Лекция 01
Инженерия ИИ: система, роли, жизненный цикл, требования к данным

Рамка курса

21 неделя

Длительность курса

Курс длится 21 неделю, включает лекции и семинары

ИИ-системы

Фокус курса

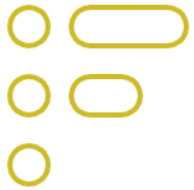
Курс охватывает ИИ-системы от данных до продакшена, а не только модели

Лекции, семинары, проект

Формат работы

Курс включает теоретические лекции, практические семинары и индивидуальный проект

Код, модель, API



Результат

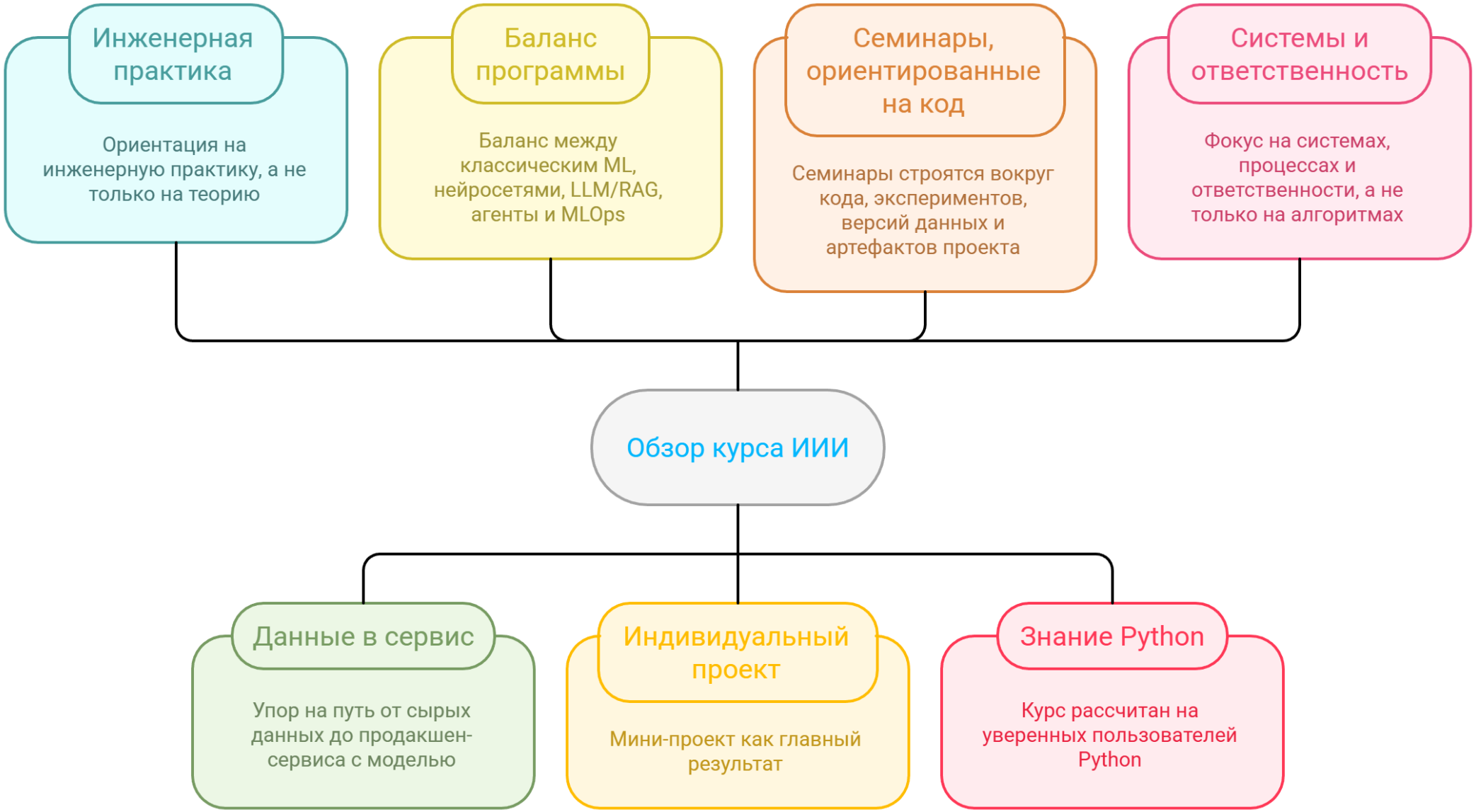
Результатом является репозиторий с кодом, моделью, API-сервисом, наблюдаемостью и безопасностью

aie-course-meta

Meta-репозиторий

Главный источник материалов и объявлений курса на GitHub

Инженерный взгляд на ИИ-системы



Что получите на выходе

Репозиторий кода

Личный репозиторий с кодом проекта, включая структуру, модули, тесты и понятный README.

