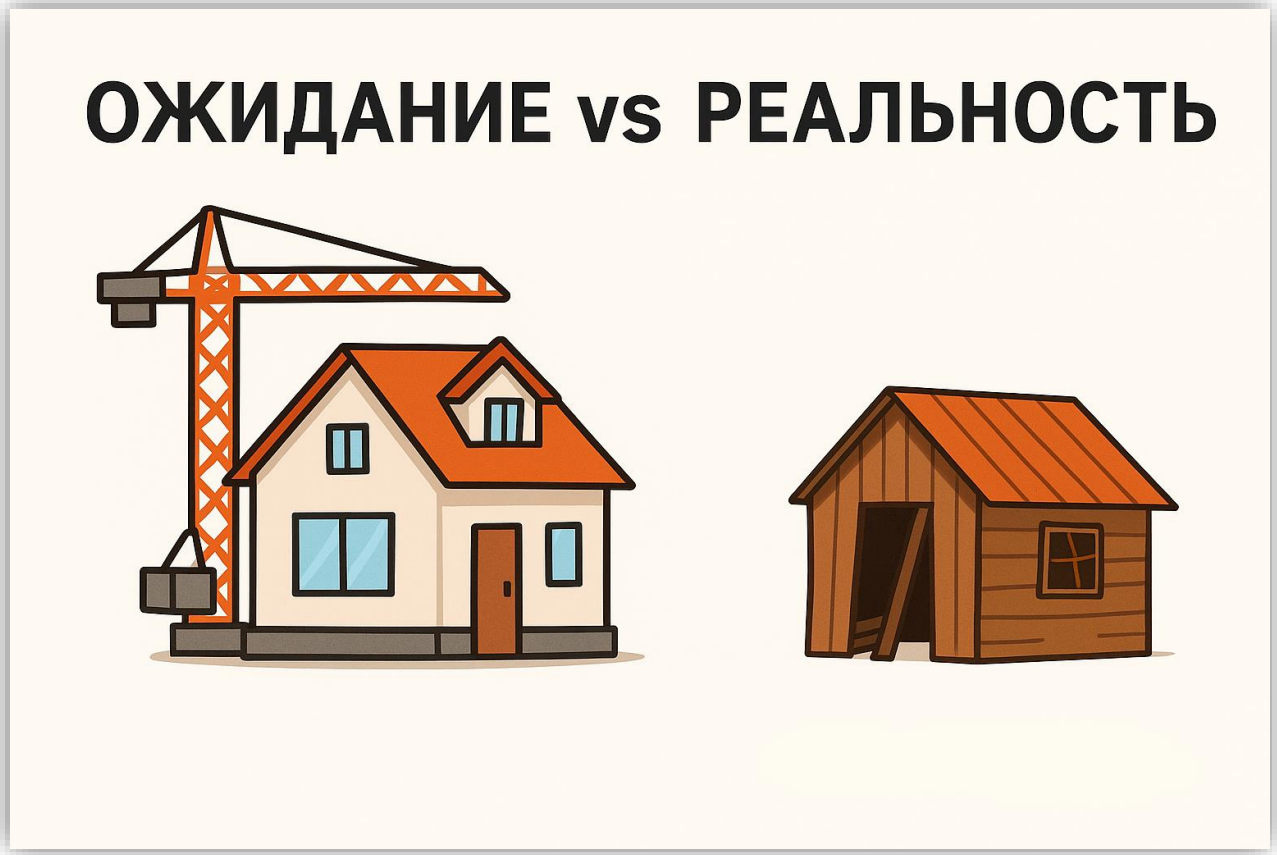
Основные этапы жизненного цикла

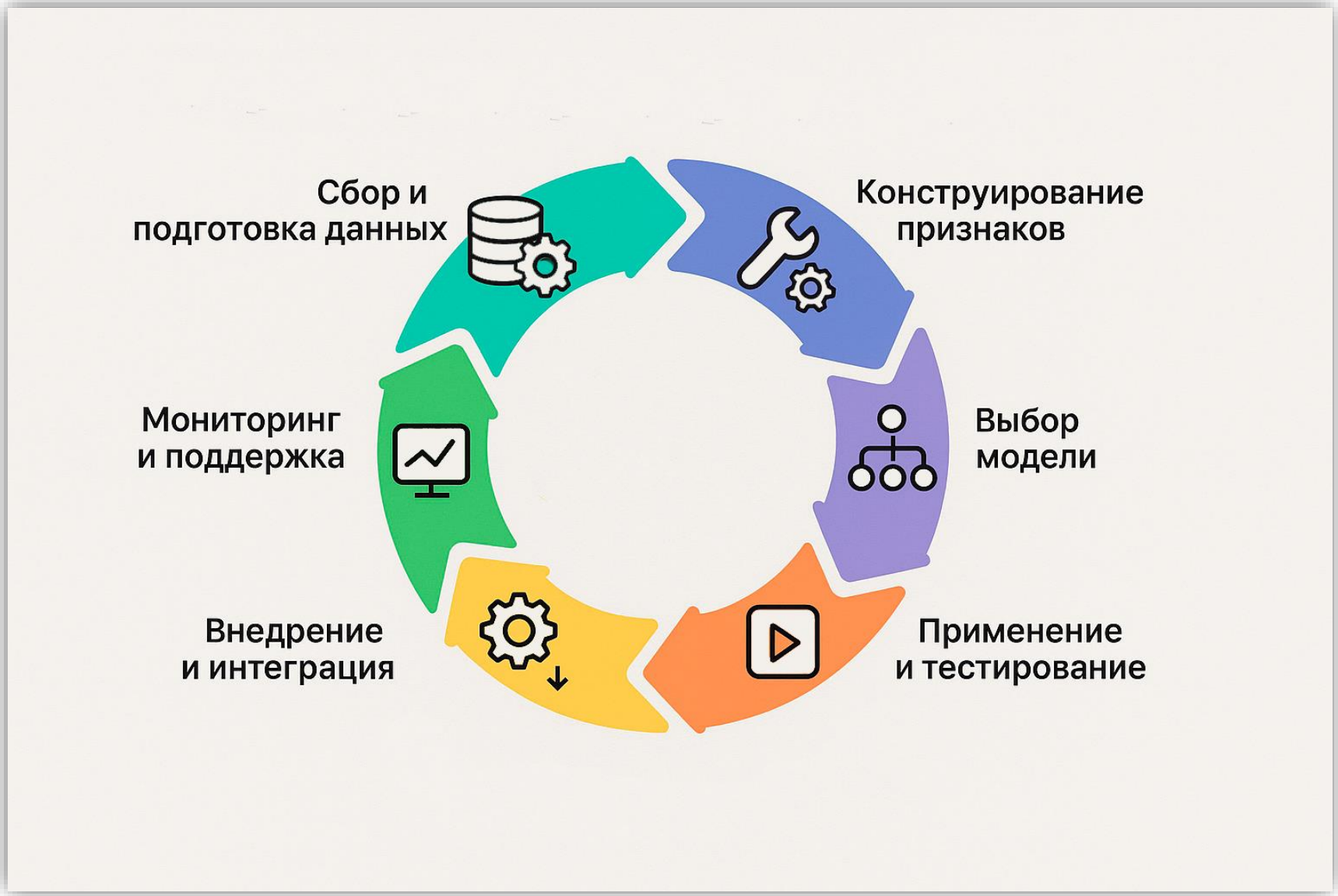
Лекция 1

Машинное обучение ≠ просто модель

- Важен **процесс от задачи до внедрения**
- Без цикла → «сырое тесто» вместо готовой пиццы
- Пример: Amazon HR-модель (дискриминация женщин)



Жизненный цикл ML



Формулировка задачи

Нужно перевести задачу в ML-формат

Примеры:

- «Предсказать цену квартиры» - регрессия
- «Определить спам» - классификация
- «Сегментировать клиентов» - кластеризация

Ошибка: «делаем модель просто потому что есть данные»

Хорошо поставленная задача - половина успеха

- Чёткая **цель**: что именно мы решаем или оптимизируем
- Правильная **метрика**: чем измеряем успех
- Реальные **данные**: помогут ли решить задачу
- **Ошибка**: расплывчатая формулировка = бессмысленный результат

