TC5044 Machine Learning Operations

1.1 Introduction

POSTGRAD

MNA

Preamble

Outline

Introducing THE Team

1. Introduction to MLOps
    1. What is MLOps?
    2. MLOps Roles
    3. ML Projects life cycle
    4. MLOps Roles in process stages
    5. ML Project challenges
    6. MLOps Maturity Levels
    7. MLOps Pipeline
    8. MLOps Building Blocks
    9. Work Environments
    10. Comparison with Xops
2. Course dynamics

THE Team

Dr. Gerardo
Rodríguez Hernández
Professor AI Researcher

Maestro Ricardo
Valdez Hernández
AI Industry Practitioner

Maestra María Mylen
Treviño Elizondo
Course Manager Champion

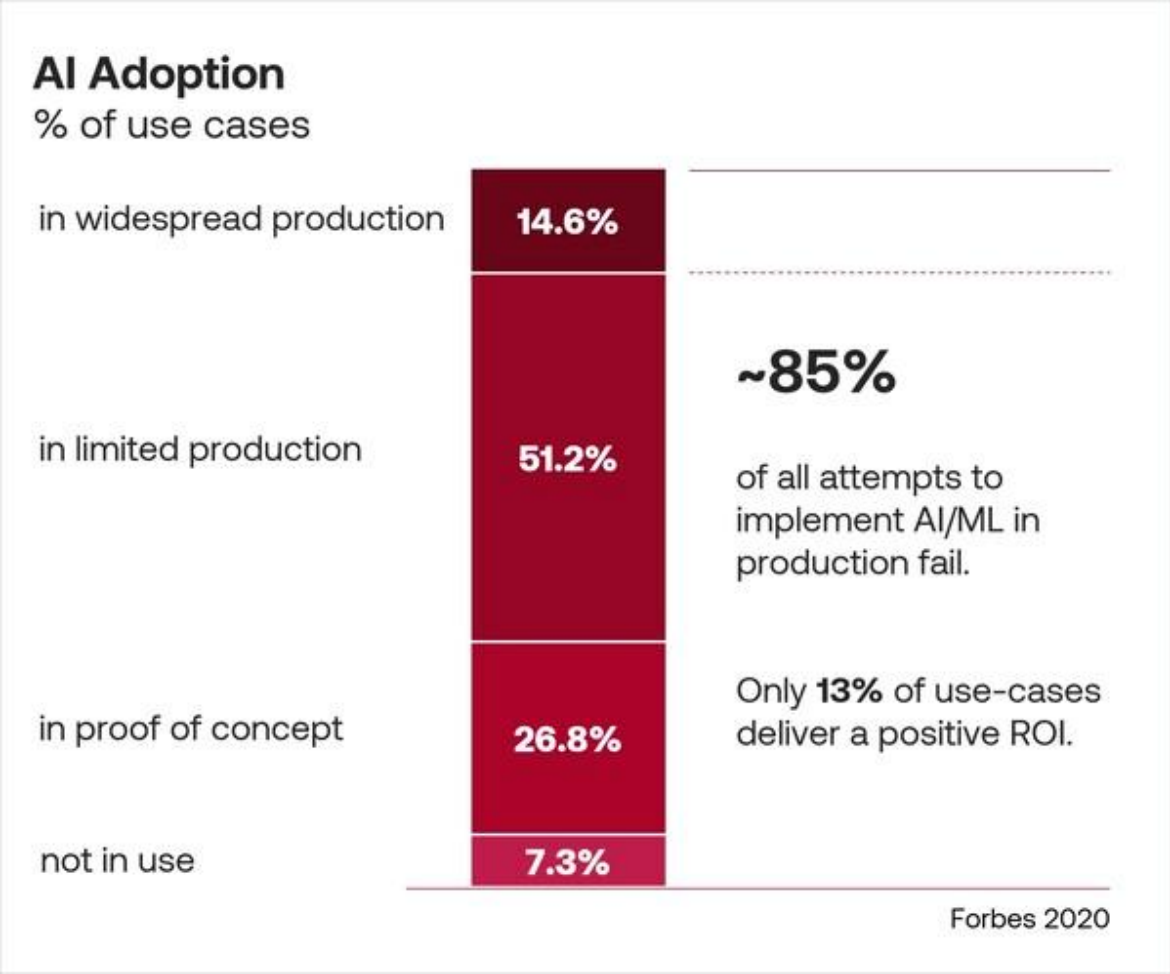
Maestro Iván
Reyes Amezcua
MLFlow Ambassador

1 Introduction to MLOPS

Introduction

- AI-based applications hold promise for the future of organizations by increasing their level of business intelligence, competitiveness and automation.
- However, despite all the technological investment and the high technical expertise of the scientists and engineers involved in these projects, few of them succeed.

AI projects success rate

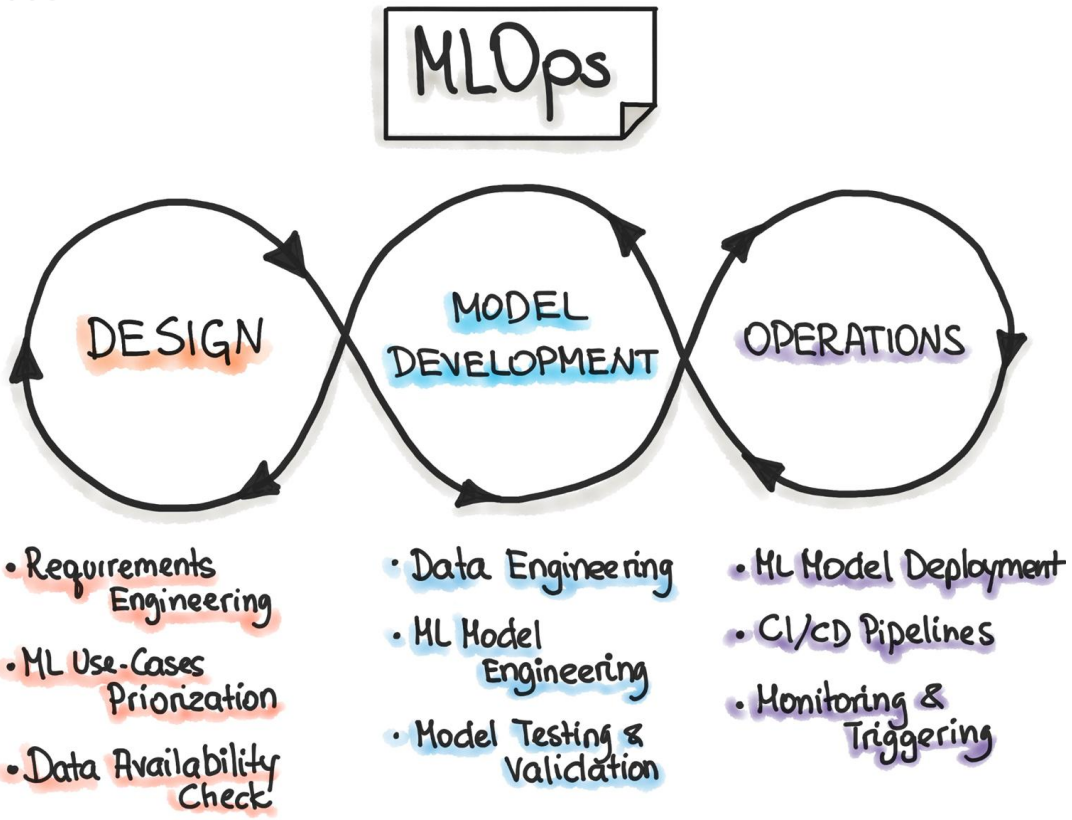


85% of AI initiatives fail (Pactera, 2017).