Модели и эксперименты

Обученные модели и протокол экспериментов, показывающие, какие варианты пробовали и какие метрики получили.

API-сервис

API-сервис с эндпоинтом для предсказаний и health-check'ом.

Контейнеризация Docker

Базовая контейнеризация с помощью Docker для локального развертывания сервиса.

Минимальный мониторинг

Минимальный мониторинг, включающий логи запросов, метрики качества и работоспособности, а также простые отчеты.

Документация по безопасности

Документация по безопасности и работе с данными, включая SECURITY.md, правила логирования и обращения с секретами.

Понимание роли ИИ

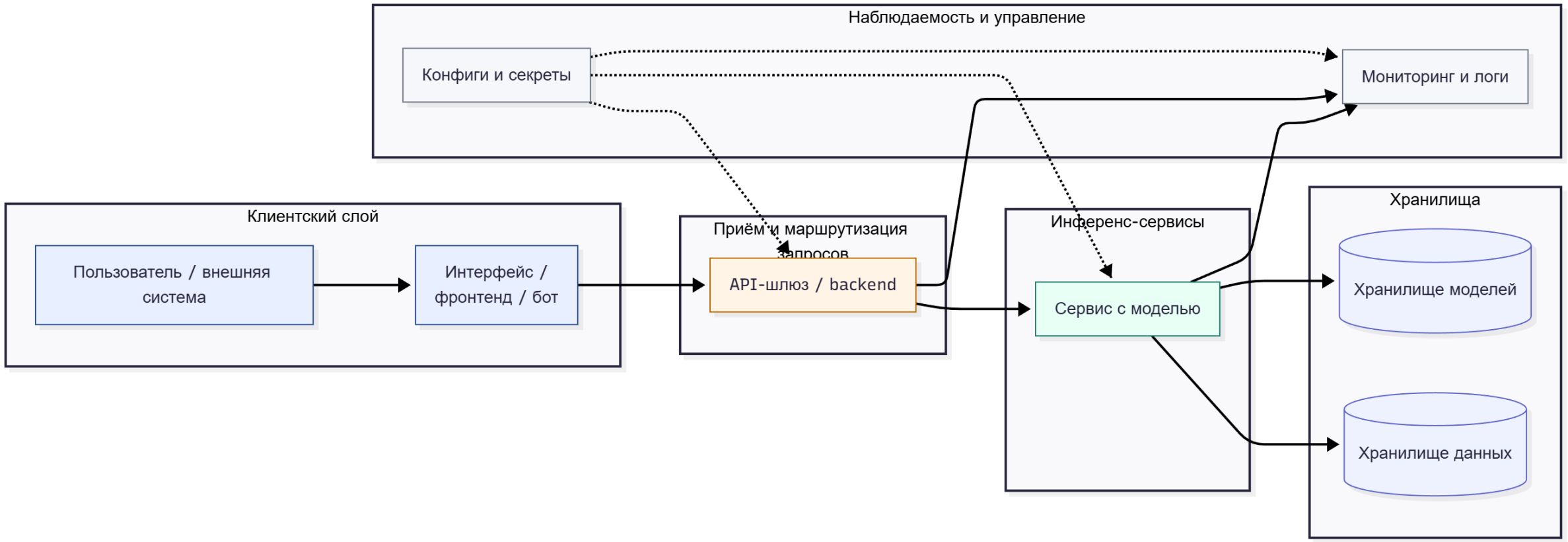
Понимание роли ИИ-компонента в системе и умение объяснить свой проект технической и нетехнической аудитории.

Преподаватели и структура курса

- Силаев Юрий Владимирович – лекции (инженерные аспекты, агентные системы, MLOps), семинары.
- Пантюхин Дмитрий Валерьевич – лекции (ИИ от классического ML до мультимодальных моделей).

Неделя	Название лекции (могут быть корректировки)	Название семинара (могут быть корректировки)
1	Инженерия ИИ: система, роли, жизненный цикл, требования к данным	Dev-среда и NumPy
2	Данные и признаки: сбор, очистка, валидация, DataOps	Pandas и контроль качества данных
3	Математический праймер для ML	EDA и визуализация
4	Метрики и экспериментальный дизайн	Валидация экспериментов и GridSearch в sklearn
5	Линейные модели	Линейные модели и калибровка вероятностей
6	Деревья решений и ансамбли	Ансамбли: Random Forest и бустинг (XGBoost/CatBoost)
7	Неподконтрольное обучение	Кластеризация и снижение размерности
8	Нейронные сети: основы	PyTorch 101: Dataset, DataLoader и первый MLP
9	Обучение нейросетей и оптимизация	Оптимизация обучения и регуляризация в PyTorch
10	Сверточные сети и компьютерное зрение	CNN: аугментации и transfer learning (ResNet)
11	Задачи CV: распознавание, детекция, сегментация	Детекция и сегментация с готовыми библиотеками
12	Последовательности и время	Временные ряды: базовый прогноз на LSTM/GRU
13	Трансформеры: архитектура и эволюция	HuggingFace: токенизация и дообучение BERT
14	Векторные представления и поиск знаний (RAG)	Векторный поиск: FAISS/pgvector и мини-RAG
15	Большие языковые модели	LLM-адаптация: LoRA/PEFT на практической задаче
16	Мультимодальные модели	Мультимодальность: CLIP-поиск и обзор диффузии
17	Агентные ИИ-системы	Агентные пайплайны: инструменты и трассировка шагов
18	Оценка качества и мониторинг	Мониторинг качества: MLflow и Evidently
19	Безопасность и доверие к ИИ	Безопасность ИИ-сервисов: атаки, guardrails и секреты
20	MLOps: управление данными и моделями	MLOps-пайплайн: DVC и MLflow на практике
21	Развертывание и эксплуатация сервисов моделей	Сервис модели: FastAPI/Flask, Docker и наблюдаемость

