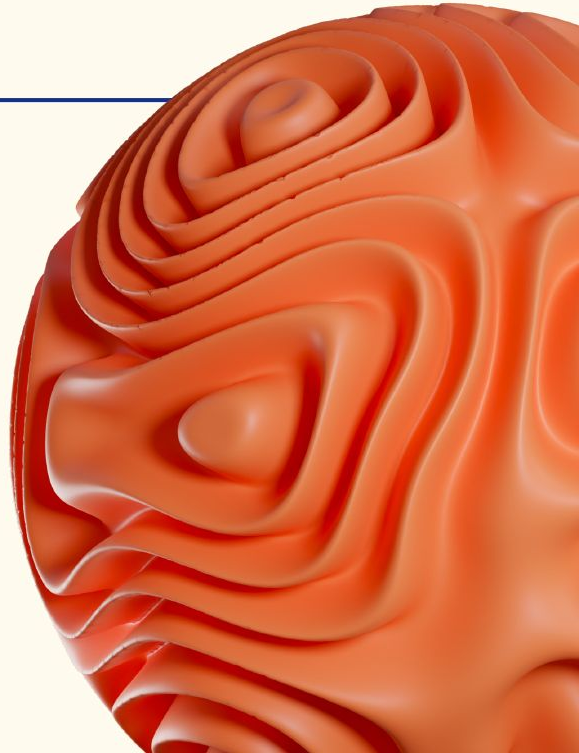
Модели ML в production

МИФИ × SKILLFACTORY

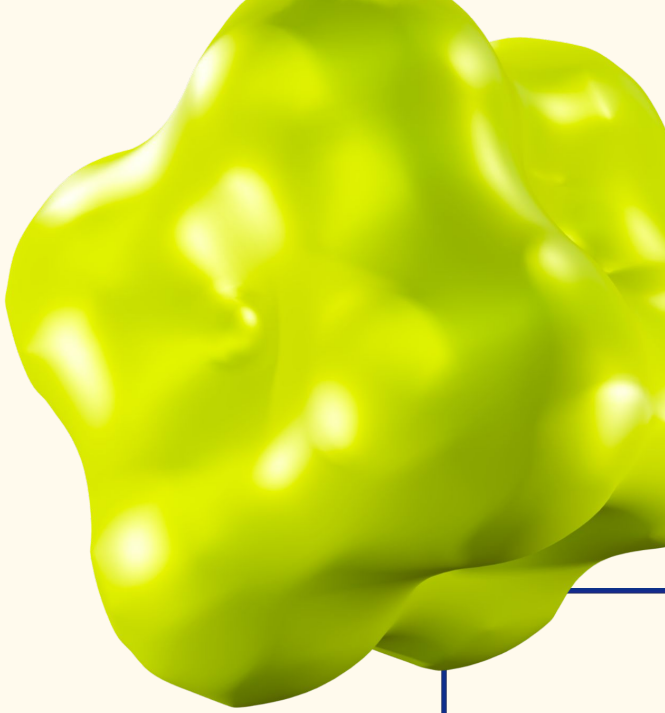
Лекция № 1
“Сериализация и сохранение моделей. Pipeline.”



Жарова Мария Александровна
DS WB-tech, Math&Python&DS lecturer
t.me/data_easy

План занятия

1. Введение: зачем нам этот курс
2. Деплой моделей и сериализация
3. Практика:
    a. Pipeline из sklearn
    b. сохранение объектов с pickle и joblib
    c. бонус: как оформить инференс для большого проекта



