

MSDS 603 – Machine Learning Operations

Robert Clements

MSDS Program

University of San Francisco



What if we were asked to build a Smart Transportation System?

- The SFMTA wants to implement a smart transportation management system to optimize traffic flow, reduce congestion, and lower emissions. Using data from traffic cameras, vehicle sensors, public transit GPS, mobile apps, weather services, and scheduled events, the system would predict traffic patterns and adjust traffic light timings, suggest route alternatives, and manage MUNI schedules. The system should be able to adjust to unexpected events, such as collisions, and ensure equitable service across neighborhoods. It should be live 24/7 and demonstrate measurable improvements in commute times and air quality.



We will come back to this!

MLOps

- In this course you will learn about MLOps
 - Sometimes called ModelOps.
- Most important aspects of the end-to-end ML pipeline will be covered
 - Introduce open source tools to get some hands-on practice.
 - Note: some steps require more *theory* than others (e.g. concept drift).
 - Note: some aspects & tools covered in previous courses (e.g. version control with git; docker; orchestration with airflow).
 - Note: different companies use different stacks – we can't learn every tool available!

In This Course...

We will be talking about:

- Requirements gathering and system design
- Experiment tracking
- Registering artifacts
- Data versioning and quality
- Orchestration
- Infrastructure as code
- CI/CD/CT
- Model serving
- Model and operational monitoring

We won't be talking about:

- How to train ML models
- Data storage/warehouses
- Model re-training
- Ethics/Responsible AI
- Compliance
- Experimentation (in the A/B testing sense)
- ALL of the tools and steps of MLOps

How We Will Learn

- Lecture and Discussion
 - Bring your Practicum/previous job experiences to class
- In-class Demos, Labs, Mini-quizzes
- Homework Assignments
- Project
- Grade is based on:
 - Participation (**5%**)
 - Quizzes (**5%**)
 - Homeworks (**10%**)
 - Labs (**40%**)
 - Final Project (**40%**)

Disclaimer: MLOps is a software engineering-heavy field
I am not a software engineer

Discussion Participation (5%)

- Some days we will break up into groups to discuss a problem
- Write answer in **#2025-msds-603-discussion** Slack channel
- Tag everybody in your group to get credit

Mini-quizzes (5%)

- Once per week on Canvas during class
- Open book/note, no talking or sharing answers
- Covers material from previous week
- Can retake it once

Homework Assignments (10%)

- 2 assignments
- Assignment 1 is a two-part assignment
 - First half done in class
 - Second half done at home
- Done *individually*

Labs (40%)

- Labs assigned once or twice per week
- Due following Monday 11:59PM
- Practice with using open source tools locally and in GCP
- Working together encouraged

Final Project (40%)

Project

- Group project
- Identical to our previous Data Science Entrepreneurship project
- 5-minute demo presentation during last class session
- More details, with rubric, will be in Canvas

Final Project Milestones

- 4 milestones will be due throughout module
- Included as part of final project grade
- Details in Canvas

Course Resources

- Office Hours: T/Th 9:00-10:00 in Room 605, or by appt
- Canvas: for sharing lectures, demos and labs
- Two Slack channels:
 - **#2025-msds-603-discussion** for discussion question participation
 - **#2025-msds-603-general** for general course Q&A and announcements
- GCP Credits (\$50) will be provided for labs and project

Path of Least Resistance

- Let's not get bogged down by technology – plenty of time for spinning your wheels when you start working at a company.
- We will use **simple, easy to set up, tools** (for the most part).
- Meant to be a fun interactive class – if you want to explore other tools, or playing more with GCP, you are free to do so, but is not always required.

Resources for Learning

- Slides and demo notebooks
- Two books in our library related to MLOps:
 - [Effective Data Science Infrastructure](#) *by Ville Tuulos*
 - [Designing Machine Learning Systems](#) *by Chip Huyen*
- Other resources I will give you during each lecture