ИИ-система состоит из взаимосвязанных компонентов



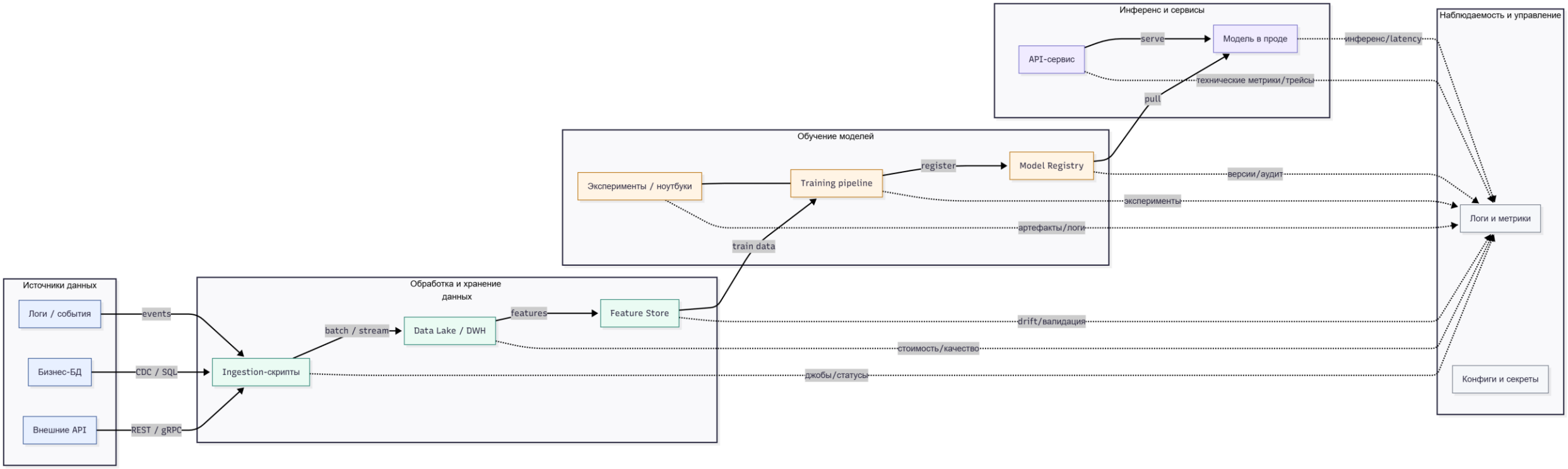
- Пользователь или внешняя система посылают запросы через интерфейс (веб, мобильное приложение, бот).
- Интерфейс и API-шлюз принимают запрос, проверяют формат и права, передают в сервис.
- Сервис с моделью обрабатывает вход и делает предсказание.
- Хранилище данных обеспечивает доступ к сырым и агрегированным данным.
- Хранилище моделей хранит версии обученных моделей и позволяет выбирать нужную.
- Мониторинг и логи фиксируют запросы, ответы, ошибки; конфиги и секреты управляют настройками и доступом.

Модель и ИИ-система – это не одно и то же

Базовые термины курса

1	**Модель** Программный объект или функция, которая по входным данным возвращает предсказание (число, класс, текст и т.п.).
2	**Датасет** Структурированный набор примеров, на которых модель обучают и проверяют (таблица, коллекция текстов, картинок и т.д.).
3	**Признак (feature)** Числовое или категориальное описание объекта, подаваемое на вход модели
4	**Метрика** Числовой показатель качества работы модели на датасете
5	**Артефакт** Любой важный результат работы, который нужно сохранять: модель, датасет, отчет, конфиг, лог.
6	**Пайплайн** Упорядоченная последовательность шагов обработки данных, обучения и применения модели, которую можно запускать как целое.

