеспечивает стратегический подход к работе с данными, помогая бизнесу принимать информированные решения и снижать риски, а также создаёт фундамент для массового

внедрения ML-решений

# Фундаментальный взгляд: Иерархия DIKW(A)


## Иерархия DIKW(A):

Иерархия DIKW представляет собой концептуальную модель, описывающую процесс преобразования данных в осмысленные действия

1. **Data (Данные):**
   - Исходная, необработанная информация, собранная из различных источников
   - Этапы: сбор и организация данных
2. **Information (Информация):**
   - Обработанные данные, предоставляющие смысл или контекст
   - Этапы: суммирование и упорядочивание данных
3. **Knowledge (Знания):**
   - Осознание и понимание закономерностей и связей в данных
   - Этапы: анализ и выявление взаимосвязей
4. **Wisdom (Мудрость):**
   - Применение знаний для принятия решений и совершения действий
   - Этапы: синтез и принятие решений
5. **Action (Действие):**
   - Финальный этап, на котором знания и мудрость превращаются в конкретные действия для достижения целей

## Внедрение DIKW(A) в организацию:

1. **Ведение бизнес-процессов в ИТ-системах:**
   - Автоматизация и цифровизация процессов для повышения эффективности
2. **Интеграция данных методами управления данными:**
   - Организация работы с данными через стандартизацию и обеспечение их качества
3. **Аналитика и построение ИИ-моделей:**
   - Использование знаний, извлечённых из данных, для построения прогнозных моделей и оптимизации бизнес-решений

## Применение:

- DIKW(A) используется для цифровой трансформации бизнеса, улучшения управления данными и

построения стратегий на основе глубокого анализа

**Итог:** иерархия DIKW(A) формирует основу для понимания и использования данных в организации, способствуя переходу от простой обработки информации к осознанным действиям и стратегическим решениям

# Разведочный анализ данных (Exploratory Data Analysis, EDA)


## Цель EDA:

EDA направлен на глубокое понимание структуры, характеристик и особенностей данных перед применением сложных методов моделирования. Это ключевой этап подготовки данных в любом аналитическом проекте

## Основные шаги EDA:

1. **«Проникновение» в данные:**
   - Изучение состава, структуры и общего состояния данных
2. **Обнаружение отклонений и аномалий:**
   - Идентификация выбросов, пропусков или других несоответствий в данных
3. **Выбор наиболее важных переменных:**
   - Определение признаков, которые наиболее значимы для дальнейшего анализа или моделирования
4. **Понимание контекста данных:**
   - Исследование взаимосвязей между переменными и влияние внешних факторов
5. **Формулировка гипотез и их проверка:**
   - Построение первых предположений на основе данных и проверка их с использованием статистических методов
6. **Разработка начальных моделей и инсайты:**
   - Использование простых методов анализа для выявления закономерностей и формирования выводов

## Этапы EDA (на диаграмме):

1. **Understand the Problem & the Data:**
   - Определение цели анализа и изучение состава данных
2. **Import & Inspect the Data:**

- Загрузка данных и первичная проверка на корректность

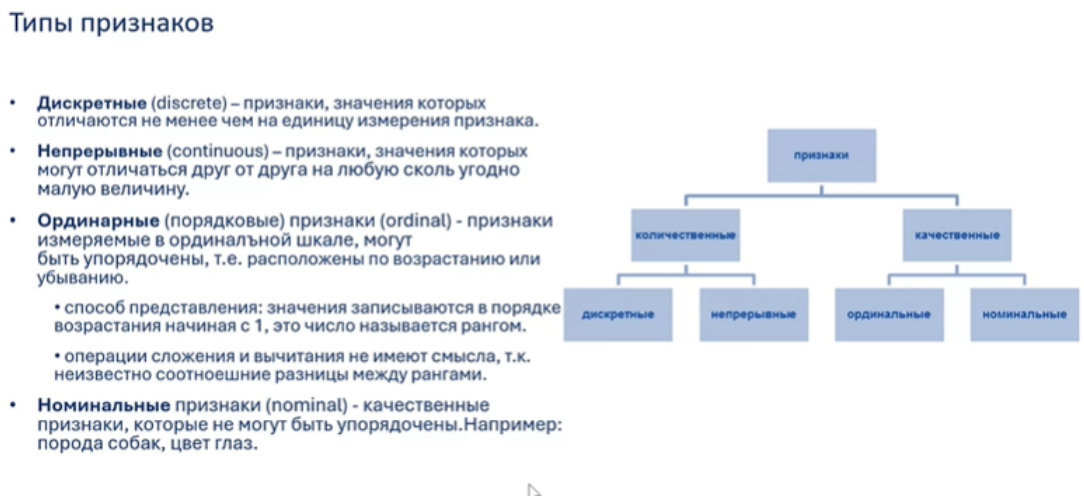
3. **Explore Data Characteristics:**
   - Анализ распределений, центральных тенденций, выбросов
4. **Perform Data Transformations:**
   - Очистка, нормализация, логарифмирование или масштабирование данных
5. **Visualize Data Relationships:**
   - Построение графиков для анализа взаимосвязей (корреляционные матрицы, scatter plots)
6. **Handle Outliers:**
   - Работа с выбросами: удаление или обработка
7. **Communicate Findings & Insights:**
   - Представление результатов анализа и выявленных закономерностей

## Результаты EDA:

- Понимание структуры данных
- Определение ключевых переменных
- Подготовка данных для моделирования
- Формирование первых гипотез и инсайтов