Сбор данных: где брать и что пойдёт не так

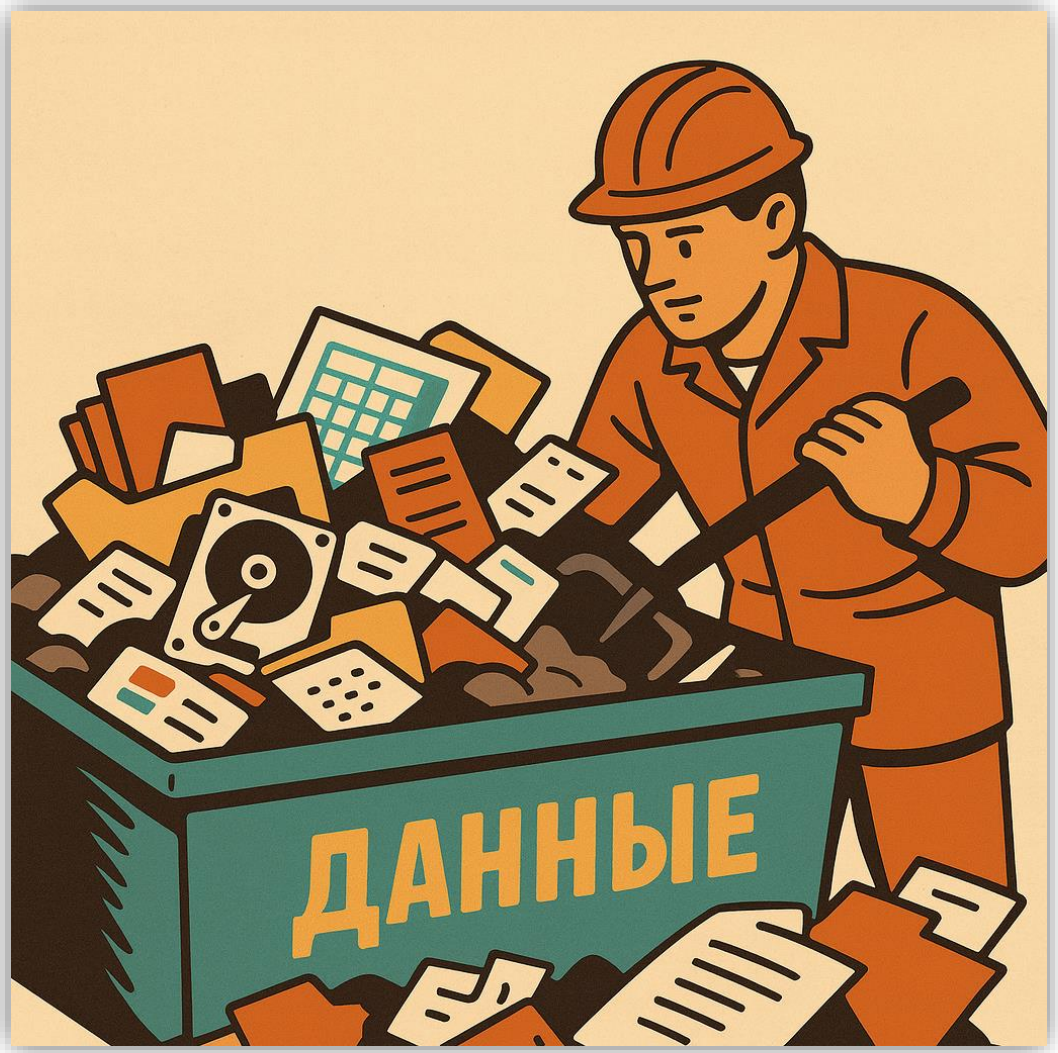
Источники:

- внутренние базы и логи, CRM
- открытые датасеты (Kaggle, HuggingFace Datasets, UCI ML Repo)
- API (Twitter, биржи, картографические сервисы)
- краудсорсинг и ручная разметка (Яндекс Задания, MTurk)