Архитектура ИИ-системы раскладывается на стандартные блоки

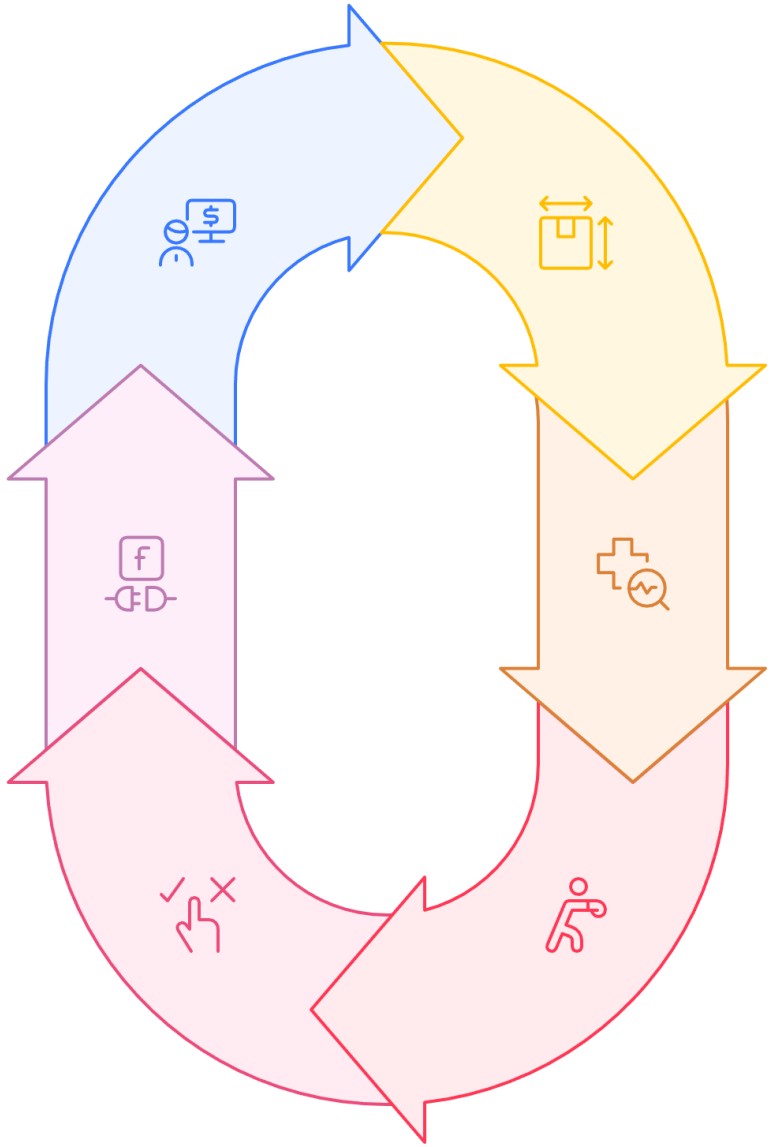


- Источники данных: логи, бизнес-БД, внешние API.
- Ingestion-скрипты забирают данные и складывают их в Data Lake или DWH.
- Feature Store хранит подготовленные признаки, которые переиспользуются в обучении и инференсе.
- Блок обучения моделей: эксперименты, training pipeline, реестр моделей.
- Блок инференса: API-сервис, который использует выбранную версию модели из реестра.
- Блок наблюдаемости: логи, метрики, конфиги и секреты, отвечающие за контроль и безопасные настройки.

Пример ИИ-системы

1. **Данные поступают в хранилище**
   Логи и данные о заказах попадают в хранилище
2. **Строятся признаки**
   Из данных строят признаки: частота покупок, категории интереса, средний чек
3. **Модель предсказывает**
   Модель предсказывает, какие товары сейчас уместно рекомендовать
4. **Сервис встраивается в страницу**
   Сервис рекомендаций встраивается в страницу: при запросе страница, API, модель, список товаров
5. **Мониторинг отслеживает**
   Мониторинг отслеживает клики по рекомендациям, конверсии и ошибки
6. **Система дообучается и обновляется**
   По этим данным систему дообучают и обновляют