**Итог:** EDA — это фундаментальный этап в анализе данных, который обеспечивает качественную подготовку данных, понимание их особенностей и выявление ключевых закономерностей, необходимых для успешной реализации проекта

# Типы признаков в анализе данных




## Классификация признаков:

Признаки делятся на **количественные** и **качественные**, а внутри этих категорий — на подтипы

## Количественные признаки:

1. **Дискретные (discrete):**
   - Значения отличаются не менее чем на единицу измерения признака
   - Примеры:
     - Количество сотрудников, число автомобилей
2. **Непрерывные (continuous):**
   - Значения могут принимать любые значения в определённом диапазоне.
   - Примеры:

- Вес, рост, температура

## Качественные признаки:

1. **Ординальные (ordinal):**
   - Измеряются в **порядковой шкале**, имеют упорядоченность
   - Пример:
     - Уровень образования (начальное, среднее, высшее)
   - Особенности:
     - Значения записываются в порядке возрастания (ранги)
     - Операции сложения/вычитания не имеют смысла
2. **Номинальные (nominal):**
   - Категории не могут быть упорядочены.
   - Примеры:
     - Цвет глаз, порода собаки, город проживания
   - Особенности:
     - Никакая количественная интерпретация недопустима

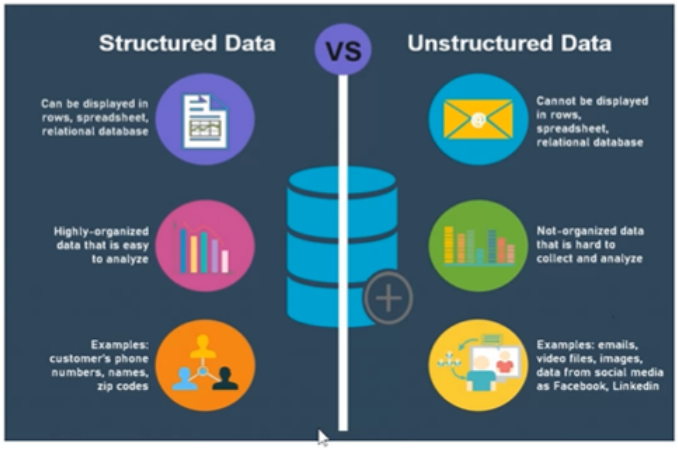
## Пример структуры:

- **Количественные:**
  - Дискретные: целые числа
  - Непрерывные: дробные значения
- **Качественные:**
  - Ординальные: могут быть упорядочены
  - Номинальные: только категории

**Итог:** правильное понимание типов признаков помогает выбирать подходящие методы анализа, обработки и визуализации данных, а также строить корректные модели машинного обучения

# Структурированные и неструктурированные данные

Структурированные и неструктурированные данные




## Структурированные данные (Structured Data):

- **Характеристики:**
  - Организованы в виде таблиц, строк, столбцов (например, в реляционных базах данных)
  - Высоко организованные данные, которые легко анализировать

- **Примеры:**
  - Номера телефонов клиентов, имена, почтовые индексы
  - Таблицы в базах данных, электронные таблицы
- **Преимущества:**
  - Легкость поиска, сортировки и анализа
  - Простота интеграции с аналитическими инструментами

## Неструктурированные данные (Unstructured Data):

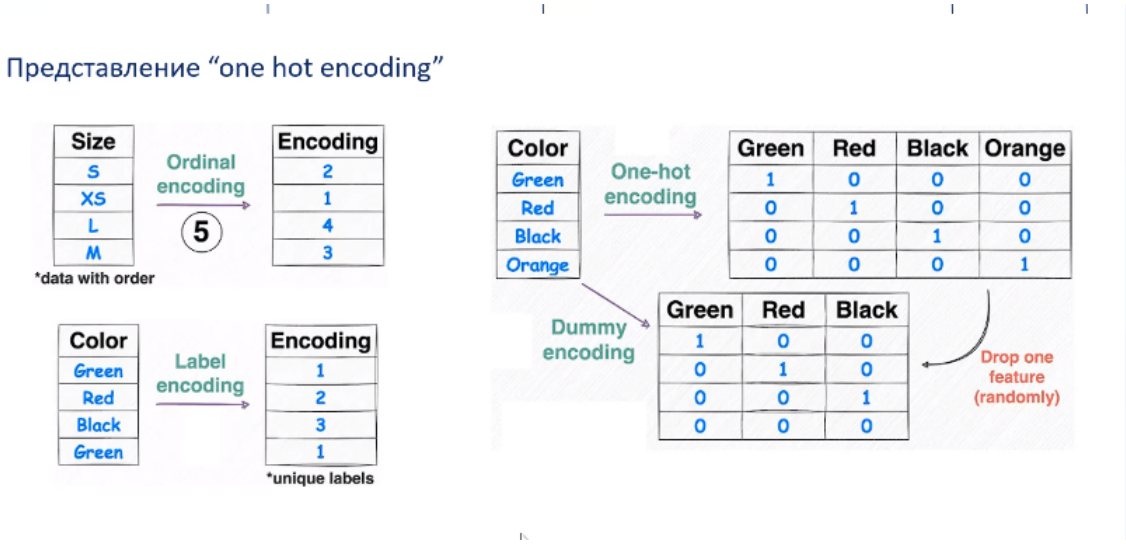
- **Характеристики:**
  - Не имеют строгой структуры и не могут быть представлены в таблицах или реляционных базах данных
  - Данные сложны для анализа и требуют специальных инструментов
- **Примеры:**
  - Электронные письма, видеофайлы, изображения
  - Данные из социальных сетей (Facebook, LinkedIn)
- **Проблемы:**
  - Сложность обработки и анализа
  - Требует специализированных решений, таких как машинное обучение, NLP (обработка естественного языка)