Проблемы:

- пропуски, дубликаты, шумнесогласованные форматы («M/F», «male/female», «1/0»)
- данные ≠ реальность (смещения, баги в источниках)

Факт: 80% времени уходит именно сюда



Подготовка данных

- Очистка от пропусков, дубликатов и прочего
- Приведение форматов: категориальные признаки, числовые поля, даты
- Масштабирование и нормализация признаков
- Балансировка классов (oversampling / undersampling)
- Ошибка: подготовили данные только для обучающей выборки → на новых примерах качество резко падает

Конструирование признаков: превращаем данные в информацию

- Признак = числовая или категориальная характеристика объекта
- Ручное проектирование (основано на предметных знаниях)
- Автоматическое извлечение (нейросети, embeddings)
- Качество признаков определяет качество модели
- Ошибка: использование «сырых» данных без преобразований

Выбор модели: от простых к сложным

Базовые модели:
- Линейная/логистическая регрессия, kNN, дерево решений
- Полезны как старт для понимания данных

Ансамбли и композиции:
- Random Forest, Gradient Boosting (XGBoost, LightGBM, CatBoost)
- Часто используются в индустрии и на Kaggle

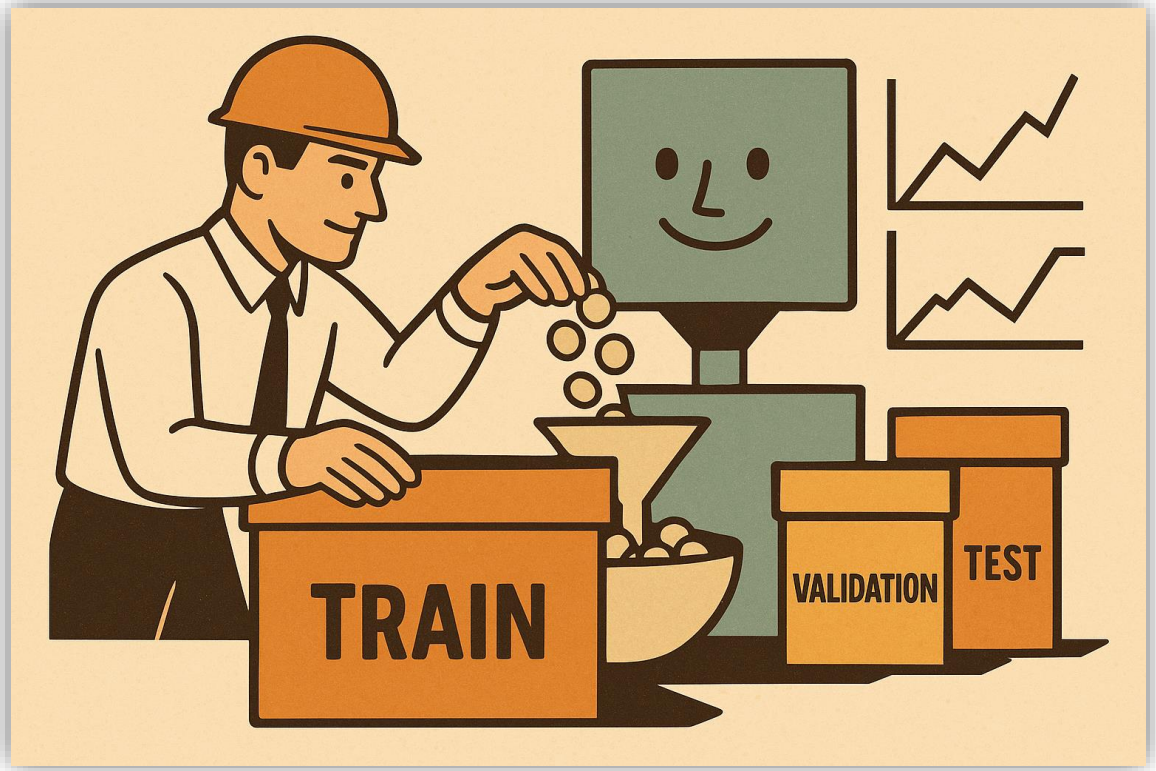
Глубокое обучение:
- Нейросети для изображений, текста, аудио
- Transfer learning, предобученные модели
- Трансформеры

Не всегда нужно начинать с самых базовых алгоритмов. Важно смотреть на лучшие практики в науке (NeurIPS, ICML, ICLR) и в соревнованиях.

