Storyline

Pathos, Ethos, Logos

- Introducing myself (Pathos) “WHOAMI”
- Sharing a bit about my background (Ethos) “BACKGROUND”
- Introducing the problem (Logos) “MOTIVATION”

Motivation to get the listeners interested

- Telling the audience about the goal of my talk, what do they get out of it “GOAL”

Concepts/Lingo, before we get started

- Software Engineering: What are DI and ABC?
- What is MLOps
- What is MLflow

Pyramid Principle - Situation, Confrontation, Message, Story1, Story2, Story3

- Reiterate “BACK TO OUR GOAL”
- Zoom in on part of the problem “CHALLENGE” - situation
- Confront these challenges “TACKLING THESE CHALLENGES” - Confrontation & Message
- Story 1: We need to standardize machine learning development and training
  (After experimentation in notebooks we need to go to proper software engineering)
- Story 2: We need to apply MLOps for CT and CM
- Story 3: We can use tools such as MLflow to apply CT and CM

Proof is in the pudding: Showcase an example

- What does MLflow look like, how does it work
- What does a training job look like with template
- How do we apply SE practices
- How does CT and CM play out here

(NOTEBOOK DEV) Experiment tracking - ml pipeline

Behoefte aan MLflow duidelijk maken door een voorbeeld te geven hoe je hyperparameters bij zou moeten houden met een notebookje. Dat dit een messy oplossing is en niet bij te houden is. En tada...Mlflow!

Experiment tracking doesn’t solve all our training problems. We still have messy Jupyter notebooks with cells that need to be executed in a specific order to get the final solution.

To solve it, we decompose the notebook into a set of building blocks executed one after another. We call a sequence of such blocks as a “machine learning pipeline”."

https://datatalks.club/blog/mlops-10-minutes.html

MLOps & MLflow

Continuous Training and Continuous Monitoring with MLflow

ANALYTIXPOWER

WHOAMI.

A little bit about myself:

- **Triathlon**
- **Motorcycles**
- **Travelling**

BACKGROUND.

Track record:

- **Economics** and **Econometrics**
- Data Science, Data Engineering, **Machine Learning Engineering**
- **Founded** a Fashion Analytics company

Currently working as a Machine Learning Engineer at **ABN AMRO** through **Analytix** Power

MOTIVATION.

Reason for this talk:

- **More than 85%** of Machine Learning Projects **fail** (Gartner)
- **Model Deployment** is intricate (**Bas** will tell more about **model serving**)
- Keeping **Model Performance** high in production is also hard

GOAL.

For me today is a success if:

**You are better equipped to
get and keep models performing well in production**

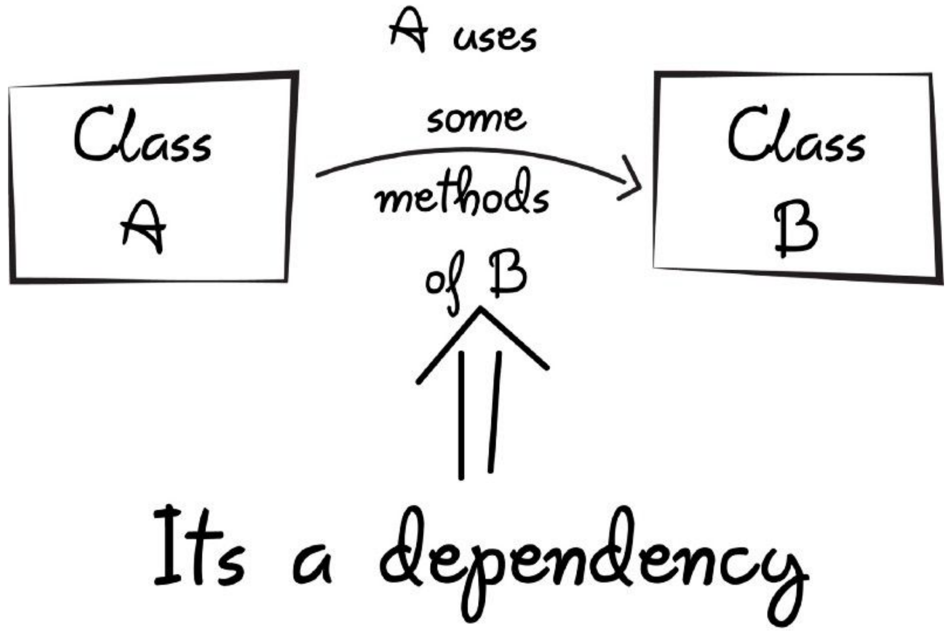
ANALYTIXPOWER

CONCEPTS.