According to Gartner, this trend has not improved over the years.

1.1 What is Machine Learning Operations?

MLOps



It is a collaborative working methodology that seeks to encompass the best practices recommended by the industry.

Its objective is to automate the delivery of high quality software on a continuous basis in production environments.

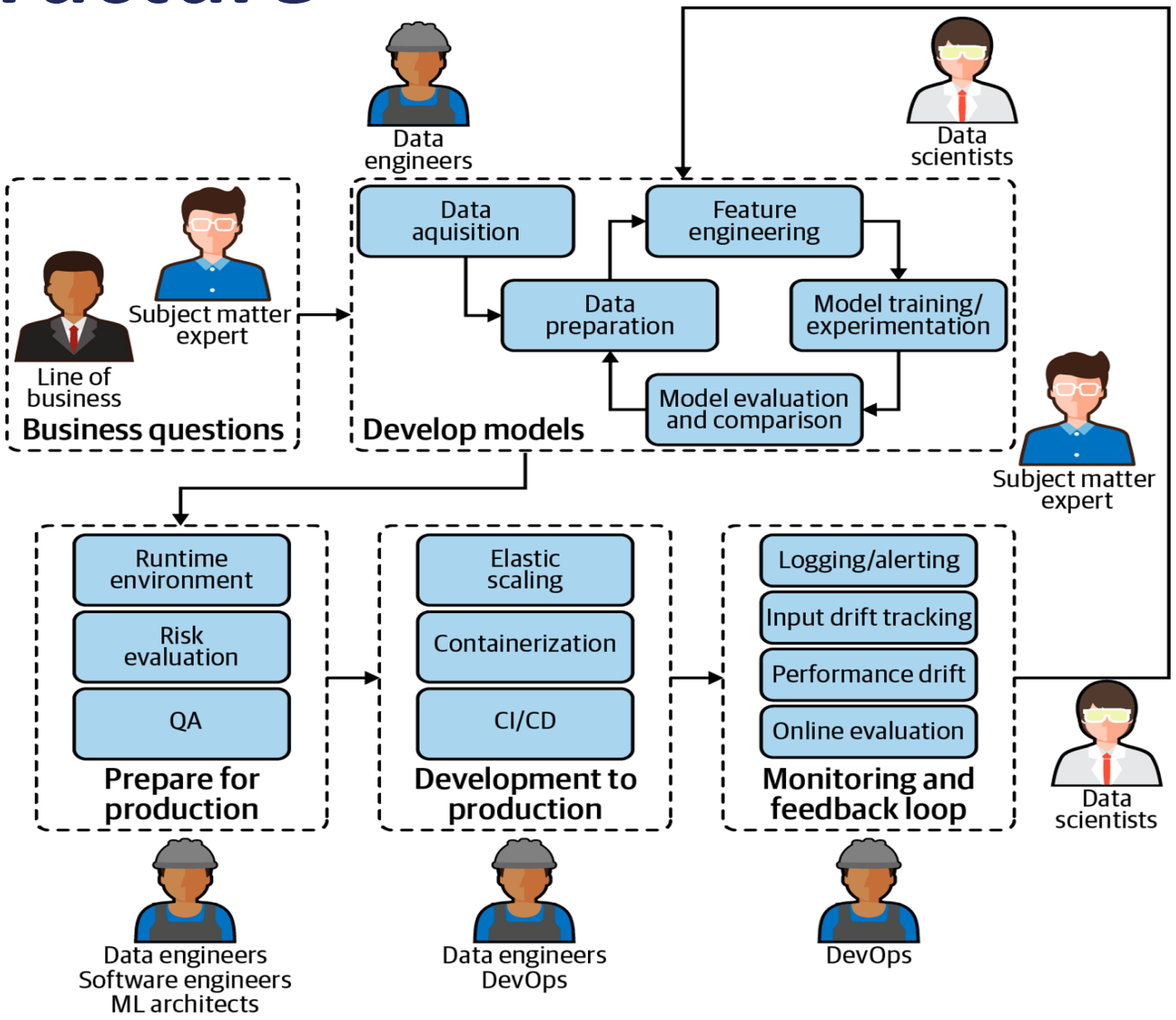
Benefits of adopting this practice

- Promotes good communication that facilitates collaboration and division of labour within a heterogeneous team.
- Ensures that the work of the team is moving in the same direction in line with the vision of the business.
- Allows problems to be identified much faster, minimizing risks to production start-up.
- Accelerates the production of ML models allowing you to experiment and make improvements much faster and with better quality.
- Ensures that ML data and models comply with required regulations

1.2 What are the roles involved in an MLOps team?

MLOps Structure

No unicorns

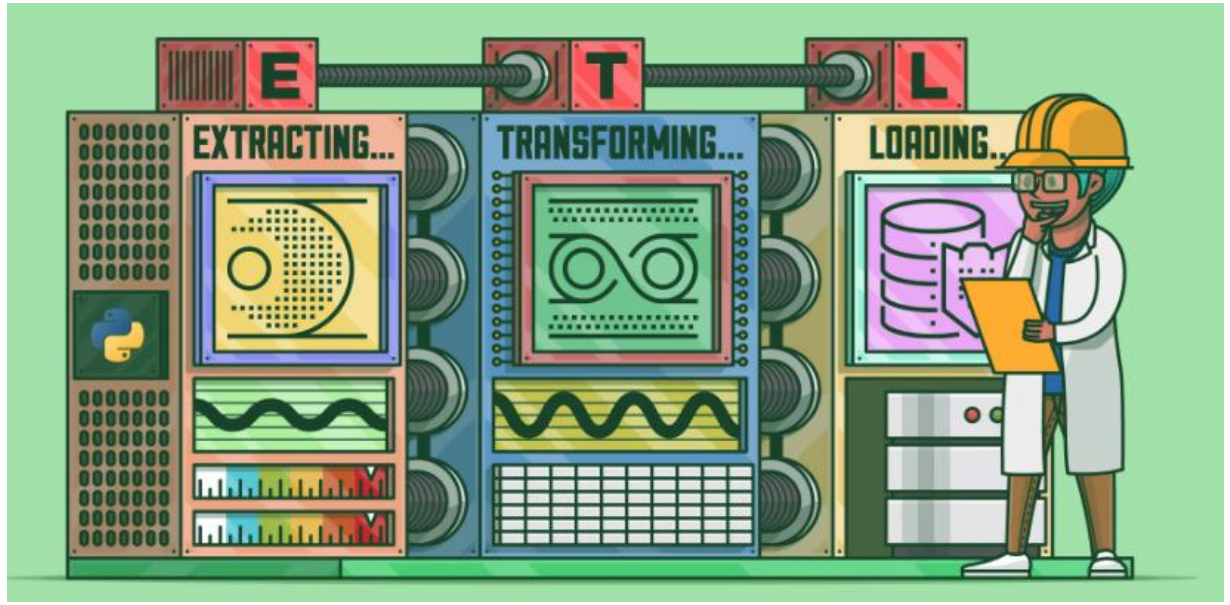


Treveil, M. and Dataiku Team: *Introducing MLOps: How to Scale Machine Learning in the Enterprise. O'Reilly, 2020.*

Stakeholders

- Owns the business strategy
- Define functional requirements and priorities
- Define the objectives to be met by the product through KPIs
- Verifies at all times that the solution adds **value**

Data Engineers



- Automates the collection, cleaning, centralization and preparation of data.
- Associated with the construction of data pipelines, data-lakes or data-warehouses that store the required data.

Data Scientist

- Design the ML solution aligned to the business requirements.
- Define the necessary data requirements.