## Сравнение:

| Аспект | Структурированные данные | Неструктурированные данные |
|---|---|---|
| **Хранение** | Реляционные базы данных | Облачные хранилища, файловые системы |
| **Примеры** | Таблицы, записи CRM | Изображения, видео, социальные сети |
| **Анализ** | Простые аналитические инструменты | Нужны инструменты Big Data и AI |
| **Простота обработки** | Высокая | Низкая |

**Итог:** для эффективного использования данных в бизнесе и аналитике важно комбинировать структурированные данные (для быстрого анализа) и неструктурированные данные (для глубоких инсайтов с помощью современных технологий

# Представление категориальных признаков: One-Hot Encoding и другие методы



## 1. Ordinal Encoding:

- **Описание:**
  - Используется для признаков, которые имеют порядок
  - Каждое значение преобразуется в числовой код, сохраняя порядок
- **Пример:**
  - Размеры одежды: XS → 1, S → 2, M → 3, L → 4
- **Особенность:**
  - Подходит только для упорядоченных категорий

## 2. Label Encoding:

- **Описание:**
  - Каждой категории присваивается уникальное числовое значение
  - Не сохраняется информация о взаимосвязи категорий
- **Пример:**
  - Цвета: Green → 1, Red → 2, Black → 3
- **Проблема:**
  - Алгоритмы могут воспринимать числовые значения как ранжированные, что может привести к ошибкам в моделировании

## 3. One-Hot Encoding:

- **Описание:**
  - Каждая категория превращается в отдельную бинарную колонку
  - Значение "1" обозначает присутствие категории, "0" — её отсутствие
- **Пример:**
  - Цвета: Green → [1, 0, 0, 0], Red → [0, 1, 0, 0], Black → [0, 0, 1, 0], Orange → [0, 0, 0, 1]
- **Особенности:**
  - Подходит для всех категориальных признаков
  - Увеличивает размерность данных

## 4. Dummy Encoding:

- **Описание:**
  - Это вариация One-Hot Encoding, где одна из колонок опускается для предотвращения избыточности
  - Избегает проблемы **мультиколлинеарности**
- **Пример:**
  - Цвета: Green → [1, 0, 0], Red → [0, 1, 0], Black → [0, 0, 1], Orange → [0, 0, 0]

## Ключевые моменты:

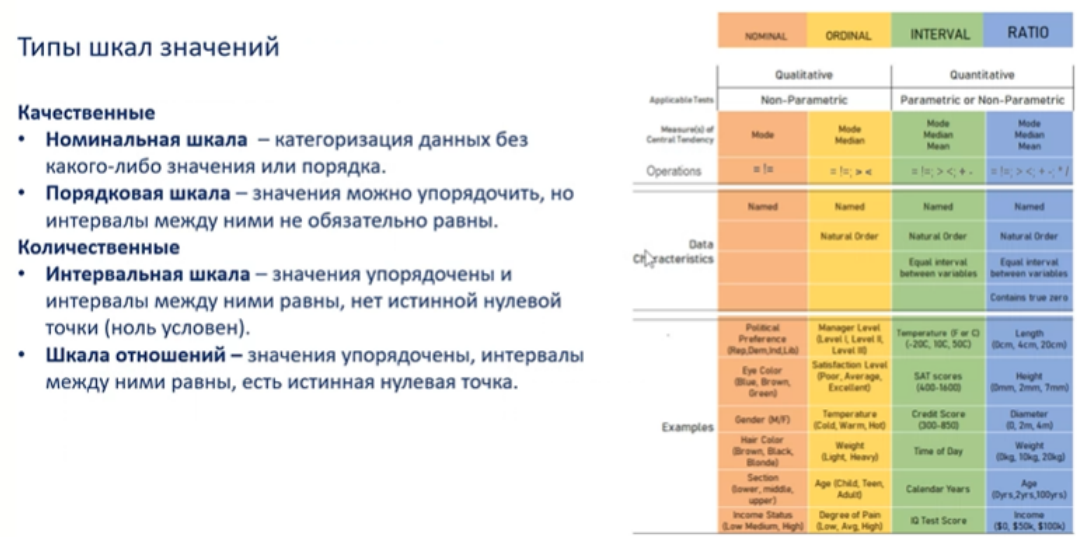
1. **Выбор метода:**
   - **Ordinal Encoding:** для упорядоченных данных
   - **One-Hot Encoding:** для неупорядоченных категорий, когда важна полная информация
   - **Dummy Encoding:** для моделей, чувствительных к мультиколлинеарности (например, линейные регрессии)
2. **Недостатки One-Hot Encoding:**
   - Увеличение количества столбцов при большом числе категорий

- Может потребоваться дополнительная оптимизация для больших данных

3. **Преимущества Dummy Encoding:**
   - Снижает размерность данных, сохраняя минимальную необходимую информацию

   **Итог:** правильный выбор метода кодирования категориальных данных зависит от задачи, типа признаков и модели. One-Hot Encoding остаётся универсальным и широко используемым, а Dummy Encoding помогает оптимизировать работу с данными в случае необходимости