1. Введение: зачем нам этот курс

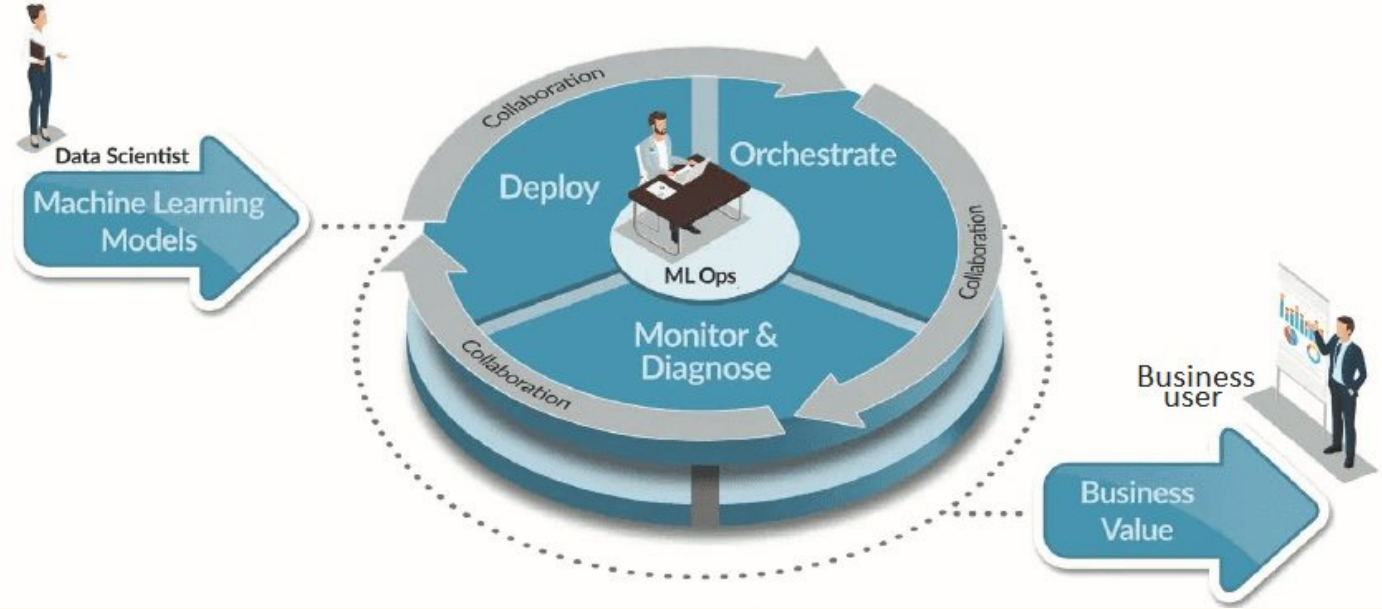
Чему научимся

МИФИ × SKILLFACTORY

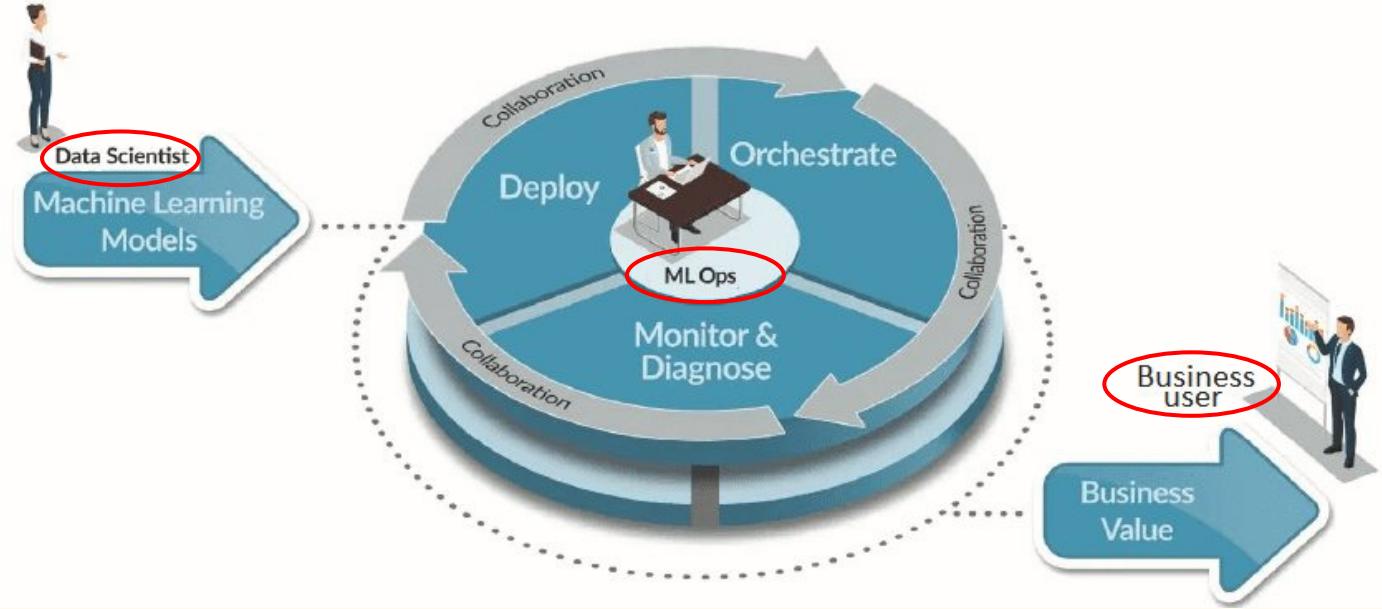
- узнаем про финальные стадии жизни DS-проектов
- познакомимся с понятиями: *изолированность, оркестрация, контейнеризация, воспроизводимость*
- разработаем сервисы для деплоя на Flask (+ может, FastAPI?)
- создадим Docker-контейнер, познакомимся с Docker-compose
- разработаем приложение с микросервисной архитектурой
- узнаем, как оценивать качество моделей после внедрения в production (AB-тестирование)