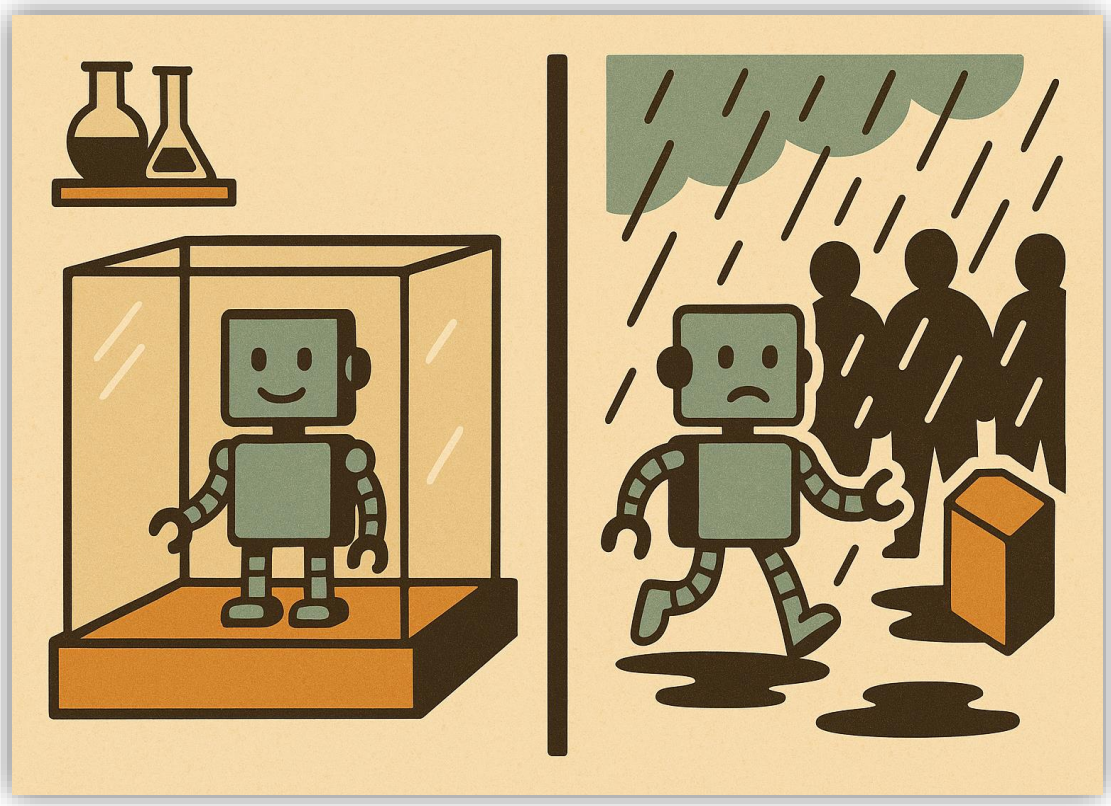
Ошибка: выбор модели «по моде» или без осознанного сравнения

Обучение и валидация

- Данные делятся на train / validation / test
- Модель обучается на тренировочной выборке
- Валидация нужна для подбора гиперпараметров
- Кросс-валидация повышает надёжность оценки
- Опасность: переобучение (overfitting)