# Типы шкал значений




## Качественные шкалы:

1. **Номинальная шкала (Nominal):**
   - Описание:
     - Категоризация данных без порядка или ранжирования
     - Категории равноправны и не имеют числового значения
   - Примеры:
     - Пол (мужской/женский), цвет глаз (синий, зелёный, коричневый)
   - Возможные операции:
     - Подсчёт, сравнение равенства
2. **Порядковая шкала (Ordinal):**
   - Описание:
     - Значения можно упорядочить, но интервалы между ними не равны
     - Не сохраняется количественная информация
   - Примеры:
     - Уровень удовлетворенности (низкий, средний, высокий), рейтинги
   - Возможные операции:
     - Сравнение: больше/меньше

## Количественные шкалы:

3. **Интервальная шкала (Interval):**
   - Описание:
     - Значения упорядочены, интервалы между ними равны, но нет истинной нулевой точки (ноль условен)
   - Примеры:
     - Температура (в градусах Цельсия или Фаренгейта), время на часах

- Возможные операции:
  - Сложение, вычитание, но не деление

4. **Шкала отношений (Ratio):**
   - Описание:
     - Значения упорядочены, интервалы равны, есть истинная нулевая точка
   - Примеры:
     - Рост, вес, возраст, доход
   - Возможные операции:
     - Все математические операции: сложение, вычитание, умножение, деление

## Сравнение шкал:

| Шкала | Пример | Операции | Характеристики |
|---|---|---|---|
| Номинальная | Пол, цвет глаз | Равенство, подсчёт | Нет порядка или интервалов |
| Порядковая | Рейтинг, уровень боли | Равенство, сравнение | Есть порядок, интервалы неравны |
| Интервальная | Температура, SAT | Сложение, вычитание | Равные интервалы, нет истинного нуля |
| Шкала отношений | Рост, вес, доход | Все операции | Равные интервалы, есть истинный ноль |

**Итог:** выбор шкалы измерения определяет доступные математические операции и методы анализа данных. Понимание различий между шкалами помогает правильно интерпретировать данные и использовать подходящие модели анализа

# Шаги разведочного анализа данных (Exploratory Data Analysis, EDA)


## Основные этапы EDA:

**Load Data (Загрузка данных):**
- Первичная загрузка данных из источников (базы данных, CSV-файлы и т.д.)
- Проверка структуры данных: форматы, типы данных, размер

**Step 1: Distinguish Attributes (Идентификация признаков):**
- Определение типов признаков (количественные, качественные)
- Разделение данных на независимые переменные (features) и целевую переменную (target)

**Step 2: Univariate Analysis (Одномерный анализ):**
- Анализ распределения отдельных признаков:
- Среднее, медиана, мода, стандартное отклонение
- Построение гистограмм и boxplot для анализа распределений
4 **Step 3: Bi-/Multivariate Analysis (Двумерный и многомерный анализ):**
-Исследование взаимосвязей между признаками:
-Корреляционные матрицы, scatter plots
- Выявление линейных и нелинейных зависимостей
**Step 4: Detect Aberrant and Missing Values (Поиск аномалий и пропусков):**
- Выявление пропусков данных и их обработка:
- Удаление, замена средним, медианой, прогнозирование
- Определение аномальных значений (например, невозможные отрицательные значения)
**Step 5: Detect Outliers (Поиск выбросов):**
- Анализ выбросов с помощью:
- IQR (межквартильный размах), Z-score
- Визуализация выбросов через boxplot
**Step 6: Feature Engineering (Создание признаков):**
- Преобразование и создание новых признаков:
- Нормализация, стандартизация
- Логарифмирование, создание полиномиальных признаков

## Результат:

- **Show Business Insights (Представление инсайтов):**
  - Обнаружение ключевых закономерностей и выводов
  - Подготовка данных для моделирования и визуализация инсайтов для бизнеса
  **Итог:** разведочный анализ данных позволяет глубоко понять структуру данных, выявить аномалии и скрытые закономерности, а также подготовить данные для построения моделей машинного обучения. Этот процесс является фундаментальным для успеха аналитического проекта

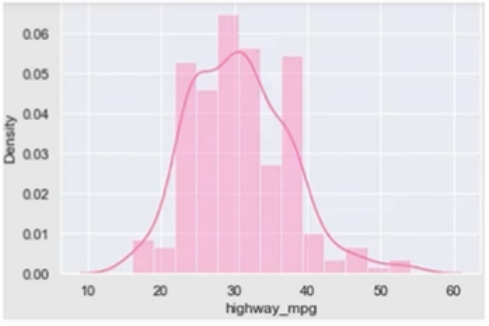
# Одномерный анализ (Univariate Analysis)

Одномерный анализ

Фокусируется на анализе одной переменной.