Software Engineers

- Supports the development of solutions for the implementation of the ML model in production.
- They also perform side tasks (modification and instrumentation of existing software to generate new data sources or enable the integration of other services that can consume the results produced by the main service of the solution).

ML Engineer

- Ensure a scalable and flexible environment for ML model pipelines, from design to development and monitoring.
- Introduce new technologies when appropriate that improve ML model performance in production.
- High-level overview of models and their resources consumed.
- Ability to drill down into data pipelines to assess and adjust infrastructure needs.

Site Reliability Engineer (DevOps)

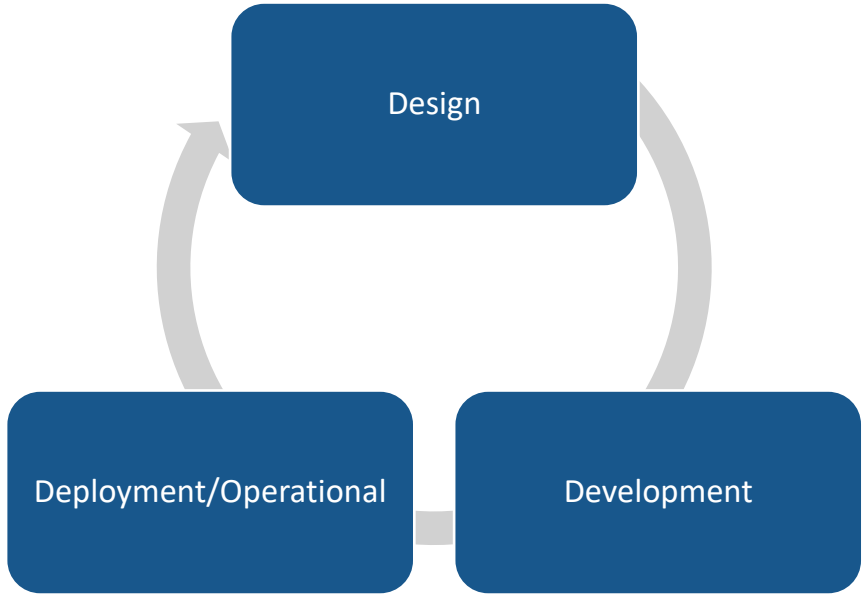
- Conduct and build operational systems and test for security, performance, availability.
- CI/CD pipeline management.
- Seamless integration of MLOps into the larger DevOps strategy of the enterprise.

1.3 What is the lifecycle of an ML project?

ML Project Lifecycle

From a high-level perspective, we can consider that Machine Learning projects involve three fundamental phases.

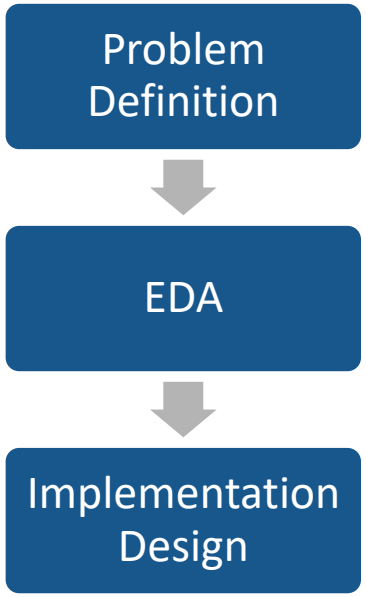
Where each of those is a cyclic process in which we iterate several times between them.



Design Phase

At this point, we must consider the following:

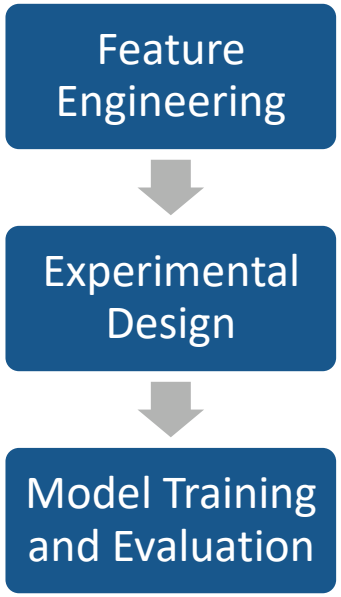
- Understand the context of the problem.
- Analyze business requirements
- Validate that the problem could be solved with Machine Learning.
- Identify those KPIs that allow us to track the progress of this initiative
- Identify the data needed to solve the problem and ensure its quality.



Development Phase

In this stage, the essential things to consider are:

- Implementing the ML pipeline.
- Combine different data and algorithms.
- Choose the best model that is going to be deployed



Deployment or Operational Phase

The main objectives are:

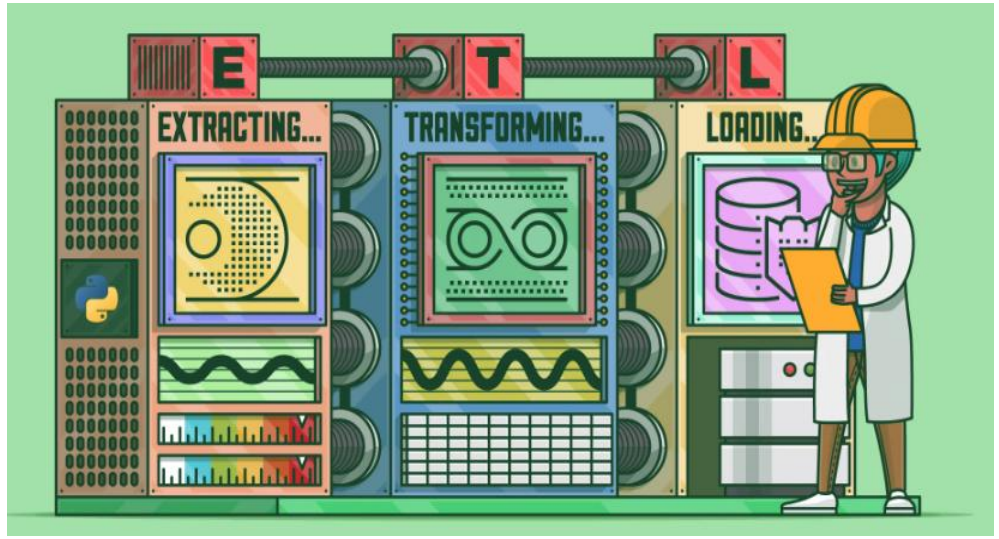
- Integrating the ML pipeline into regular business processes, considering tools and services that facilitate seamless integration.
- Track and monitor the behavior of the ML pipeline that was deployed.

1.4 How are roles involved in the stages of projects?

Stakeholders

- Design.
  - Problem and requirements definition.
- Development.
  - Ensure, from a business perspective, that the initial results of the model align with expectations.
- Deployment.
  - Review that the final goal of the project is what was expected during the monitoring phase.

Data Engineers



- Design.
  - Participates in the design of solutions for data ingestion and processing.
- Development.
  - Collaborates in the feature engineering process, validates business rules, and supports data scientists.
- Deployment.
  - Ensures data integrity through monitoring.

Data Scientist

- Design.
  - Performs EDA
- Development.
  - Leads all model training and evaluation tasks.
- Deployment.
  - Ensures deployed model predictions align with business rules.

Software Engineers

- Design.
  - Collaborates in solution design to ensure code quality.
- Development.
  - More focused on code standards, it supports data scientists in this process.
- Deployment.
  - Packages and deploys the model into production.