Жизненный цикл ИИ-системы состоит из повторяющихся этапов



Постановка задачи и метрик

Определение целей и метрик

Сбор и анализ данных

Сбор, анализ и оценка качества данных

Проектирование признаков и обучение моделей

Разработка признаков, выбор baseline, подбор архитектур, эксперименты

Валидация и выбор решения

Валидация моделей, выбор оптимального решения, sanity-check

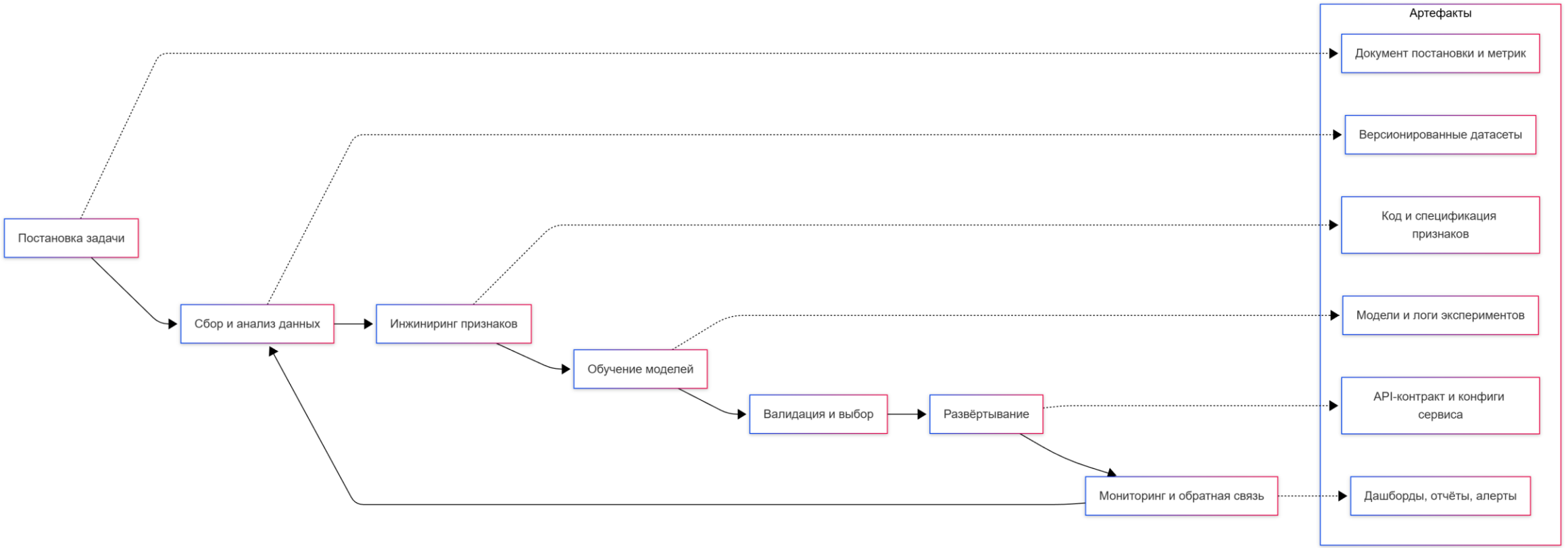
Развертывание и интеграция

Развертывание модели и интеграция в продукт

Мониторинг и улучшение

Мониторинг производительности, выявление дрейфа, переобучение

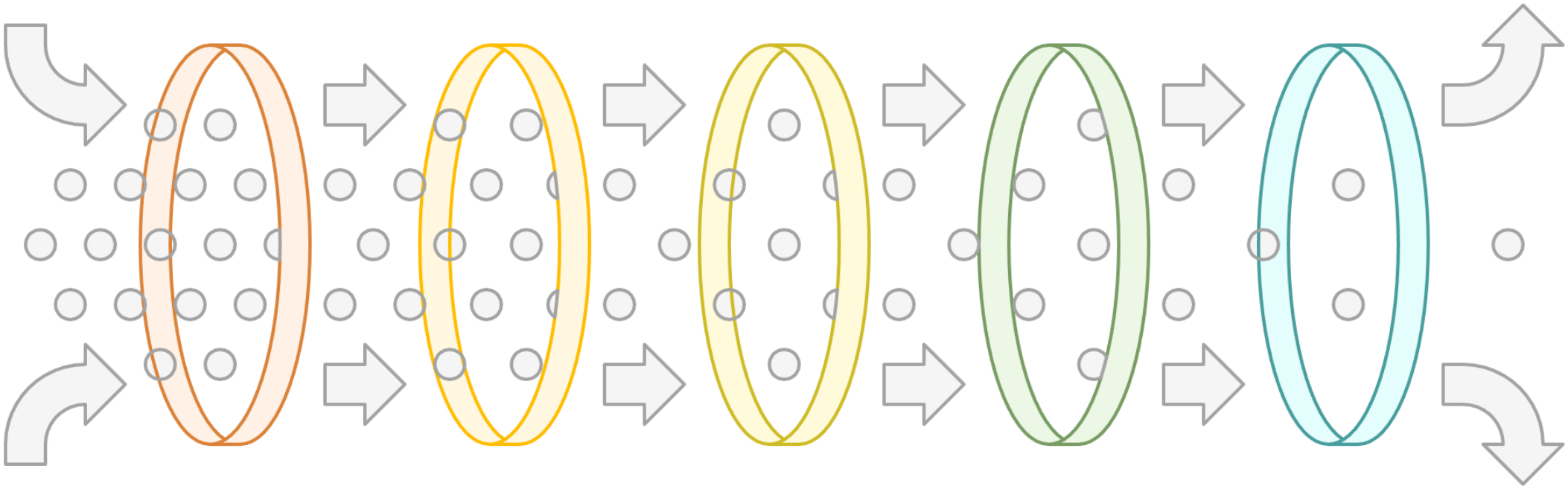
Жизненный цикл как pipeline с артефактами