Цель: Понять распределение переменной, центральную тенденцию и разброс данных. :

- Описательная статистика (среднее значение, медиана, мода, дисперсия, стандартное отклонение)
- Визуализации (гистограммы, диаграммы размаха, столбчатые диаграммы, круговые диаграммы)
- Определение закона распределения, проверка на нормальность распределения
- Анализ выбросов и экстремальных значений




## Фокус:

Одномерный анализ сосредоточен на изучении одной переменной, чтобы понять её распределение, центральные тенденции и разброс данных

## Цель:

1. Понять, как распределяются значения переменной

2. Выявить центральную тенденцию (среднее, медиана, мода)
3. Определить наличие выбросов и аномалий

## Методы:

1. **Описательная статистика:**
   - Среднее значение (mean): показывает средний уровень значений
   - Медиана (median): центральное значение в отсортированных данных
   - Мода (mode): наиболее часто встречающееся значение
   - Дисперсия (variance): мера разброса данных относительно среднего
   - Стандартное отклонение (standard deviation): корень из дисперсии, показывает разброс данных
2. **Визуализация:**
   - Гистограммы: анализ распределения значений
   - Диаграммы размаха (boxplot): для выявления выбросов
   - Столбчатые диаграммы: для категориальных данных
   - Круговые диаграммы: для долей категориальных переменных
3. **Анализ закона распределения:**
   - Проверка на нормальность распределения с использованием тестов:
     - Тест Шапиро-Уилка
     - Q-Q plot (график квантилей)
4. **Анализ выбросов:**
   - Определение экстремальных значений, которые сильно отклоняются от основного набора данных
   - Методы:
     - Межквартильный размах (IQR)
     - Z-оценка (Z-score)

## На графике:

- Гистограмма с плотностью распределения демонстрирует, как значения переменной ( `highway_mpg` ) распределены по диапазону
- Выявлены пики, соответствующие часто встречающимся значениям

**Итог:** одномерный анализ помогает быстро понять базовые характеристики переменной, выявить проблемы с данными (например, выбросы) и подготовить их к дальнейшему многомерному анализу или моделированию

# Анализ и заполнение пропущенных значений

# 1. Проблема пропущенных значений:

- **Пропущенные значения (Missing Values):**
  - Возникают в данных из-за отсутствия информации, ошибок сбора данных или неправильной записи
  - Пропуски могут искажать распределение данных и снижать точность моделей машинного обучения

# 2. Методы заполнения пропущенных значений:

## a. Удаление пропусков (Dropping Missing Values):

- Удаляются строки или столбцы с пропущенными значениями
- **Плюсы:** простота реализации
- **Минусы:** потеря информации, особенно при большом объёме пропусков

## b. Заполнение фиксированными значениями:

- **Zero Imputation (Замена на 0):**
  - Все пропуски заменяются на 0
  - **Плюсы:** простота и понятность
  - **Минусы:** может сильно искажать распределение

## c. Заполнение статистическими значениями:

- **Mean Imputation (Среднее):**
  - Пропуски заменяются средним значением колонки
  - **Плюсы:** сохраняется общее распределение данных
  - **Минусы:** уменьшается вариативность, не учитывается структура данных
- **Median Imputation (Медиана):**
  - Замена на медианное значение для устойчивости к выбросам
- **Mode Imputation (Мода):**
  - Для категориальных данных пропуски заполняются наиболее частым значением

## d. Алгоритмические методы:

- **KNN Imputation:**
  - Заполнение на основе схожих наблюдений (ближайших соседей)
- **Регрессионное заполнение:**
  - Пропуски предсказываются с помощью других переменных

## e. Продвинутые методы:

- **Множественная иммутация (Multiple Imputation):**
  - Генерируются несколько возможных значений для пропусков с учетом их вероятности
- **Использование моделей:**
  - Пропуски заполняются с использованием ML-алгоритмов

# 3. На графиках:

- **Исходное распределение:**
  - Среднее (mean) = 2.08, стандартное отклонение (std.) = 1.95

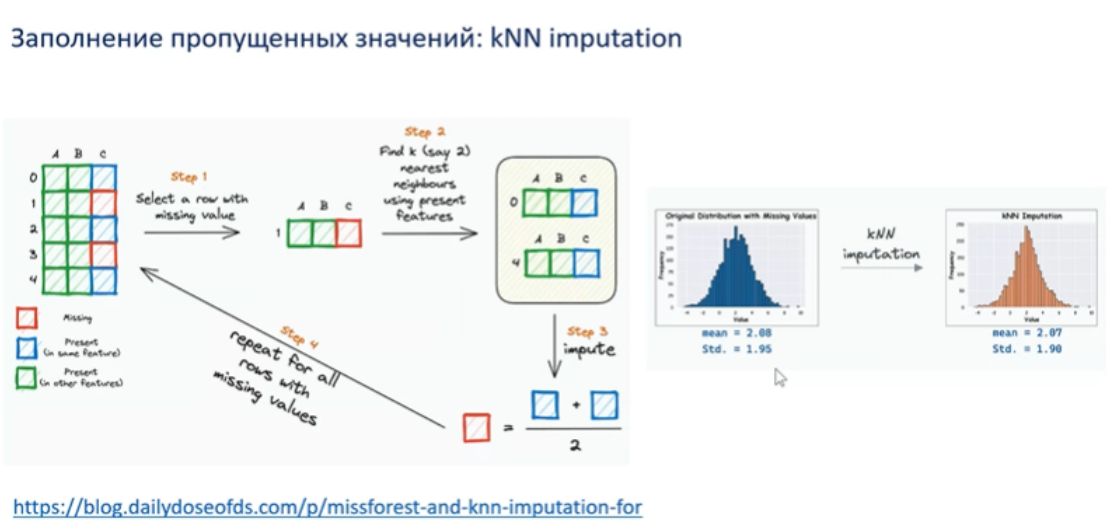
- **Zero Imputation:**
  - Среднее уменьшилось (mean = 1.66), стандартное отклонение практически не изменилось
- **Mean Imputation:**
  - Среднее сохранилось (mean = 2.08), но уменьшилось стандартное отклонение (std. = 1.76)