ML Engineer

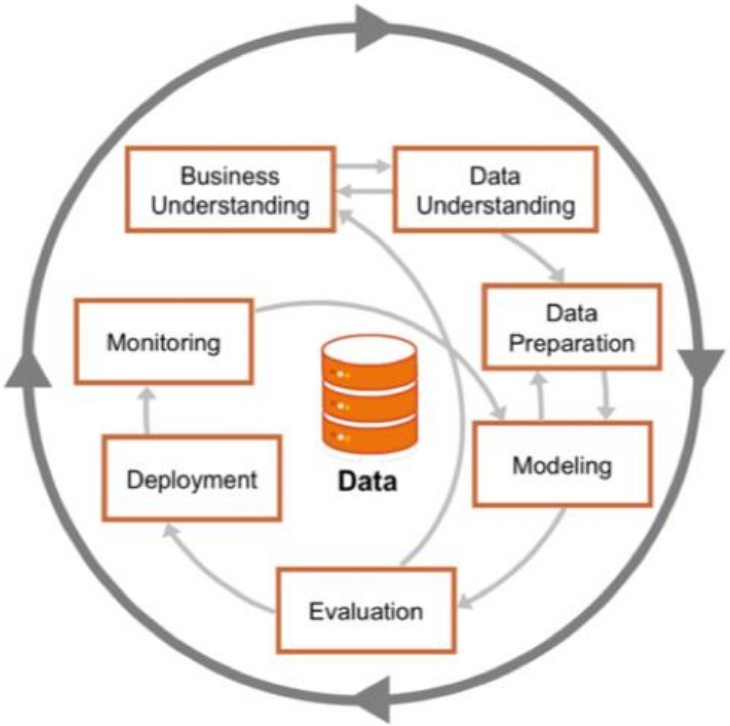
- Design.
  - Designs ML pipelines with scalability and maintainability in mind
- Development.
  - Builds and optimizes training, validation, and inference pipelines.
- Deployment.
  - Automates deployment, monitors model drift, and manages retraining strategies.

Site Reliability Engineer (DevOps)

- Design.
  - Design the infrastructure required for development and deployment.
- Development.
  - Configures services to support the pipeline execution.
- Deployment.
  - Ensures infrastructure reliability, scaling, and monitoring.

1.5 What are the challenges of an ML project?

How ML Development is traditionally made

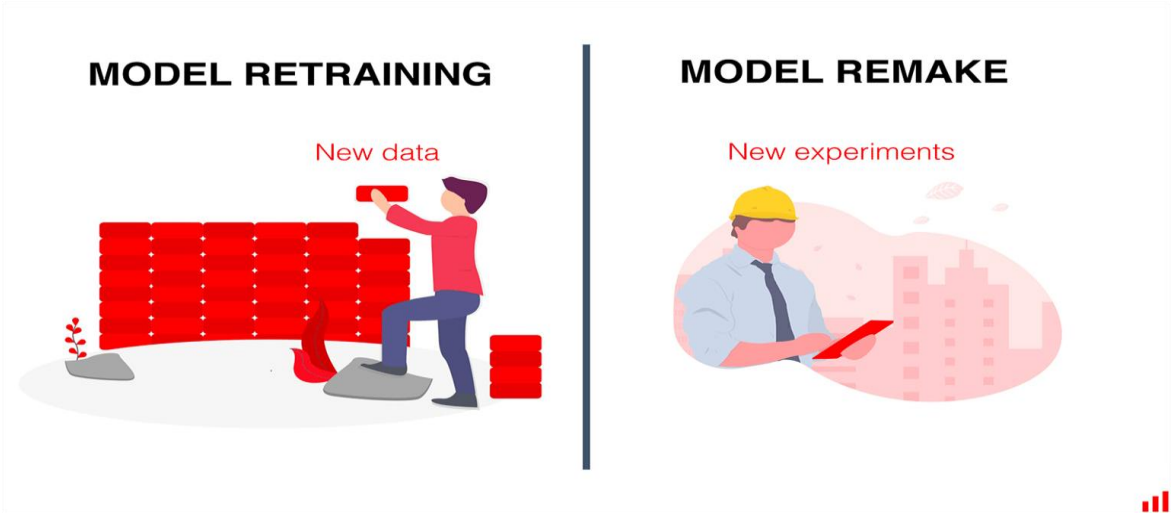


- Iterative process that is carried out in isolation as a series of experiments.
- Manual execution of data extraction and transformation, and model training, evaluation, packaging, and deployment.

Challenge 1

Organization of experiments, deployments to production, monitoring of algorithms in production, and versioning of code and data

MLOps seeks to optimize system maintenance and performance in the system deployment, monitoring, and retraining stages by automating the training pipeline.



Challenge 2

Mitigate technical debt

Unlike traditional software development, the technical debt present in ML systems is not only present in the code, but throughout the system.
The different forms of debt in these systems are due, in large part, to the use of ML data and algorithms for the development of the system.

Challenge 2 (cont.)

Mitigate technical debt

Examples include:

- Interdependence: All components have an effect on the training of the system.
- Correction cascades: during model training, corrections are chosen in certain cases. These corrections can skew the metrics under which the model is optimized.
- Unstable data dependencies: The distributions of some data are unstable, that is, they can change behavior over time.
- Feedback loops: Over time, when updating an ML system, its behavior is influenced by the system itself.

Challenge 2 (cont.)

Mitigate technical debt

To mitigate technical debt in ML systems, it is possible to use the already known software development techniques:

- Code Refactoring
- Implementing unit tests.
- Code cleanup methods.
- Documentation of the processes.

In addition, depending on the scenario, the methods of selecting variables, hyperparameters, samples, etc., can be strengthened, the metrics of the models and data distributions can be monitored and, as far as possible, processes can be automated.

1.6 Maturity levels of an ML implementation

	level		level	
Microsoft	0	No MLOps	0	**Google**
	1	DevOps, no MLOps		
	2	Automated training	1	
	3	Automated model deployment		
	4	Full MLOps Automated Operations	2	

https://www.compact.nl/articles/operationalization-of-machine-learning-models-in-audit-innovation-projects/

Playground (Level 0)

Getting started with Kaggle:

- Clean, curated data
- It's a safe environment to start modeling.
- The data is already structured, labeled, and often cleaned.
- There is limited flexibility regarding the questions that can be addressed, but the labels are already defined, and the characteristics are obvious or easy to transform.

Blank canvas (Level 1)

- There are no tags or features; You need to create them before you can start training.
- It is essential to perform a thorough Exploratory Data Analysis (EDA) to understand the data, constraints, and begin experimenting in an agile, highly mutable environment.
- Base ML models such as regressions (linear/logistic), tree-based sets, and SVMs are a good starting point.

A model in production (Level 2)

- A model has finally been deployed into production.
- It is essential to establish audits to measure your performance over time.
- These measures require evaluation metrics such as recall or accuracy, and the results are recorded in a database for future retraining.