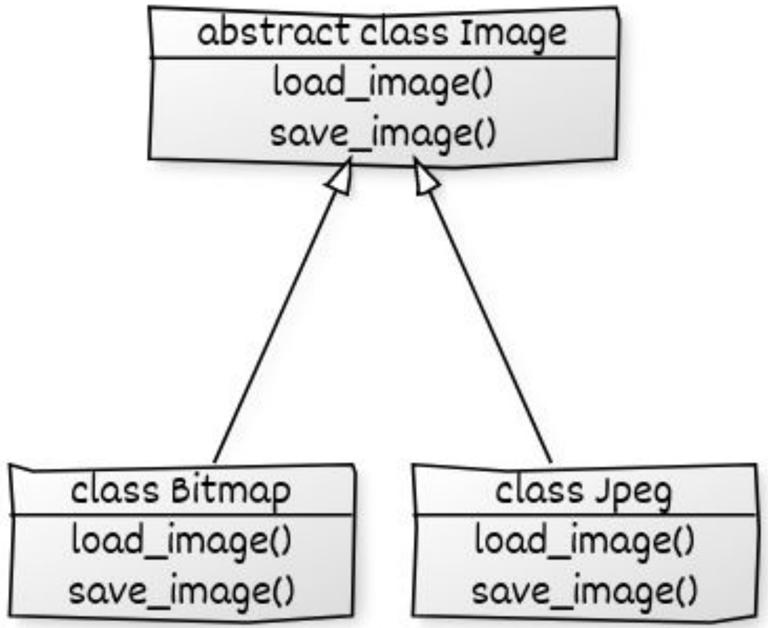
Before we continue, I want to start introduce some topics:

- **Software Engineering Concepts**
- **MLOps**
- **MLflow**

DEPENDENCY INJECTION.



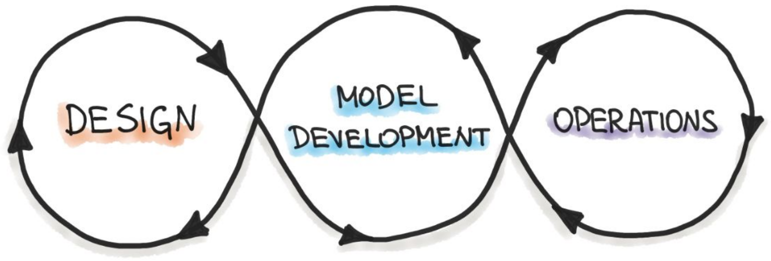
ABSTRACT BASE CLASS.



MLOPS PROCESS.

Iterative-Incremental process in MLOps

- **Design**
- **Model Development**
- **Operations**



ANALYTIXPOWER

MLOPS PRACTICES.

Another way to look at it:

- Continuous Integration (**CI**) and Continuous Delivery (**CD**)
- Continuous Training (**CT**)
- Continuous Monitoring (**CM**)

ANALYTIXPOWER

MLFLOW.

Components of MLflow

- **MLflow Tracking**
- MLflow Projects
- MLflow Models
- Model Registry

MLFLOW TRACKING.

MLflow Tracking for the Model Training Pipeline

- **Experiment**
- **Runs**
- Logging, artifacts, reproducibility, versioning, lineage

BACK TO OUR GOAL.

For me today is a success if:

You are better equipped to
get and keep models performing well in production

ANALYTIXPOWER

CHALLENGES.

Why is keeping models performing well in production hard?

- Experimental nature of model development (**Experimentation**)
- The world changes (**Data Distribution Shifts**)
- Your model changes the world (**Feedback Loop**)

TACKLING THESE CHALLENGES.

We can tackle these challenges using:

- Software Engineering
- MLOps
- MLflow

Getting and keeping models in production shouldn’t be hard!

APPLICATION.

In order to make getting and keeping models in production easy:

- We need to apply **Software Engineering** practices to the **Machine Learning Pipeline**
- We need to apply **MLOps practices**
- We should use tools, such as **MLflow**, to apply **MLOps**

ANALYTIXPOWER

C: EXPERIMENTAL NATURE OF MODEL DEVELOPMENT

Notebooks are great for experimentation, not so for Software Engineering

- Poor State Management (cells run in arbitrary order)
- Easy to avoid writing Functions, Classes, Modules
- Easy to forgo Software Engineering standards and practices

S: STANDARDIZING MODEL DEVELOPMENT & TRAINING.