## 4. Выбор метода:

- Зависит от:
  - Количества пропусков
  - Характера данных (числовые, категориальные)
  - Влияния пропусков на результаты анализа

**Итог:** заполнение пропущенных значений — критически важный этап анализа данных. От выбора метода заполнения зависит, насколько точно модель сможет интерпретировать данные и делать прогнозы

# Заполнение пропущенных значений: kNN Imputation




https://blog.dailydoseofds.com/p/missforest-and-knn-imputation-for

## Суть метода kNN Imputation:

- Используется алгоритм ближайших соседей (k-Nearest Neighbors), чтобы заполнить пропущенные значения, основываясь на схожести с другими строками (наблюдениями)
- Пропуски заполняются средним (или медианой) значений ближайших соседей

## Шаги алгоритма:

**Step 1:** Выберите строку с пропущенным значением.
- Определите, какие значения в строке отсутствуют, а какие присутствуют
**Step 2:** Найдите k ближайших соседей
- Рассчитайте расстояние (например, Евклидово расстояние) между строками, используя только доступные значения
**Step 3:** Заполните пропущенное значение
- Пропуск заменяется средним (или медианой) значений признака среди k ближайших соседей
**Step 4:** Повторите процесс
- Для всех строк с пропущенными значениями алгоритм выполняется повторно, пока все пропуски не будут заполнены

## Преимущества kNN Imputation:

- **Сохраняет структуру данных:**
  - Заполнение выполняется на основе наблюдений с аналогичными характеристиками
- **Подходит для числовых и категориальных данных.**
- **Гибкость:** возможность настройки числа соседей (k) и метода расчета расстояний

## Недостатки:

- **Высокая вычислительная сложность:**
  - При большом объёме данных алгоритм может быть медленным
- **Чувствительность к выбору параметров:**
  - Число соседей (k) и метрика расстояния влияют на качество заполнения
- **Не всегда корректен при наличии большого числа выбросов**

## На графиках:

- **Исходное распределение:**
  - Среднее (mean) = 2.08, стандартное отклонение (std.) = 1.95
- **После применения kNN Imputation:**
  - Среднее практически не изменилось (mean = 2.07), стандартное отклонение уменьшилось (std. = 1.90), что показывает сохранение структуры данных

**Вывод:** kNN Imputation — мощный метод для заполнения пропущенных значений, который подходит для случаев, где данные имеют внутренние закономерности. Его использование особенно полезно для небольших и средних наборов данных с минимальными выбросами

# Двухмерный и многомерный анализ


## Определение:

- **Двухмерный анализ (Bivariate Analysis):**
  - Изучает взаимосвязь между двумя переменными
- **Многомерный анализ (Multivariate Analysis):**
  - Анализирует взаимодействие более чем двух переменных для выявления сложных взаимосвязей

## Цели анализа:

1. Понять, как одна переменная влияет на другую
2. Выявить зависимости или взаимосвязи между переменными

3. Оценить сложные закономерности в данных

## Методы и инструменты:

1. **Графические методы:**
   - **Диаграммы рассеяния (Scatter Plots):**
     - Используются для визуализации зависимости между двумя числовыми переменными
   - **Парные графики (Pair Plots):**
     - Отображают взаимосвязь нескольких переменных в виде набора диаграмм рассеяния
   - **Графики параллельных координат:**
     - Используются для анализа многомерных данных
2. **Корреляционный анализ:**
   - **Корреляционная матрица:**
     - Показатели линейной зависимости между переменными (например, коэффициент Пирсона)
   - **Тепловые карты (Heatmaps):**
     - Визуализация корреляций между переменными
3. **Анализ категориальных переменных:**
   - **Хи-квадрат тест:**
     - Оценка связи между категориальными переменными
4. **Методы снижения размерности:**
   - **PCA (Principal Component Analysis):**
     - Преобразование данных в набор главных компонент для уменьшения размерности
   - **t-SNE:**
     - Нелинейное снижение размерности для визуализации сложных данных
   - **Кластерный анализ:**
     - Группировка данных на основе их схожести

## На графике (справа):

- Пример **парных графиков (Pair Plots):**
  - На диагонали показаны распределения переменных
  - Вне диагонали — диаграммы рассеяния, демонстрирующие взаимосвязи между переменными

## Результаты анализа:

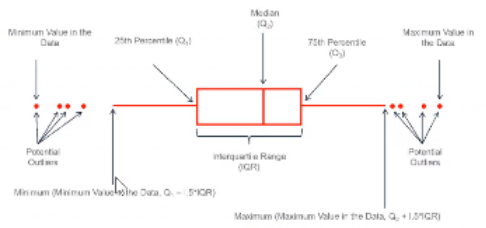
1. Выявление значимых взаимосвязей между переменными
2. Понимание структуры и закономерностей в данных
3. Определение переменных, которые следует использовать в моделировании

**Итог:** двухмерный и многомерный анализ являются мощными инструментами для изучения зависимостей и выявления закономерностей, которые могут быть неочевидны при одновременном анализе только одной переменной. Эти методы особенно важны для подготовки данных к построению моделей машинного обучения

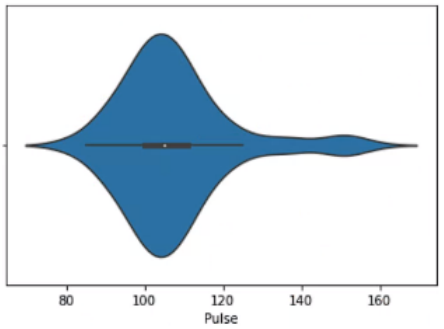
# Выявление выбросов (Outliers Detection)