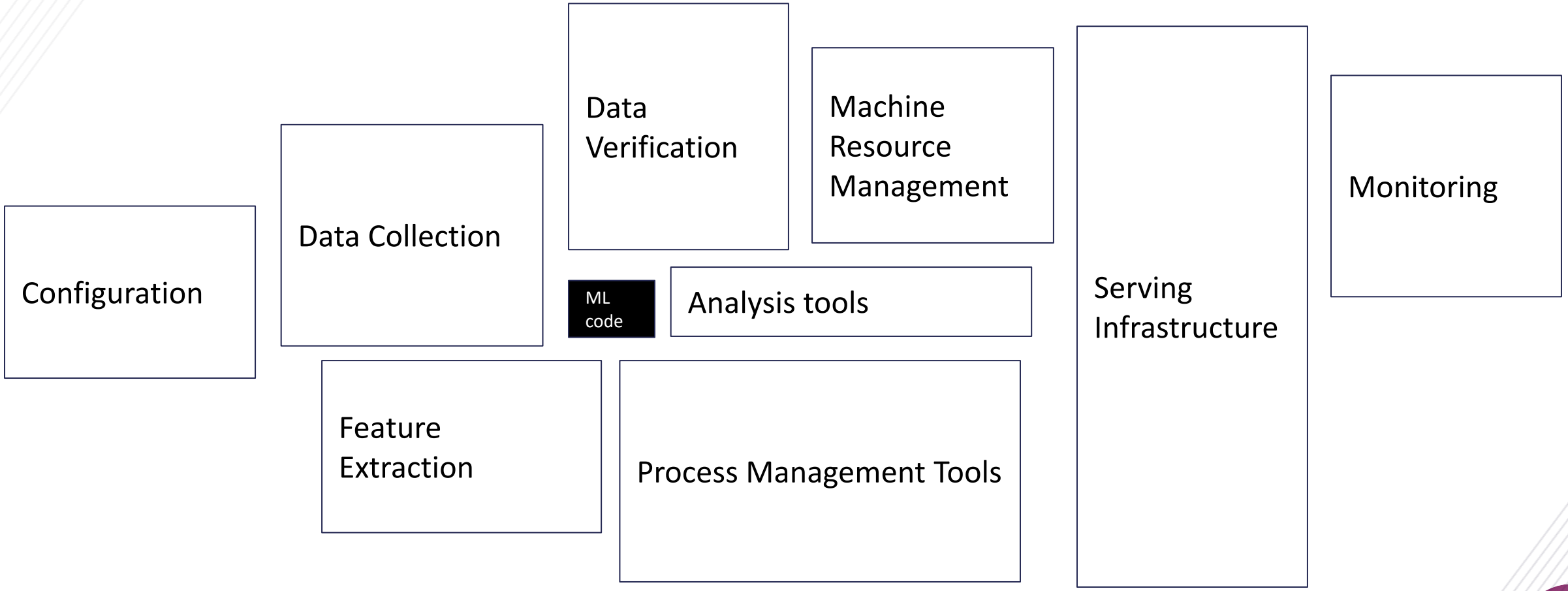
Manual updates on the model (Level 3)

- The world evolves and with it data (concept drift).
- If in a predefined interval the performance falls below a threshold, it is necessary to retrain the model with new data:
- Compile a new set of human-labeled training: the more rigorous, the more expensive.
- Use existing audited label records: less expensive and effective if the number of samples is high.
- Combine predicted and audited tags as a training set. Using Data Version Control (DVC)?

Autopilot (Level 4)

- The ML team deploys a large number of models.
- Maintaining all of these models manually can be overwhelming and quickly generate technical debt.
- The process is automated by reaching high maturity.
- Manual maintenance of the model is performed only when there is a fault, allowing more time for research and development.

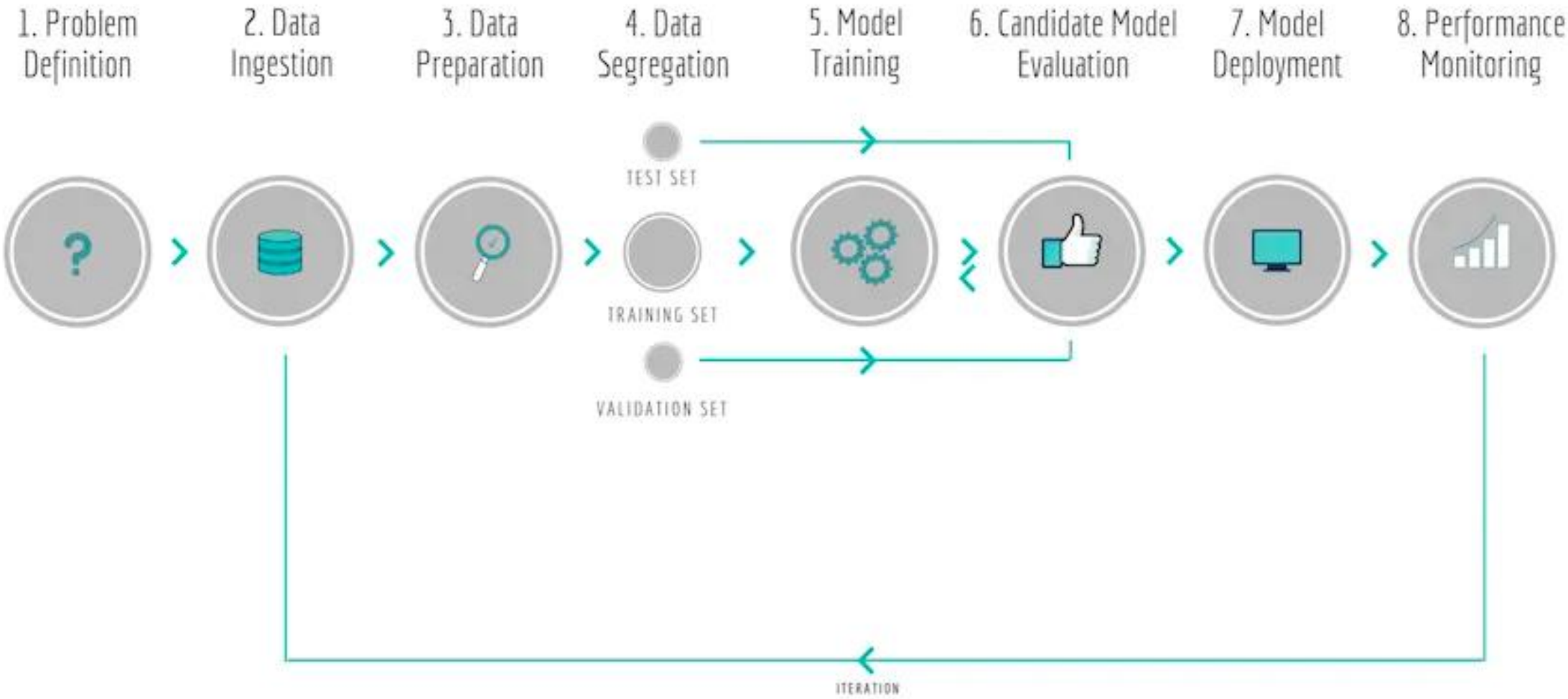
Structure of a Project Autopilot (Level 4)