- Каждый шаг жизненного цикла фиксируется артефактами: документами, кодом, данными, отчетами.
- Постановка задачи рождает документ с формулировкой проблемы и целевых метрик.
- Сбор и анализ данных – версионированные датасеты и описание их качества.
- Инжиниринг признаков – код и спецификации feature’ов, которыми можно переиспользовать.
- Обучение и валидация – набор моделей и протокол экспериментов с метриками.
- Развертывание и мониторинг – API-контракты, конфиги сервиса, дашборды и алерты.

Путь от бизнес-проблемы до первой модели проходит через несколько шагов

Бизнес-кейс оттока клиентов



Формулировка ML-задачи

Предсказать вероятность оттока и выбрать метрику

Сбор данных

История оплат, активности, обращения в поддержку

Построение baseline

Логистическая регрессия (например) и оценка качества

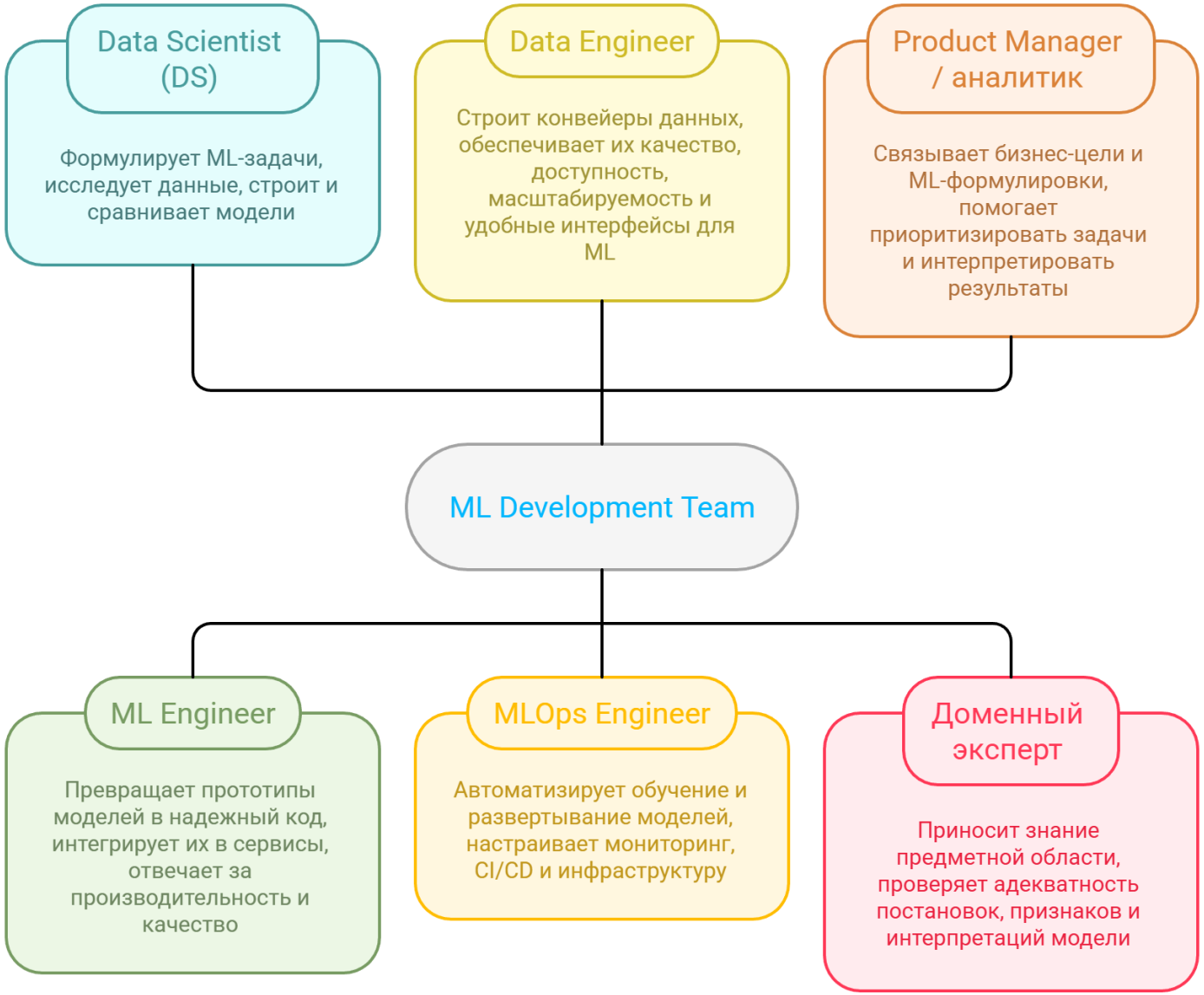
Оценка качества

Оценивается качество по согласованной метрике

Обсуждение результатов

Обсуждение точности, ошибок и интеграции модели

Роли в команде ИИ-проекта взаимно дополняют друг друга



Роли в ИИ-команде отличаются фокусом и зонами ответственности

Роль	Основной фокус	Типичные артефакты
Data Scientist	Модели и эксперименты	Ноутбуки, отчеты, baseline-модели
ML Engineer	Код и сервисы	Модули с моделью, API, тесты, CI-скрипты
Data Engineer	Данные и конвейеры	ETL-пайплайны, схемы БД, data quality-отчеты
MLOps Engineer	Инфраструктура и процессы	Pipeline-конфиги, deployment-скрипты, дашборды
Product Manager	Бизнес-цели и приоритизация	Roadmap, требования, A/B-планы
Доменный эксперт	Предметная область и ограничения	Описания сценариев, правила, чек-листы

Роли в ИИ-проекте решают одну задачу через совместный сценарий

Product Manager

Формулирует проблему и определяет метрику успеха (удержание).

Data Engineer

Налаживает сбор и подготовку данных (события, оплаты, обращения).

MLOps Engineer

Настраивает pipeline переобучения, развертывание и мониторинг качества в проде.

Data Scientist

Уточняет задачу, анализирует данные, строит прототип модели.

ML Engineer

Упаковывает модель в сервис, проектирует API, добавляет валидацию и тесты.

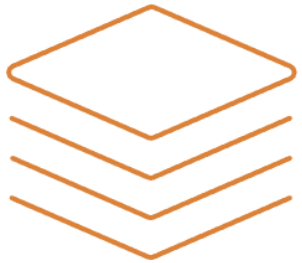
Итог

Кейс требует скоординированной работы всех ролей.

Качество, объем и актуальность данных ограничивают возможности модели

Качество данных

Отсутствие ошибок, пропусков, дубликатов и странных значений.



Объем и покрытие

Достаточно примеров для редких классов и граничных случаев.

Актуальность

Свежие данные соответствуют текущему миру.

Репрезента-тивность

Данные обучения похожи на данные в реальности.

Качество разметки

Надежные целевые метки, без систематических ошибок.