Выявление выбросов (outliers)

Box-plot




Violinplot




# 1. Что такое выбросы?

- **Выбросы** — это данные, которые значительно отличаются от остальных значений набора данных
- Могут возникать из-за ошибок измерений, особенностей системы или редких, но значимых событий

# 2. Методы визуализации выбросов:

## a. Box-Plot (Ящичковая диаграмма):

- **Описание:**
  - Отображает ключевые статистики: медиану, квартили и потенциальные выбросы
- **Элементы:**
  - **Q1 (25-й перцентиль):** Нижняя граница ящика
  - **Q3 (75-й перцентиль):** Верхняя граница ящика
  - **IQR (межквартильный размах):** Разница между Q3 и Q1
  - **Усы:** Границы, выходящие за пределы IQR (обычно на 1.5 × IQR от Q1 и Q3)
  - **Точки за пределами усов:** Потенциальные выбросы
- **Применение:**
  - Быстрый способ выявления выбросов

## b. Violin-Plot (Виолончельная диаграмма):

- **Описание:**
  - Сочетает функции box-plot и плотности распределения
- **Элементы:**
  - Показаны медиана и межквартильный размах
  - Гладкая форма демонстрирует распределение данных
- **Применение:**
  - Используется для понимания распределения данных и выявления выбросов
  - Подходит для сравнения нескольких групп

# 3. Как интерпретировать?

## Box-Plot:

- Значения за пределами "усов" — это выбросы.
- Пример:
  - Влево или вправо от границ IQR можно увидеть точки, представляющие выбросы

### Violin-Plot:

- Ширина "виолончели" показывает плотность данных
- Узкие участки указывают на низкую плотность, а аномальные пики — на выбросы

## 4. Выводы:

1. Box-plot проще для базового анализа выбросов
2. Violin-plot даёт более полную картину распределения данных

**Итог:** для выявления выбросов лучше комбинировать методы визуализации. Box-plot помогает быстро найти выбросы, а Violin-plot предоставляет контекст распределения, что особенно важно для сложных наборов данных

# Основные подходы к созданию признаков (Feature Engineering)


## 1. Feature Engineering:

Процесс преобразования данных в формат, подходящий для анализа и моделирования, включает создание, преобразование и отбор признаков

## 2. Основные подходы:

### a. Data-Driven Feature Engineering (Основанный на данных):

1. **Domain Knowledge (Знание предметной области):**
   - Использование знаний об области анализа для создания новых признаков
   - Пример: расчёт возраста пользователя по дате рождения
2. **Data Munging (Обработка данных):**
   - Очистка, преобразование и подготовка данных
   - Пример: исправление ошибок в данных, удаление пропусков
3. **Exploratory Data Analysis (EDA):**
   - Использование визуализации и анализа для выявления скрытых закономерностей
   - Пример: создание признаков на основе корреляций

### b. Technical Feature Engineering (Технический):

1. **Text Vectorization (Векторизация текста):**

- Преобразование текстовых данных в числовой формат
- Пример: TF-IDF, Bag-of-Words

2. **Binning & Categorization (Категоризация):**
   - Группировка числовых данных в интервалы
   - Пример: преобразование возраста в категории (молодой, взрослый, пожилой)
3. **Data Standardization (Стандартизация данных):**
   - Приведение данных к единому масштабу
   - Пример: нормализация (Min-Max Scaling)

## 3. Техники Feature Engineering:

1. **Binning:**
   - Преобразование непрерывных данных в интервалы
   - Пример: разделение дохода на категории
2. **Log Transform (Логарифмическое преобразование):**
   - Уменьшение влияния выбросов
   - Пример: логарифмирование дохода
3. **One-Hot Encoding:**
   - Преобразование категориальных данных в бинарные признаки.
   - Пример: цвет (зеленый, красный) → [1, 0], [0, 1].
4. **Scaling (Масштабирование):**
   - Приведение данных к единому масштабу (например, от 0 до 1)
   - Пример: использование Min-Max Scaling или Z-Score
5. **Imputation (Заполнение пропусков):**
   - Заполнение пропущенных значений
   - Пример: заполнение медианой или средним
6. **Outlier Handling (Обработка выбросов):**
   - Удаление или корректировка выбросов
   - Пример: замена выбросов на границы IQR

## 4. Применение:

Feature Engineering играет ключевую роль в успешности моделей машинного обучения, обеспечивая:

- Улучшение качества данных
- Оптимизацию предсказательной силы признаков

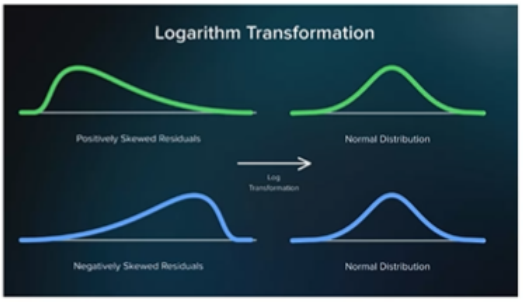
**Итог:** комбинация подходов и техник Feature Engineering позволяет эффективно готовить данные, улучшать качество моделей и выявлять скрытые зависимости

# Log Transformation (Логарифмическое преобразование)

Log transformation

Преимущества:

- Приведение к нормальному распределению
- Уменьшение влияния выбросов
- Стабилизация дисперсии
- Линеаризация экспоненциальных отношений
- Улучшение работы линейных моделей