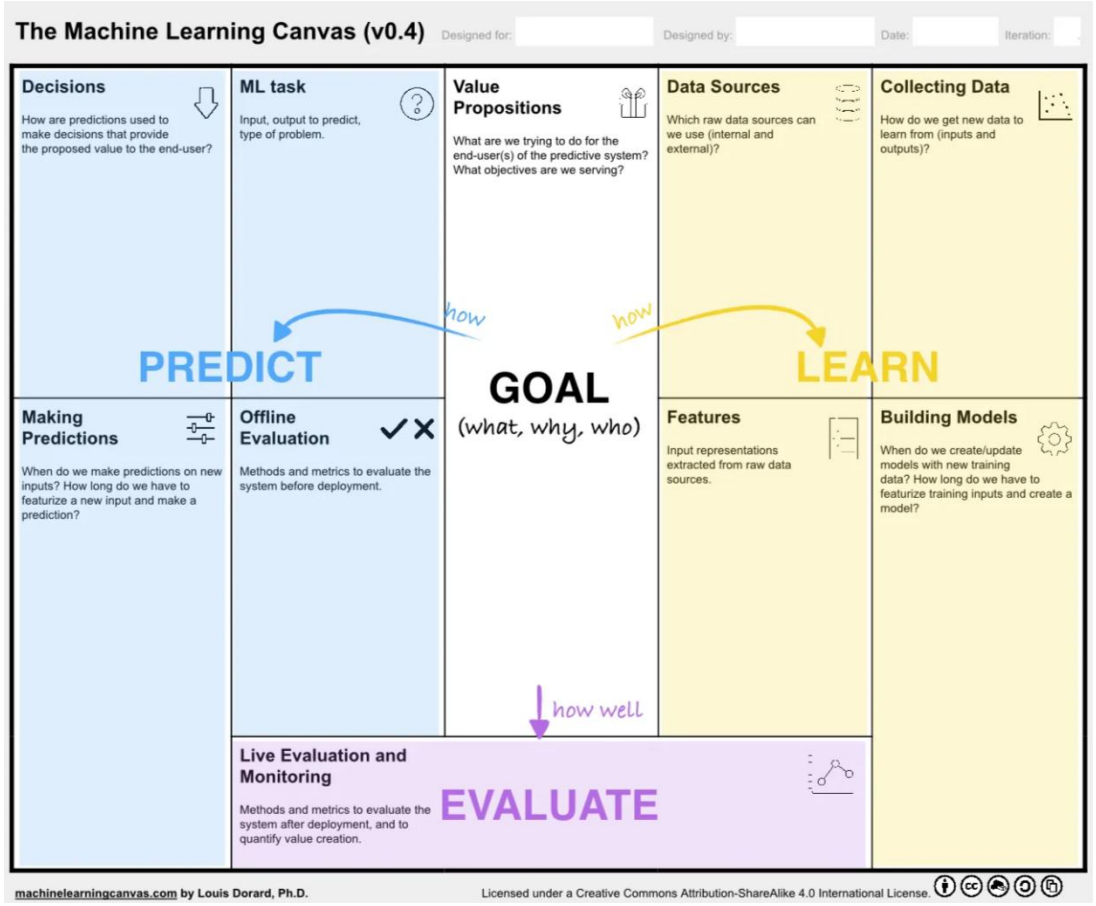
https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

1.7 Designing an ML pipeline

MLOps Pipeline



1) Problem definition (MLCanvas)



- Is common that there exist a disconnection between the people who can build accurate predictive models, and those who know how to reach the organization objectives. The ML Canvas allows to explain this in a clear manner.

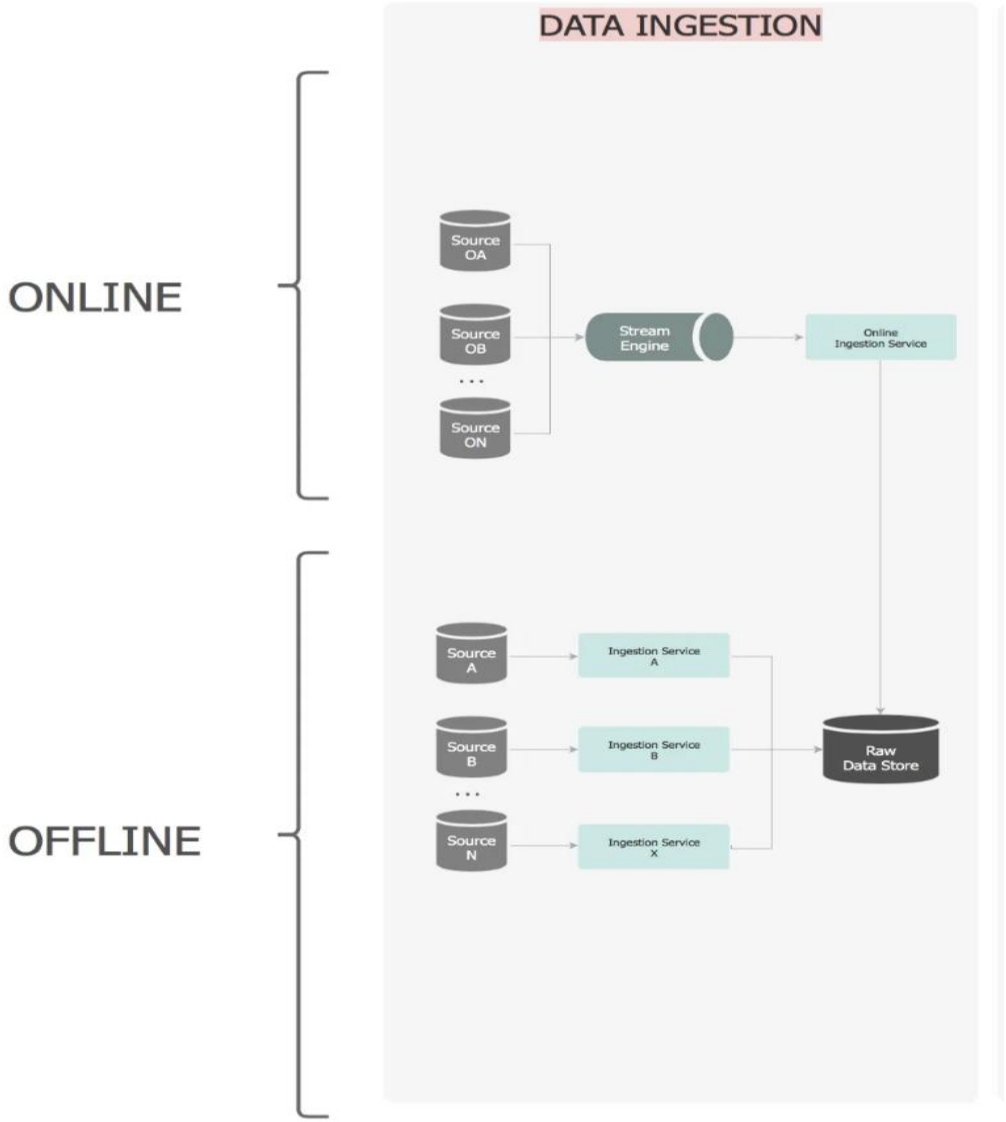
1) Problem definition (MLCanvas) cont.

- Value proposition: What we are trying to achieve? Why it is important and Who is going to use it?
- ML Task: Which type (classification, regression, etc.), What is the input? What is the output?
- Decisions: How are predictions used to make decisions that provide the proposed value to the end user?
- Making predictions: When do we make predictions on new inputs and how long do, we have for that?
- Offline evaluation: Which methods and metrics can we use to evaluate the way predictions are going to be made and used, prior to deployment?

1) Problem definition (MLCanvas) cont.

- Data Sources: Which raw data sources can we use?
- Collecting data: How do we get new data to learn from?
- Features: Input representation to extract from data sources
- Building models: When do we create/update models with new training data and how long do we have for that?

2) Data ingestion



- Offline: Independent pipelines dedicated to passing information to the Raw Data Store.
- Online: The streaming engine is responsible for passing the information to the Raw Data Store and also to the next layers that need it.

3, 4) Data preparation and segregation

- Offline: At the end of the ingestion pipeline(s), this pipeline is activated. Features are produced and saved in the Feature Data Store (for the purpose of maintaining them for future model training).
- Online Feature Data Store (in order to obtain low-latency queries for real-time predictions).

5) Model Training

- Training every 2 hours? once a day? Once a month?
- Dedicated pipeline for each model?
- Get the initial configurations, the parameters learned, the metadata about the training set, the training times, etc. and save them in the Model Candidate Data Store

6.1) Evaluation of candidate models

- The Model Evaluation Service sends a request to the Data Segregation API to get the test set and apply the evaluation metrics to each model in the Model Candidate Data Store. Evaluation metrics are stored in a repository.
- Hyperparameter optimization and regularization techniques can be applied.
- The best model is marked for deployment.