Этика и право – рамки работы с данными на курсе

Учебные данные

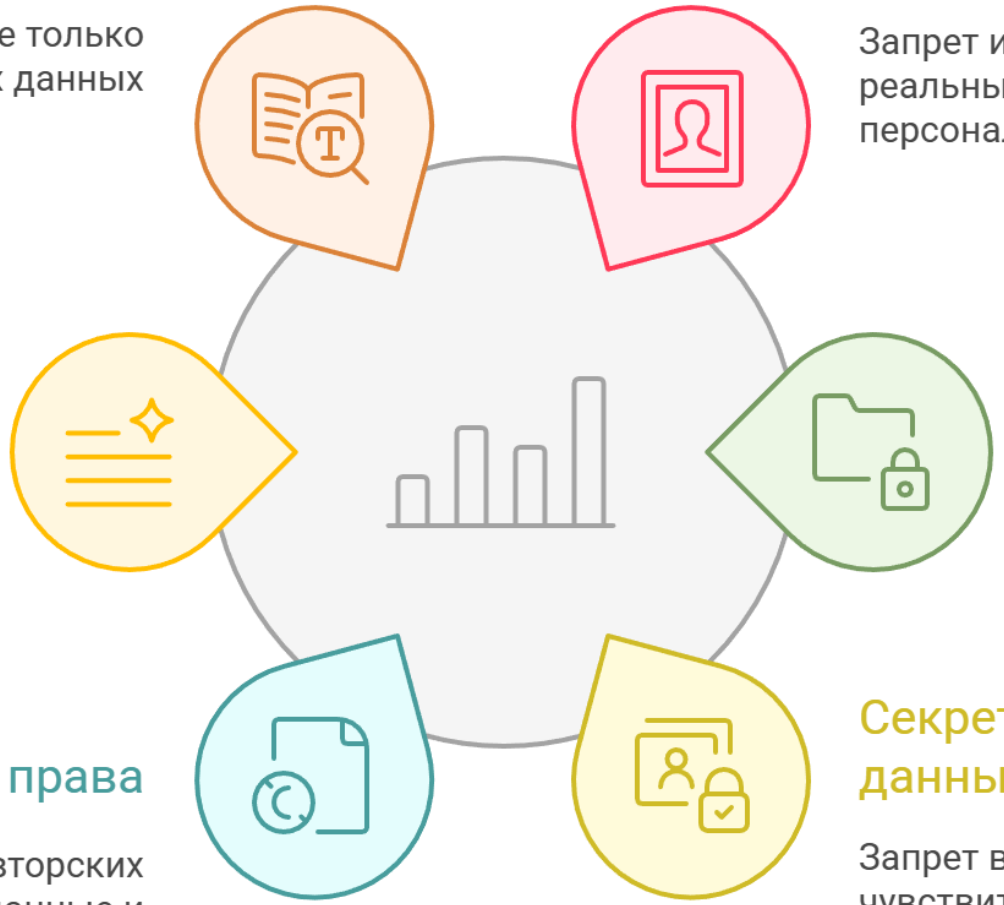
Использование только учебных данных

Персональные данные

Запрет использования реальных персональных данных

Реалистичные кейсы

Моделирование на синтетике или публичных датасетах



Конфиденциальные данные

Запрет использования конфиденциальных или коммерчески чувствительных данных

Авторские права

Уважение авторских прав на данные и материалы

Секретные данные

Запрет выгрузки чувствительных данных и секретов в публичные места

Плохой кейс с логами

Модель в ноутбуке выглядит как обычная функция предсказания


```python
import numpy as np
from sklearn.linear_model import LogisticRegression

# Игрушечные данные: признак = число сессий за неделю
X = np.array([[1], [2], [3], [6], [8], [10]])
y = np.array([1, 1, 0, 0, 0, 0])  # 1 = риск, 0 = ок

model = LogisticRegression().fit(X, y)

def predict_risk(sessions_last_week: int) -> int:
    proba = model.predict_proba([[sessions_last_week]])[0, 1]
    return int(proba > 0.5)  # 1, если риск высокий

print(predict_risk(1), predict_risk(7))  # пример "мини-API"
```

- В ноутбуке мы строим крошечную модель на игрушечных данных.
- Функция predict_risk – это уже «микро-API»: на вход число, на выход предсказание.
- В реальном сервисе внутри будет похожий вызов модели, просто обернутый в HTTP/JSON.
- Мы увидим, что модель – это не «магия», а обычный объект, который вызывается как функция.
- На следующих занятиях вокруг такой функции появится полноценный сервис с логами и валидацией.

Переход от ноутбука к сервису добавляет API, конфиги и логи

Структура курса на 21 неделю как единая траектория

Недели 1-7

Фундамент

Постановка задач, жизненный цикл, данные, признаки, классический ML и базовые метрики

Недели 8-12

Нейросети и компьютерное зрение

Основы глубокого обучения, простые архитектуры, практические кейсы

Недели 13-16

Трансформеры, LLM и RAG

Работа с текстом, генеративные модели, поиск с дополнением контекстом