Правда жизни: Модель в вакууме бесполезна

Правда жизни: Модель в вакууме бесполезна



Правда жизни: Модель в вакууме бесполезна

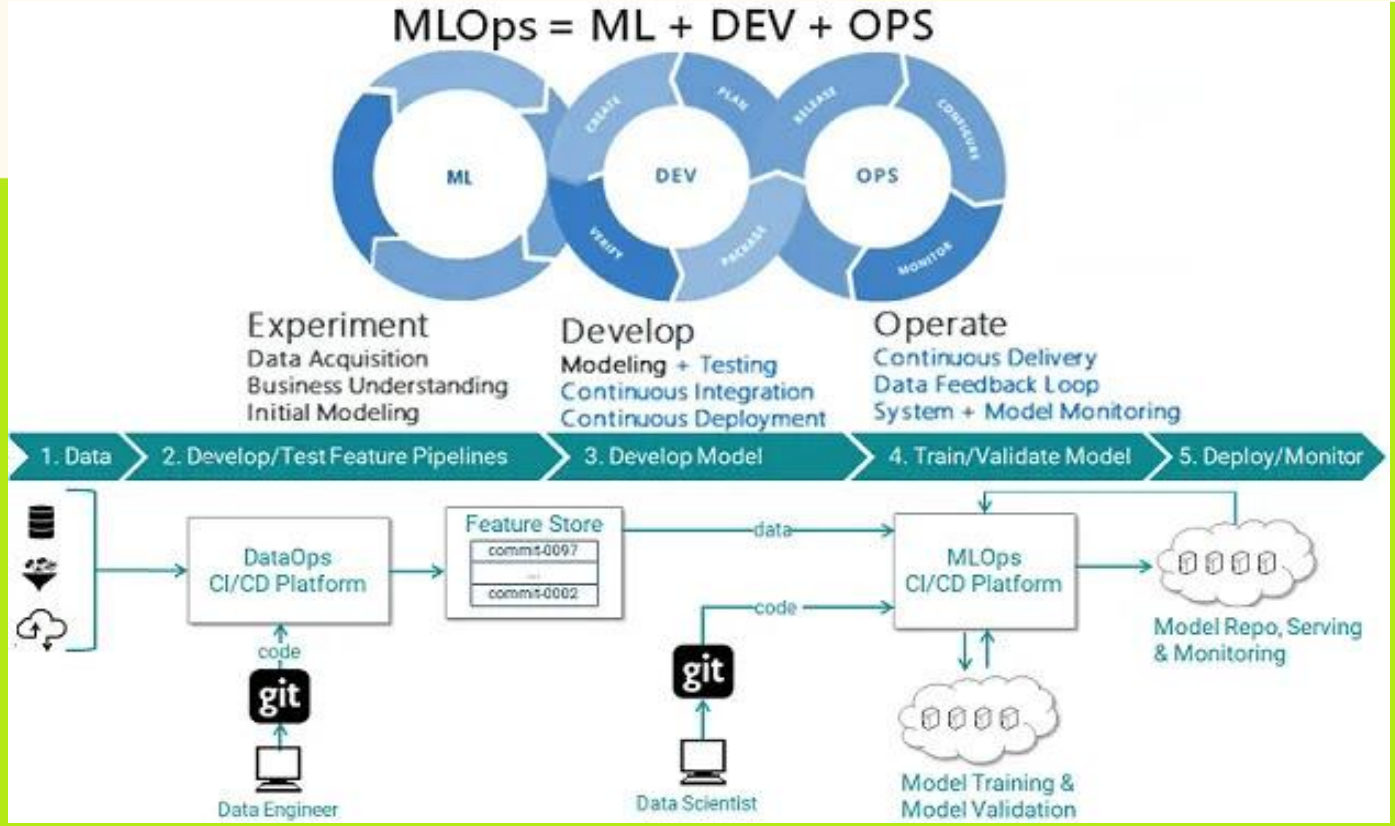


Процесс работы с данными

Термины

- ***Деплой*** (развёртывание программного обеспечения) - это все действия, которые делают программную систему готовой к использованию
- ***Инференс*** - это непрерывная работа какой-либо модели на конечном устройстве (== работа на реальных данных в production).
- ***Production*** - рабочая версия кода, которая попадает заказчику или публикуется для скачивания пользователями.
- ***MLOps*** = ML + DevOps

Термины: MLOps



2. Деплой моделей и сериализация

Сохранение моделей

— С точки зрения питона ***модель - это объект*** (но сложный: включает в себя много зависимостей и подклассов; те же гиперпараметры и параметры)

— ***Сериализация*** - это процесс трансформации *любой структуры данных*, поддерживаемой в языке, в последовательность битов (или байтов). Обратная операция - ***десериализация***.

— гарантированно получаем те же данные
— id при сериализации-десериализации меняется
— сериализовать можно любой объект, поэтому будьте осторожны:)

Pickle и joblib

— Преобразовываем и исследуем данные, обучаем модель.

— Сериализуем модель, сохраняя её в бинарный файл при помощи библиотек.

— Далее этот файл может быть использован для инференса моделей, работы в сервисах и приложениях.

— Для воспроизведения предобработки данных на инференсе используем Pipeline (это слово уже получило нарицательное название “пайплайн”)

Pickle или joblib?

pickle	joblib
менее эффективен при работе с большими массивами данных, не делает специальную оптимизацию для них; медленнее	оптимизирован для работы с большими массивами числовых данных, т.к. использует numpy для хранения; быстрее для числовых массивов, т.к. не требуются доп. преобразования при сериализации