Применение и тестирование

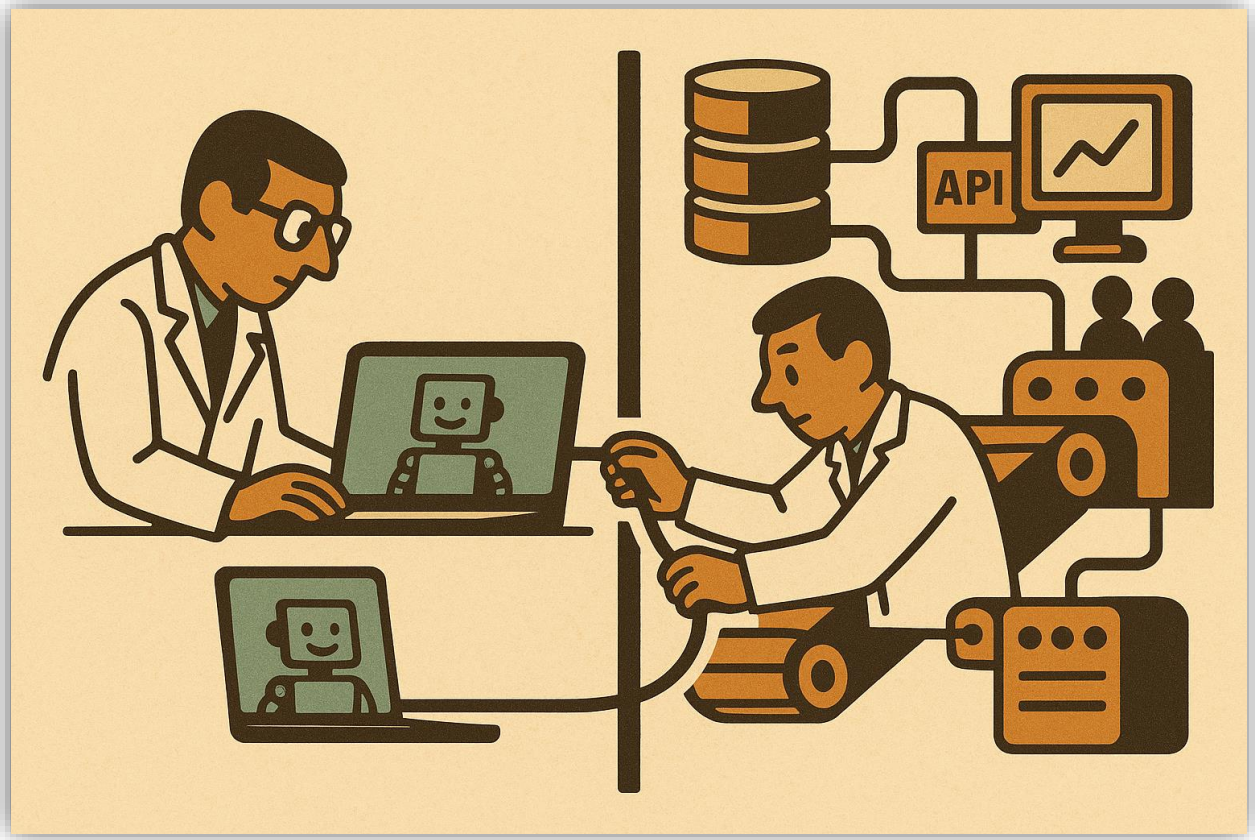


📌 **3 золотых правила тестирования модели:**

- **Тестируем на новых данных** - модель должна работать за пределами обучающей выборки.
- **Сравниваем с baseline** -новая модель должна быть лучше простого решения или приближена к текущим разработкам (SOTA).
- **Оцениваем по целевым метрикам** - метрики должны отражать задачу, а не только «красиво смотреться».