6.2) Model Scoring

The Scoring Service prepares the data, generates the variables, and searches for the extra variables from the Feature Data Store. When predictions are generated, they are saved in the Score Data Store and sent to the customer.

7) Deployment of the best model

- Via Continuous Delivery:
  - Model files are packaged.
  - The model is validated through tests.
  - The model is deployed, for example, by means of a container.

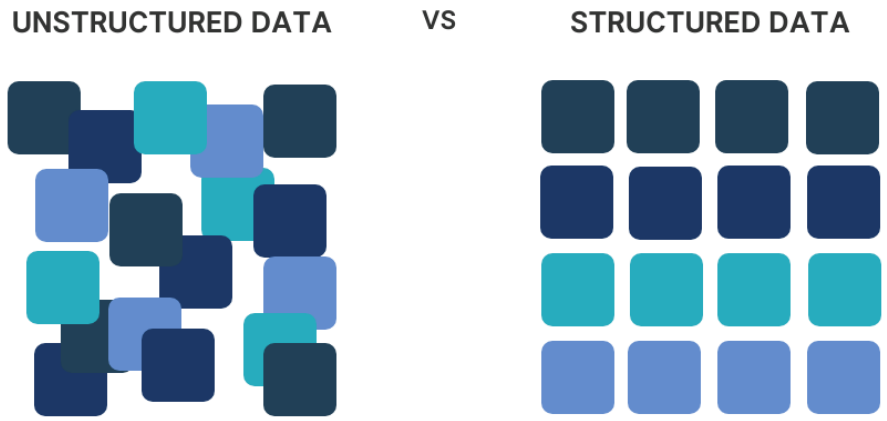
- Both offline and online models should behave as closed as possible.

8) Performance logging and monitoring

- Model ID
- Model deployment date
- Number of times the model has served
- Distribution of variables used
- Predictions vs. observations (error)
- Prediction latency

1.8 Building blocks of an ML pipeline

Data

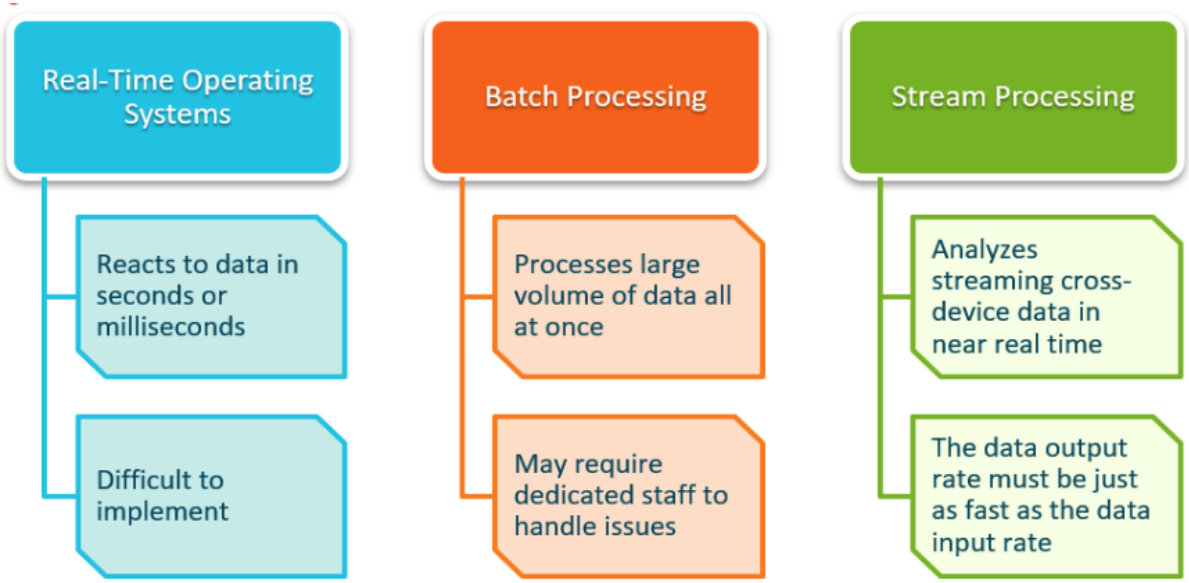


Data types according to their structure

- Structured: data that can be stored in a relational database (tables).
  Examples: CSV, TSV (ref)
- Unstructured: Data that does not have an associated data model.
  Example: photos, videos, audio, PDFs (ref)
- Semi-structured: data that has some defining characteristics but does not conform to a structure as rigid as that expected in a relational database.
  Examples: JSON, XML(ref)

Data

Types of data processing according to their speed



- Batch processing: A large volume of data is processed sequentially; processing time can take anywhere from seconds to hour.
- Near real time processing: Information is processed at a guaranteed rate, with delays varying between 1 and 10 seconds.
- Low latency processing: information is processed with a delay on the order of milliseconds.

Big Data

Big data generally refers to data that meets the principles established by the 3Vs:

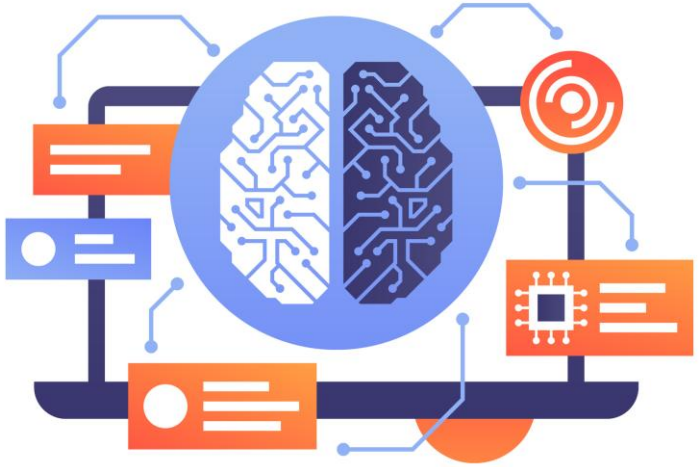
- Volume: You have a large volume of data (Terabytes, Petabytes)
- Velocity: Data flows and is created quickly. Some systems operate in real-time or near-real-time
- Variety: There is a high diversity of data types (structured, unstructured, and semi-structured).

Depending on the bibliography, other concepts can be kept in mind, such as:

- Veracity: Consistency and quality of data
- Value: Data must have or generate value for the organization

Models

- Are the core of machine learning workflows, representing the algorithms trained on data to make predictions or classifications. In an MLOps framework, the focus is on creating, validating, deploying, monitoring, and updating models to ensure their performance aligns with business goals and technical requirements.



Data Governance

It refers to the process of managing the availability, usability, integrity, and security of data in an organization's systems. Its principles are as follows:

- Data Integrity and Quality: Ensuring that data is accurate, reliable, and consistent over time and across systems.
- Transparency: Processes and activities related to data management should be clear and open to review.
- Accountability: There must be designated individuals or teams, such as data custodians or stewards, responsible for the quality and correct use of data.
- Data Protection: Ensuring data privacy and security, complying with relevant regulations (such as GDPR in Europe).

Data Governance

- Audit and Compliance: Have processes in place to review and ensure that policies and regulations are being followed.
- Standardization: Adopt consistent standards for data-related definitions, metrics, and processes.
- Availability: Ensuring that data is available to those who need it, when they need it.
- Architecture and Technology: Use technical and architectural solutions that support efficient data management and access.