We can standardize after the experimental phase

- **Design Patterns** (Factory Pattern, Strategy Pattern)
- **Design Principles** (DRY, YAGNI, SOLID)
- **Software Engineering** (From Notebook to IDE)

C: THE WORLD CHANGES.

Data Distribution Shift

Supervised model inputs X and its output Y
Training data samples of joint distribution.

$P(X, Y) = P(Y|X)\,P(X)$
$P(X, Y) = P(X|Y)\,P(Y)$

- **Covariate Shift**
  $P(X)$ changes, but $P(Y|X)$ remains the same
- **Label Drift**
  $P(Y)$ changes, but $P(X|Y)$ remains the same
- **Concept Drift**
  $P(Y|X)$ changes but $P(X)$ remains the same

C: THE MODEL CHANGES THE WORLD.

Degenerate Feedback Loop

- Retail Sales
- Song Recommender
- Resume Screener

S: CONTINUOUS MONITORING & TRAINING.

- **Data Monitoring** & **Data Drift**
- **Model Monitoring** & **Model Drift**
- **Schedule** and/or **Trigger** based **Retraining**

DEMO.

- **Server** (Storage)
- **Interface**
- **Code**

END.

That’s all folks

ANALYTIXPOWER

REPOSITORY.

The Slides & Code that was used in this talk can be found here:

Link: https://github.com/Pverheijen/i4talent-meetup-29-06-2022

GRAVEYARD

-

ANALYTIXPOWER

OLD C: MODELS COME IN ALL SHAPES & SIZES.

- **Different kinds** of models (from Linear Regression to Deep Neural Networks)
- Get developed and trained in **different ways** (technologies, orchestrators)
- And deployed at **different frequencies or triggers** (daily, weekly, monthly, performance degradation)

OLD S: STANDARDIZING MODEL DEVELOPMENT & TRAINING.

From notebook to IDE, from experimentation to software engineering

- Job as orchestration
- Training Template
- Dependency Injection
- Abstract Base Classes

Orchestrating the Model Training pipeline

- Training Template
- Extensible
- Flexible

-

ANALYTIXPOWER

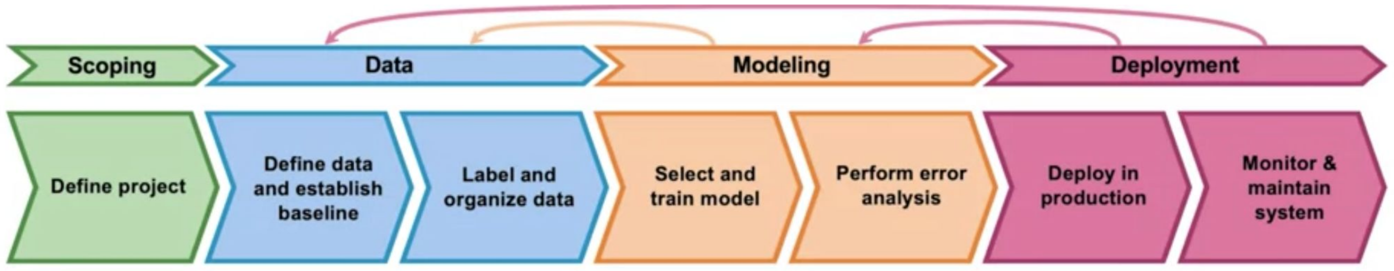
LOGGING

- Logging information
- Logging artifacts
- UI
    - Experiments (Runs, logging, artifacts, parameter comparisons) -> GS CV
    - Models (Deployment, Image, Pyfunc, flavours, env & stage)

ML LIFECYCLE.

Different components of the lifecycle: Monitor & Maintain system

- Scoping
- Data
- Modeling
- Deployment



MLOps

Why is there such a thing as MLOps?

- Getting a model into production is only the start (Data Requirements)
  - Ben Wilson Ml eng in action-> Table ETL processes
- Data Distribution Shifts
  - Chip Huyen -> Label Drift, Covariate Drift, Concept Drift
- Feedback loop
  - Degenerative, reinforcing

ANALYTIXPOWER

MLflow Tracking & Model Registry

Experiment tracking

- Experiment Tracking
- Logging & Comparing
- Model Store