## Описание:

Логарифмическое преобразование — это математическое преобразование данных, которое применяется для уменьшения асимметрии и стабилизации дисперсии в данных. Оно полезно, если распределение данных скошено или имеет выбросы

## Преимущества:

1. **Приведение к нормальному распределению:**
   - Снижает скошенность данных (положительную или отрицательную)
   - Упрощает работу с методами, чувствительными к распределению данных (например, линейные модели)
2. **Уменьшение влияния выбросов:**
   - Уменьшает эффект экстремальных значений, делая их менее значимыми
3. **Стабилизация дисперсии:**
   - Снижает изменчивость данных, что важно для некоторых моделей
4. **Линеаризация экспоненциальных отношений:**
   - Преобразует экспоненциальные или мультипликативные связи в линейные
5. **Улучшение работы линейных моделей:**
   - Повышает точность линейных регрессий и других моделей, основанных на предположении о нормальном распределении

## Пример на графике:

1. **До логарифмического преобразования:**
   - Верхние графики показывают положительно и отрицательно скошенные распределения
2. **После логарифмического преобразования:**
   - Преобразование приводит распределения к симметричной форме, близкой к нормальной

## Когда применять:

- При наличии положительно или отрицательно скошенных данных.
- Для работы с данными, которые имеют выбросы
- Если переменные показывают экспоненциальную зависимость

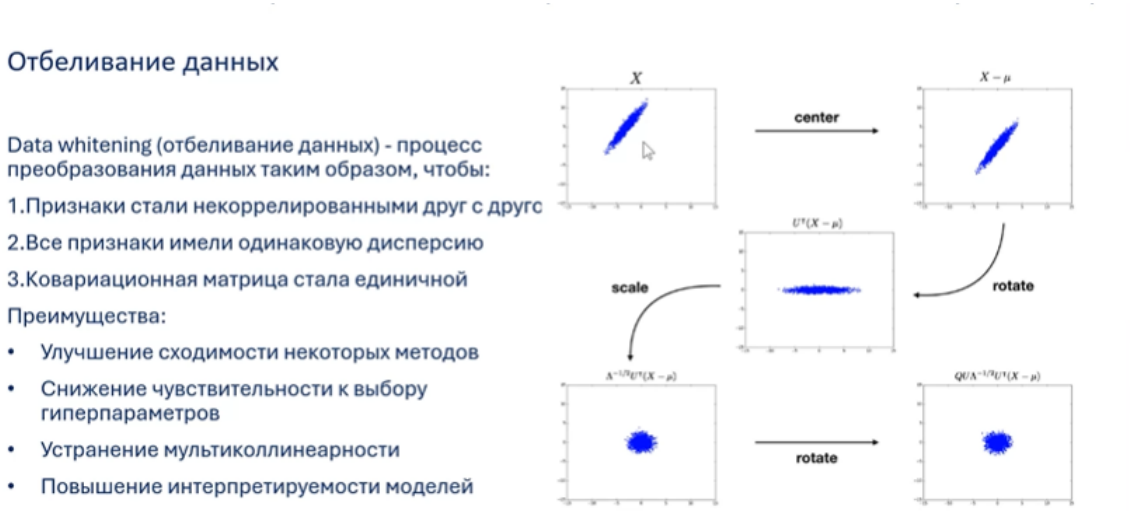
## Примечания:

- Логарифмическое преобразование требует, чтобы значения данных были положительными. Для работы с нулями или отрицательными значениями используется модификация, например, log(x+c), где

c — константа

**Итог:** логарифмическое преобразование — это мощный инструмент для подготовки данных, который улучшает их пригодность для моделирования, особенно для линейных и статистических методов

# Отбеливание данных (Data Whitening)




## Определение:

Отбеливание данных — это процесс преобразования, направленный на то, чтобы:

1. Признаки стали некоррелированными друг с другом
2. Признаки имели одинаковую дисперсию
3. Ковариационная матрица стала единичной

## Этапы процесса:

1. **Центрирование (Center):**
   - Удаляется среднее значение каждого признака.
   - Формула: `X_centered = X - μ`
   - μ — среднее значение признака.
2. **Масштабирование (Scale):**
   - Приведение признаков к одинаковой дисперсии.
   - Формула: `X_scaled = X_centered / σ`
   - σ — стандартное отклонение признака.
3. **Ротация (Rotate):**
   - Устраняется корреляция между признаками, ковариационная матрица становится единичной
   - Формула: `X_whitened = Q * Λ^(-1/2) * Q^T * X_scaled`
   - Q — матрица собственных векторов
   - Λ— диагональная матрица собственных значений

## Преимущества:

- **Улучшение сходимости методов:** Ускоряет обучение моделей
- **Снижение чувствительности к выбору гиперпараметров:** Данные становятся однородными
- **Устранение мультиколлинеарности:** Признаки становятся независимыми
- **Повышение интерпретируемости:** Упрощается анализ данных

## Иллюстрация:

1. Исходные данные (X) — признаки коррелированы
2. Центрирование ($X - \mu X$) приводит данные к нулевому среднему
3. Масштабирование ($X/\sigma X$) нормализует дисперсию
4. Ротация ($Q * \Lambda(-1/2) * Q^T$) устраняет корреляцию
   **Итог:** отбеливание данных — это важный этап обработки данных, который улучшает качество данных для моделей машинного обучения и повышает эффективность алгоритмов