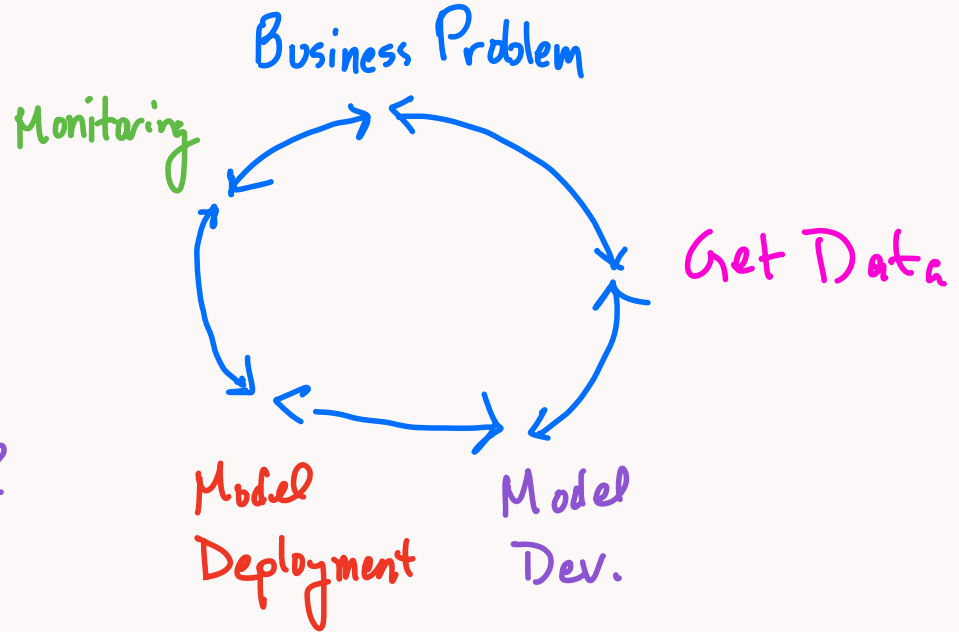
What is MLOps?

What is MLOps?

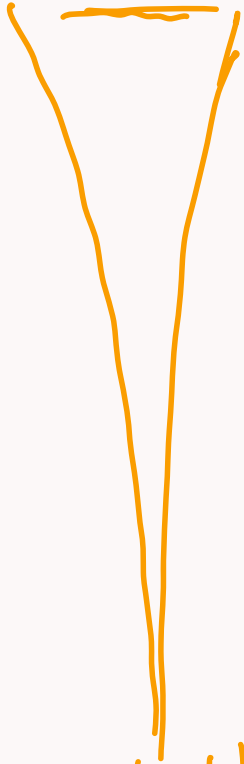
- Develop, deploy and maintain ML applications - <https://madewithml.com/#mlops>
- Set of best practices for putting machine learning models into production - <https://github.com/DataTalksClub/mlops-zoomcamp>
- DevOps (software apps) and MLOps (ML apps) are both about streamlining processes
 - MLOps is more complex than DevOps, though.

The MLOps Pipeline/Lifecycle

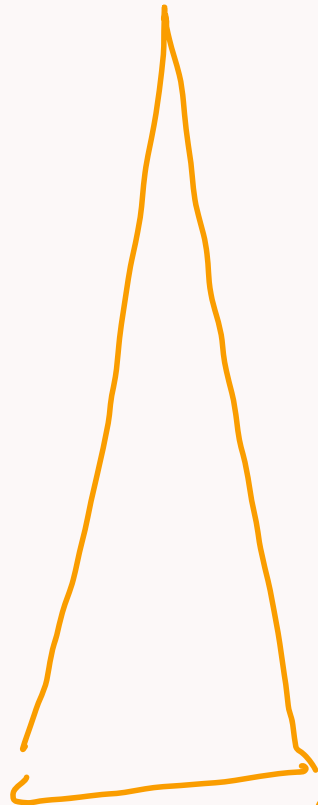
- What is goal?
- Do we need ML/AI?
- Do we have data?
Where? Labeled?
- Do we have infra?
How to keep track of experiments?
- Can we put into production?
Is it scalable, secure, etc.?
- Does the data drift?
Should we retrain? Explain predictions?



Infrastructure



How much the
DS cares.



How much infra needed?

Model training
Feature Engineering
Model Operations
Architecture
Versioning

Compute Resources
People
Data storage

A Side Note about Titles

- **Data Scientist** – develops and prototypes models/solutions, looks for insights (analytics). A sometimes all-encompassing title for some below roles.
- **ML Eng** – implements the model in a scalable, production-ready way
- **Data Eng** – sets up data pipelines for input/output data
- **DevOps Eng** – deploys applications in prod and maintains them
- **Infra/Platform Eng** – provide general pieces of infra (data warehouse, compute platforms) for many applications to use
- Product/Program/Portfolio/Project managers, business owners, etc.
- *AI Engineer - implement AI products*

Current Trends

- ML development is getting easier:
 - CV and NLP has tons of pre-trained (**foundation**) models ready for use
 - Huggingface being the most popular example
 - Azure, GCP, AWS offer ml-as-a-service (MLaaS)
 - Though these tend to also be NLP and CV-based, there are some that do time series forecasting and anomaly detection
 - AutoML can do a lot of the tedious work for you
- MLOps has some notable toolkit offerings, some of which we'll be exploring in this course

Back to our example!