Внедрение и интеграция

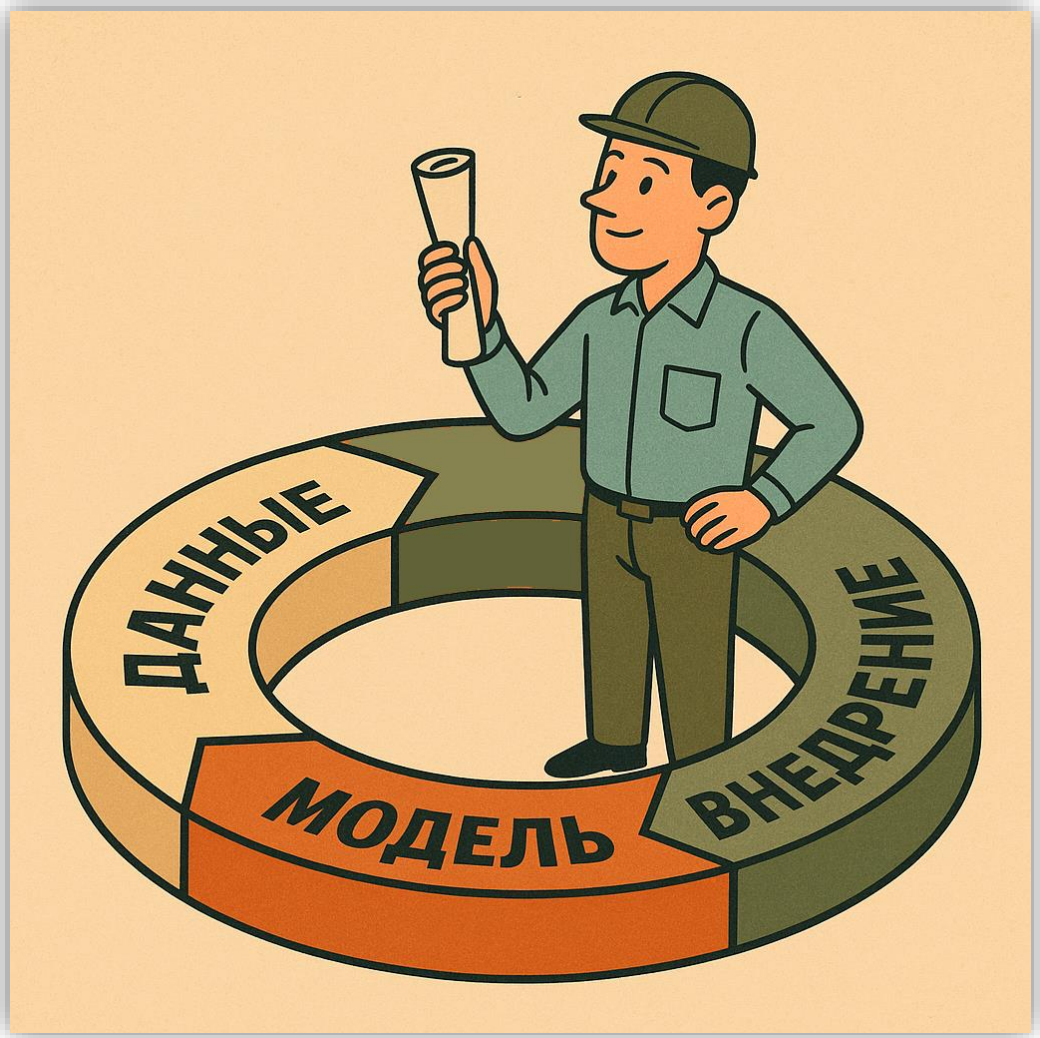
- Интеграция в существующие IT-системы и процессы
- Требования к скорости (latency) и масштабируемости
- A/B-тесты и пилотные запуски перед полным внедрением
- Взаимодействие с другими сервисами (API, базы данных, очереди сообщений)
- Ошибка: модель работает в экспериментальной среде, но не в продакшене



Мониторинг и поддержка

- Качество модели со временем меняется (data drift, concept drift)
- Важно отслеживать метрики в продакшене
- Регулярное обновление и переобучение на новых данных
- Логи, алерты и автоматизация процессов поддержки
- Ошибка: «обучили один раз — и забыли»

Замкнутый цикл: когда и почему возвращаемся к началу



- ML-процесс итеративный, а не линейный
- Ошибки на поздних этапах → возврат к данным и признакам
- Новые данные требуют обновления модели
- Улучшение качества = повторение цикла

Итоги: что мы прошли

- Поняли, что ML = процесс, а не только модель
- Разобрали этапы жизненного цикла:
  - постановка задачи
  - работа с данными
  - признаки и выбор модели
  - обучение и валидация
  - применение и тестирование
  - внедрение и поддержка
- Увидели, что цикл замкнутый и итеративный
- Поняли, что ошибки на любом шаге рушат проект