Data Governance

- Training: Provide employees with the necessary training on how to handle and use data correctly.
- Change and Evolution: Data Governance is not a one-size-fits-all project; It is an ongoing process that must adapt and evolve over time and as the needs of the organization change.
- Stakeholder Engagement: Involve all stakeholders, from senior executives to end users, in the definition and implementation of governance policies.
- Business Value: Data governance must align with the organization's strategic objectives and demonstrate business value.

Data Bases

A database is an organized collection of information; requires a Database Management System, that is, a software program that provides a way to store and retrieve information in a practical and efficient way.

Types of databases:

- Relational databases (Mysql, PostgreSql).
- NoSQL databases (MongoDB, CouchDB).
- Graph databases (Neo4j, Neptune).
- In-Memory databases (Redis, Memcached).
- Time-Series databases (InfluxDb, Prometheus).

Web Services

A web service is a self-contained piece of software available over the internet via http; It usually transmits messages in JSON or XML format. They can in turn communicate with other web services or directly with customers.

Within an ML project, web services can be used in different ways, for example:

- Data Monitoring and Ingestion
- Reception of user-generated events when browsing a website (time spent on the page, clicks on the page, reading position, navigation between links, etc.)

Web Services

A very popular type of web service is the Recommendations
It is a web service that through the ingestion of data from the user's history or the availability of specific information on a topic, allows us to make some type of recommendation.

These types of services make it easy to build applications that can deliver a wide range of personalization experiences, including product-specific recommendations and personalized product reclassification.

Applications

- Voice assistants.
- Personalized marketing.
- Fraud detection.
- Self-driving cars.
- Transport optimization.
- Health care.
- Chatbots.

Documentation

There are different types of documentation that are important for the communication, operation, and progress of the project:

- **Product Documentation:** Objectives, scope of the project, description of roles and responsibilities.
- **Requirements Documentation:** Describes the functionality required by the solution. It includes use cases and business rules.
- **Architecture Documentation:** Global diagram of the solution. Describe and list the important components. List of base technologies and infrastructure. It includes assumptions, business rules, costs, future growth.

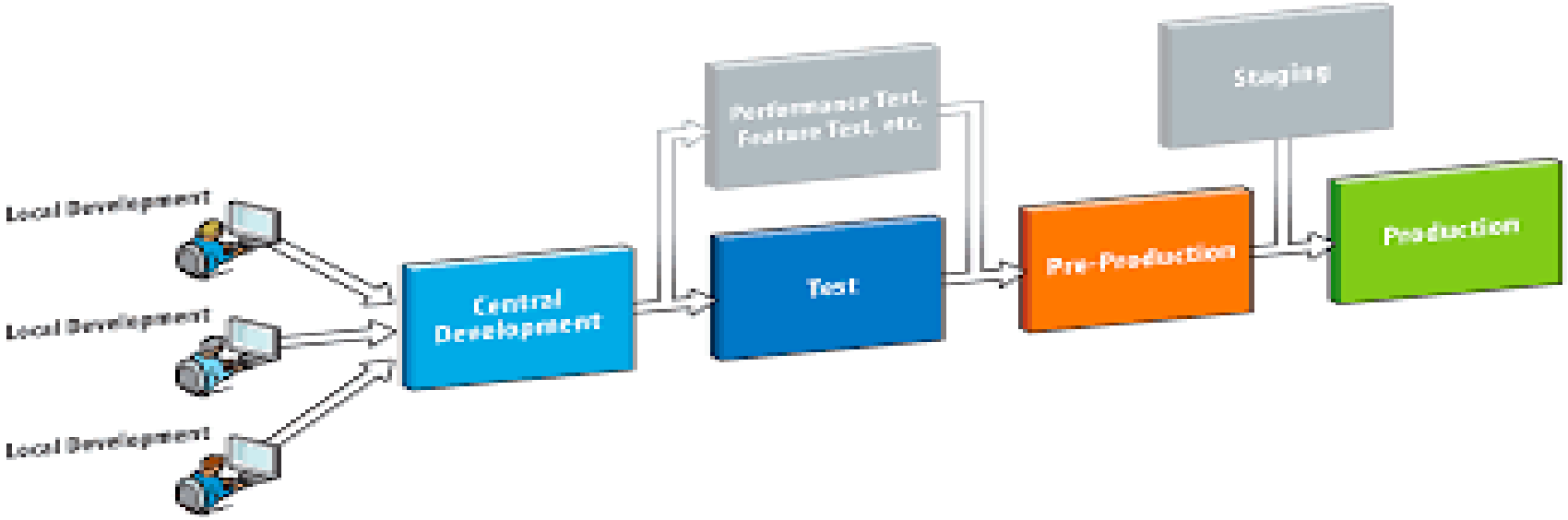
Infrastructure

Infrastructure serves as the foundation that supports the development, deployment, and management of machine learning models. It encompasses the computing resources, networking, storage, and orchestration tools required to ensure smooth operation of the MLOps pipeline.

1.9 Work environments

Work environments

- Development: For experimentation with different features, algorithms, etc.
- Testing/QA: For unit and integration tests, validate the data and its quality, and validate the model and its quality.
- Staging: (Optional) To test new versions of a data product gradually; e.g., sending only 10% of traffic to the Staging environment with the new version of a model.
- Production: It contains all active models, receives real customer data, and sends responses to requests. Continuous, fault-free, and low-latency operation is expected.



1.10 What about other “Ops” practices?

MLOps vs other Xops.

- **DevOps** is a more general methodology that integrates development and operations teams to enable continuous integration (CI) and continuous delivery (CD), reducing time-to-market and improving software quality.
- **MLOps** extends DevOps practices to ML workflows, enabling versioning of datasets, reproducible training, experiment tracking, automated model deployment, and continuous monitoring.
- **DataOps** is another methodology that focuses on enhancing the speed, quality, and collaboration of data analytics and data engineering workflows. Similar to DevOps but for data pipelines.
- **AIOps** utilizes AI/ML to analyze large volumes of log, metric, and event data, detecting anomalies, predicting incidents, and automating remediation.
- **LLMOps** is a specialized branch of MLOps focused on LLM fine-tuning, prompt management, retrieval-augmented generation (RAG), vector databases, and continuous evaluation of generative AI.
- **EmbedOps** is focused on managing embedding generation pipelines, vector store refreshes, drift monitoring, and performance evaluation for semantic search and retrieval-based systems.

LLMOps.

If we can provide a high-level perspective of what it involves, considering the same three phases, we can group them like this:

- Design
    - Data Sourcing.
    - Base Model Selection.
- Development.
    - Prompt engineering.
    - Chains and Agents
    - RAG vs Fine-tuning
    - Testing
- Operations
    - Deployment
    - Monitoring and Observability
    - Cost Management
    - Governance and Security

2 Course Dynamics

Teamwork basis

- One of the objectives of the course is for you to understand the dynamics of interaction among different roles in MLOPs schema
- For this reason:
  - the teams will be assigned based on personal profiles
  - the teams will be mostly conformed by 5 students
  - collaboration and experience sharing is expected among team members
- Teamwork activities are intended to consider role's interactions (but not limited to just the roles involved by a given phase)
- There will be an integration activity via discord next Thursday 18th of September 7:30pm with your assigned Teaching Assistants. Please save the date.
- Teamwork deliverables 2 and 3 (final) will have co-evaluations on TeamMates platform.

Canvas walkthrough

- Course requirements
- Evaluation plan
- Class rules
- Resources

- Discord platform will be the first point of contact for any technical issue
- For any other unresolved issues please contact the Teaching Team


Questions?

Week 1 Activity

- Software installation and environment setup
- Please review technical documentation of the tools
- Installation guides are in CANVAS