What if we were asked to build a Smart Transportation System?

- The SFMTA wants to implement a smart transportation management system to optimize traffic flow, reduce congestion, and lower emissions. Using data from traffic cameras, vehicle sensors, public transit GPS, mobile apps, weather services, and scheduled events, the system would predict traffic patterns and adjust traffic light timings, suggest route alternatives, and manage MUNI schedules. The system should be able to adjust to unexpected events, such as collisions, and ensure equitable service across neighborhoods. It should be live 24/7 and demonstrate measurable improvements in commute times and air quality.



Get in groups and discuss the following:

Discuss three potential issues when designing and implementing the Smart Transportation System? Think of one issue for each of these categories:

- Technical: the ML/AI model
- Technical: the development of the system
- Non-technical: anything related to safety, personnel, operations, politics, sociological, etc..

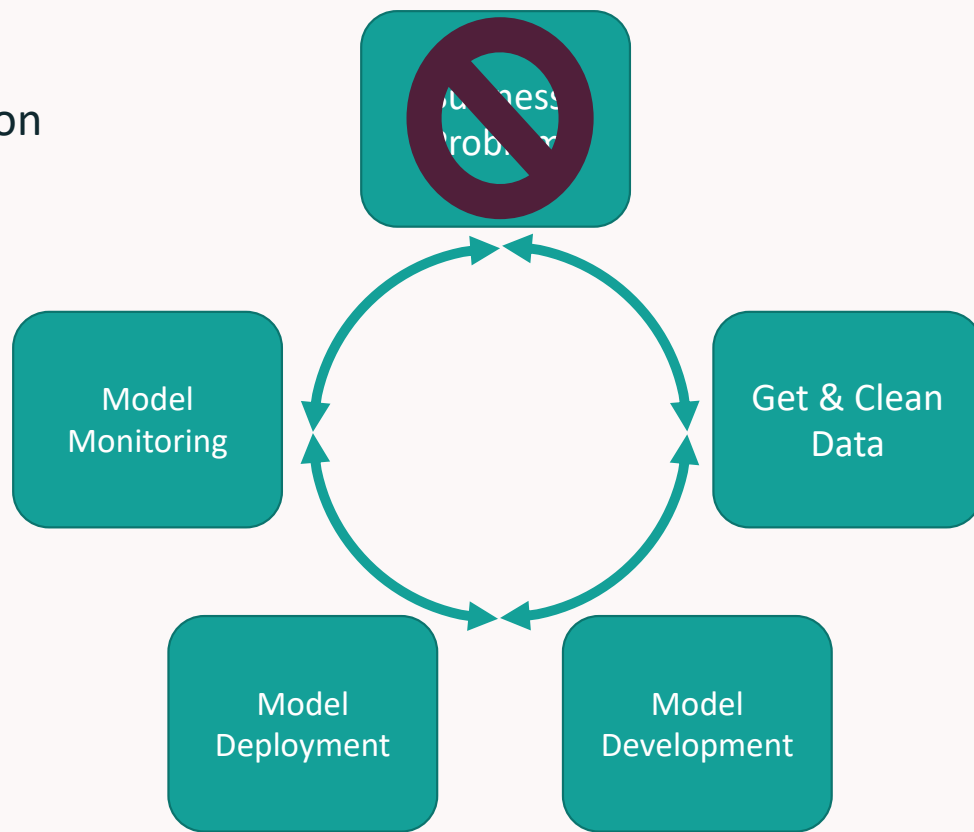
Write your answers as three bullets, and tag your group members, in **2025-msds-603-discussion**