сохраняет объекты без компрессии, что может привести к более тяжелым файлам, особенно если объект включает большие числовые массивы	сохраняет данные в сжатом виде, что позволяет снизить их объем, особенно для больших моделей; полезно для ML
pickle лучше использовать для объектов с более сложными структурами, которые не связаны с машинным обучением или не включают большие массивы чисел	joblib лучше всего подходит для сохранения моделей машинного обучения, содержащих большие массивы данных, особенно – числовые

PMML

— **PMML** (Predictive Model Markup Language) - это XML-диалект, который применяется для описания статистических и DS-моделей.

— PMML-совместимые приложения позволяют легко обмениваться моделями данных между собой.

— Могут быть интегрированы с другими языками программирования.

Пример: “перенести” в SAS модель, разработанную на Python.

Где лучше применить?

Подходит для традиционных моделей машинного обучения и при необходимости в интеграции со старыми системами (где нейронные сети не используются).

ONNX-ML

— **ONNX** (Open Neural Network Exchange) - это открытый стандарт для обеспечения совместимости моделей машинного обучения (Microsoft, Amazon, Facebook).

— Используется для конвертации из одного фреймворка в другой (например, из PyTorch в TensorFlow и наоборот).

— Используется для переноса DL-моделей в приложения с ограниченными вычислительными ресурсами (например, для мобильных устройств).

ONNX Runtime — высокооптимизированная среда выполнения для ускоренного вывода моделей в формате ONNX.

Лучше для нейронных сетей и сложных моделей глубокого обучения, особенно если требуется оптимизация или работа с разными фреймворками.

3. Практика

Спасибо
за внимание!

МИФИ × SKILLFACTORY