Недели 17-21

Агенты, мониторинг, безопасность и MLOps

Наблюдаемость, дрейф, guardrails, пайплайны и эксплуатация

Лекции

Теория и «картина мира»

Семинары дают код, эксперименты и шаги по сквозному проекту

Мини-проект – ожидания и критерии

5 **Документация**

Описание: задача, данные, ограничения, этика, безопасность.

4 **Эксплуатация**

Минимум: API, логирование, мониторинг или отчет.

3 **Качество моделей**

Измеримость: метрики, baseline, обоснование выбора решения.

2 **Репозиторий**

Воспроизводимость: структура, README, инструкции, зависимости.

1 **Итог проекта**

Целостная ИИ-система: данные, код, сервис, мониторинг, документация.

Итоги: пять опорных идей Лекции 01

Модель ≠ ИИ-система

Модель - лишь часть сложной ИИ-системы.

Жизненный цикл и артефакты

ИИ-система развивается циклически, создавая артефакты.

Данные, качество и безопасность

Данные и правила использования ограничивают возможности модели.

Роли и командная работа

Разные специалисты смотрят на систему с разных точек зрения.

Мини-проект и структура курса

Полный цикл разработки за 21 неделю.

Дополнительные материалы

- Блог vas3k – Машинное обучение для людей (https://vas3k.blog/blog/machine_learning/)
- Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*
- Dietmar Jannach et al. – *Recommender Systems: An Introduction*
- Ralph Kimball – *The Data Warehouse Toolkit*
- Andriy Burkov – *Machine Learning Engineering*
- Emmanuel Ameisen – *Building Machine Learning Powered Applications*
- Chip Huyen – *Designing Machine Learning System*
- Mark Treveil et al. – *Introducing MLOps*
- Noah Gift – *Practical MLOps*
- EUCathy O’Neil – *Weapons of Math Destruction*
- OWASP – *Logging Cheat Sheet и остальные Cheat Sheet*
- Google – *Machine Learning Glossary*
- Google Cloud – *MLOps: Continuous delivery and automation pipelines in ML*
- Eugene Yan – личный блог
- Scikit-learn User Guide: Logistic Regression и модели

Q&A и контакты

Группа по дисциплине:

https://t.me/+8dShF1tFSDg0ZmJi