Tools

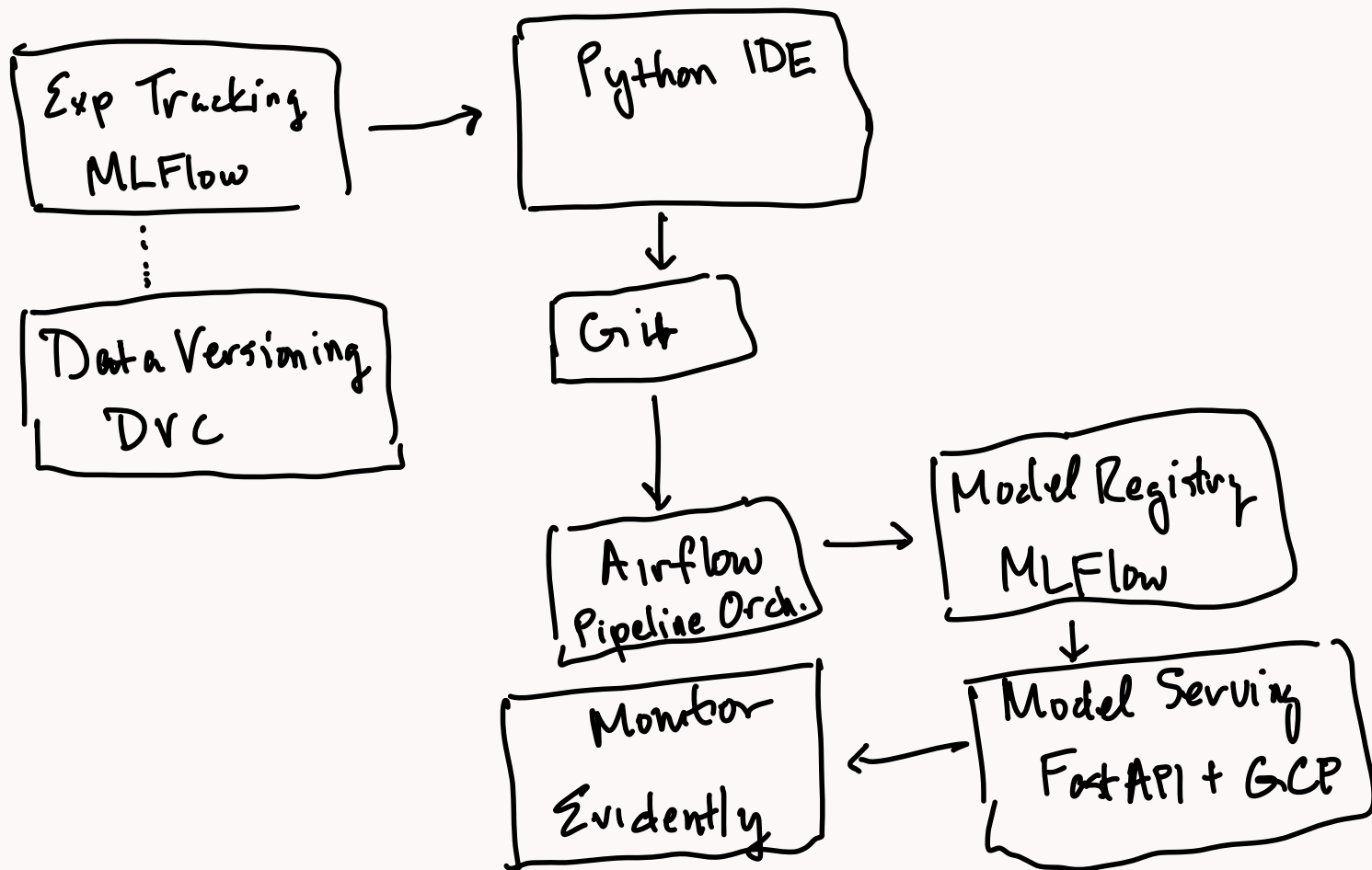
MLOps Pipeline Revisited

1. Experiment tracking
2. Artifact/model tracking & registration
3. Code versioning
4. Data versioning
5. Data quality
6. Data & Feature stores*
7. ML Orchestration
8. Orchestration & Scaling
9. CI/CD/CT
10. Model serving
11. Model monitoring
12. Model explainability/fairness*

*Limited coverage in this class



The MLOps Stack



MLOps Pipeline Tools

1. Model Dev - python (local and GCP/Colab)
2. Exp tracking - MLFlow (local and GCP)
3. Model registry - " "
4. Code versioning - Github
5. Data versioning - DVC (local)
6. Data Quality - Great Expectations (local)
7. Data + Feature pipeline - dbt
8. ML Orchestration - Metaflow (local + GCP)
9. Orch + scaling - Docker, K8s, Google cloud run, Terraform, Argo

So. Many. Tools

CI/CD/CT - Actions
Model Serving - FastAPI + Streamlit
Monitoring - Evidently

The MLOps tool that we choose isn't important,
it's the *why* we use it that matters.

Model Development Environment

Environments



DEV

Environments

STAGE



The diagram consists of a large, light pink rectangular area. A vertical dark purple line runs from the top to the bottom of this area, positioned approximately one-fifth of the way from the left edge. The word "STAGE" is written in bold, black, uppercase letters above the line, near the top right corner of the pink area.

Environments



PROD

Optimal Development Environment

- Optimizes for two things:
 - Rapid prototyping of models
 - Interaction with the production environment
- Laptop vs Cloud
 - Cloud can be an on-prem "cloud" or true cloud instance (AWS, GCP, Azure, etc.)
 - Laptops are not: secure; scalable; like the production environment; easy to monitor and support

IDEs and Notebooks

Virtual Environments



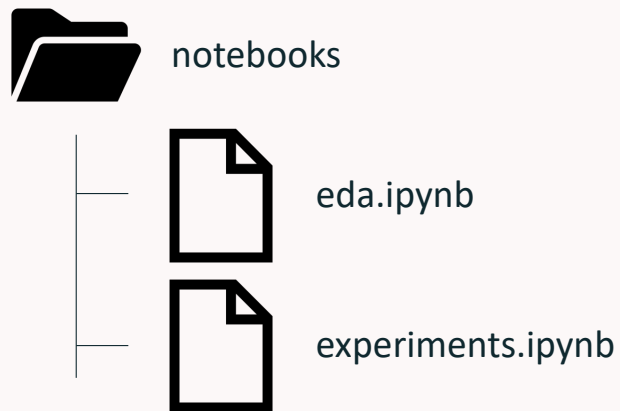
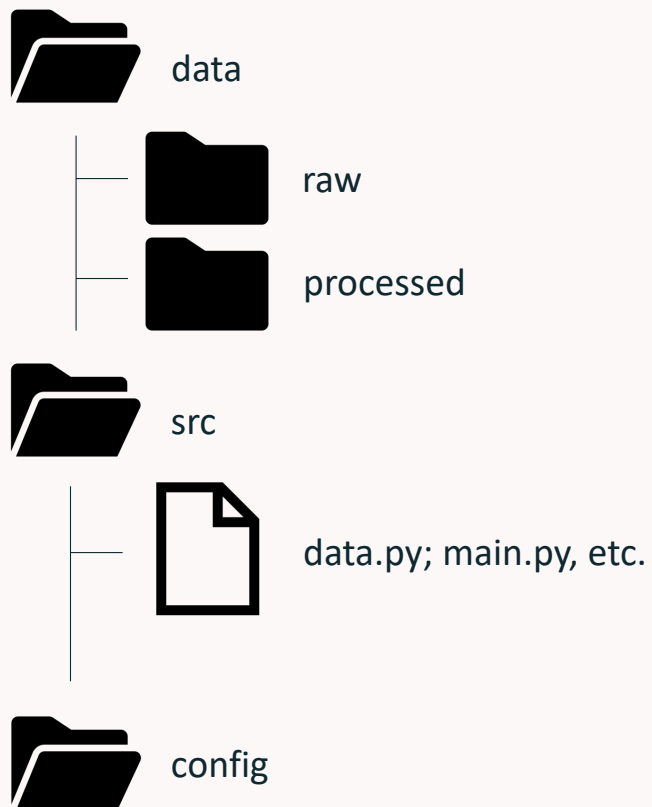
requirements.txt

python libraries and
versions

setup.py or
pyproject.toml

Project metadata
and instructions for
setting up
environment

Project Organization



Do what feels comfortable for you, I am not going to be **prescriptive**.

Project Documentation

Be explicit

```
def get_random_ingredients(kind: array=None) -> array:
```

```
    """
```

```
    Return a list of random ingredients as strings.
```

```
    :param kind: Optional "kind" of ingredients.
```

```
    :type kind: list[str] or None
```

```
    :raise lumache.InvalidKindError: If the kind is invalid.
```

```
    :return: The ingredients list.
```

```
    :rtype: list[str]
```

```
    """
```

```
        return ["shells", "gorgonzola", "parsley"]
```

Use descriptive doc strings
(use [sphinx](#) or mldocs/mkdocstrings to
generate a [readthedocs style site](#)).

OR

Create a dedicated site for your project
with all the bells and whistles.

Dev Environment Demo and Lab