From Jupyter to Production: A Pragmatic Guide to Deploying Machine Learning Models

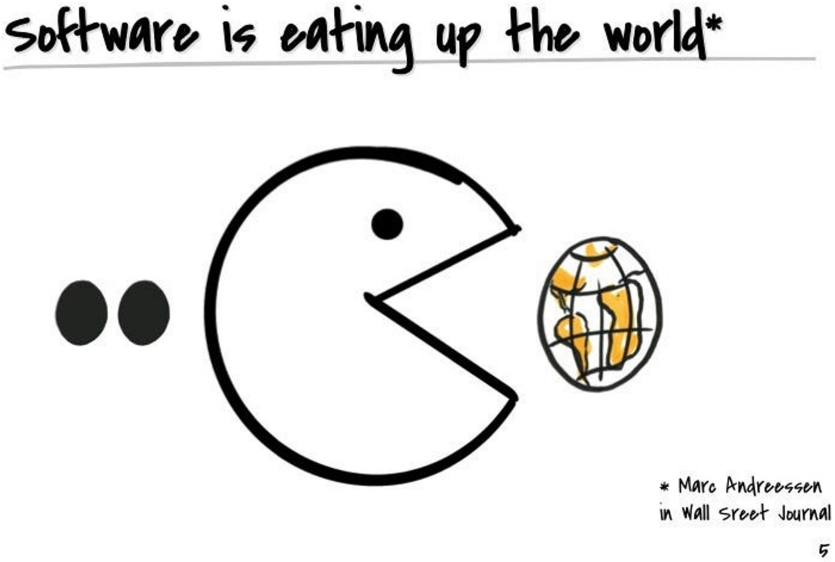
Yunuscan Kocak

About Me:

```
{
    "name": "Yunuscan Kocak",
    "role": "Machine Learning Scientist",
    "company": "Booking.com",
    "previously": [ "Twitter", "Beat", "KPN", "Teknopar" ],
    "education": [ "MSc @ TOBB UoET", "BSc @ TOBB UoET" ]
}
```

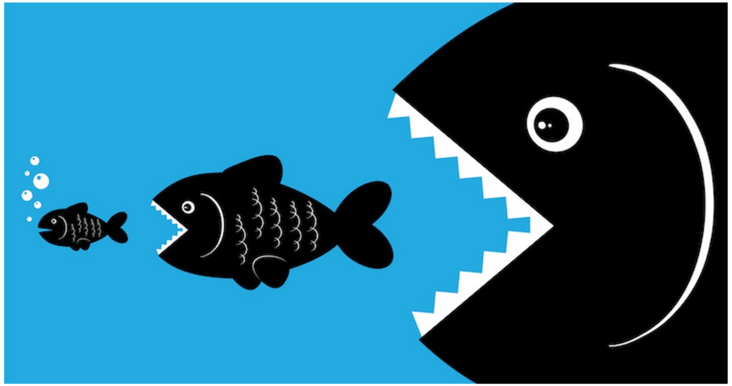
Machine Learning is Eating the World

- 2011: Software is eating the world:
  - Transformative and disruptive force across various industries, fundamentally changing the way businesses operate and deliver value.
  - The leading book retailers, video services providers, music companies, entertainment companies, and even movie production companies were essentially software companies.



Machine Learning is Eating the World

- 2023: Machine Learning is eating the world:
  - Similar transformation is happening across industries, machine learning models are being added to every companies products
  - Smart Assistants (e.g., Siri, Google Assistant, Alexa)
  - Recommendation Systems (e.g., Netflix, Amazon, Spotify)
  - Speech Recognition (e.g., Speech-to-Text applications)
  - Autocorrect (e.g., on smartphones)
  - Image and Video Recognition (e.g., Google Photos, Instagram, Snapchat filters)
  - Language Translation Services (e.g., Google Translate)



Why does deploying ML models matter?

- Essentially, deployment enables usage of machine learning model and providing value to the end users.
- Deploying models transforms data science from an experimental phase to a practical, impactful solution.
- It's the bridge between insights gained in Jupyter notebooks and tangible, real-world applications.

An example use case:

- Automated stock market trading system
  - A machine learning model for stock market prediction
- Input:
  - Historical price information of stocks
  - Financial information about companies
- Output:
  - Tomorrow's price

Every good story starts with a Jupyter notebook

- Obtain some historical price and financial information about companies:
  - stock_price.csv
  - financial_info.csv
- Extract features from these datasets
  - Trends
  - Moving averages
  - Transformations
- Experiment with a few ML algorithms
- Evaluate the performance of the models
  - Metrics: RMSE
- Pick the best one
  - Does it end here?

Challenges:

- In order to run this model and provide value we need:
    - Continuous data collection about financial data and stock market prices
    - Re-create our features and train many new models
    - Evaluate these models and decide which one to trust
    - Run your model on new data everyday to provide next day's price predictions
    - Integrate this into another service that creates buy/sell decisions using our price predictions
    - Monitor their performance in real world conditions

More Challenges:

- In addition to this:
  - Continuously integrate changes and deliver new versions
  - A robust and operational infrastructure is needed to run the software
    - Servers
    - GPUs
  - It needs to be up to date with dependencies we require
    - Python libraries
    - Other software

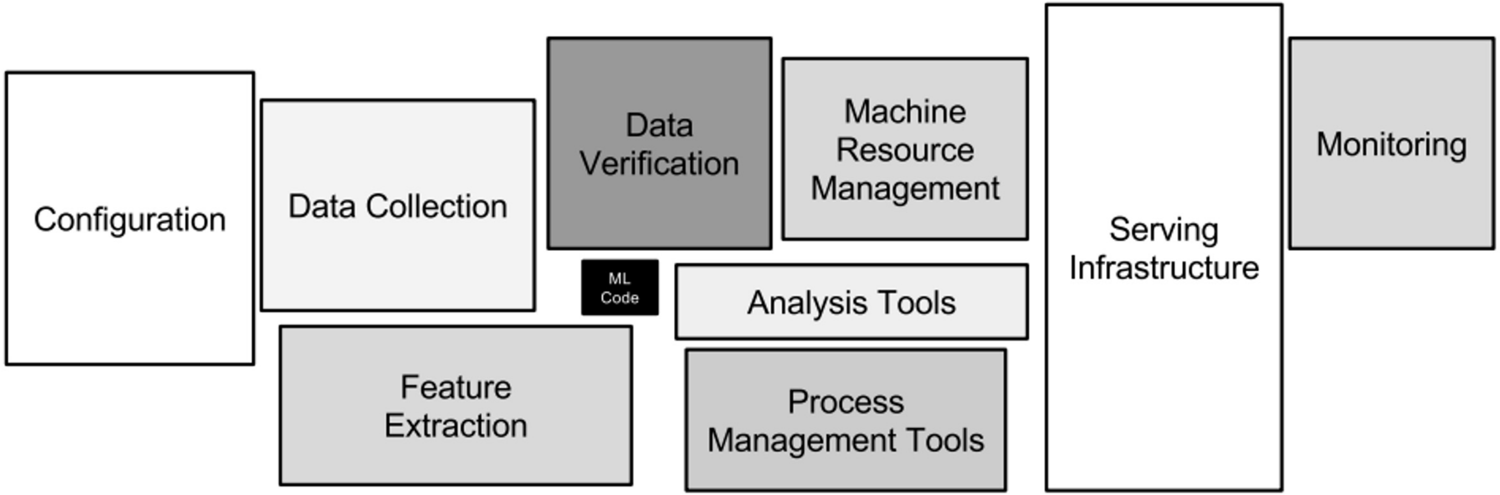


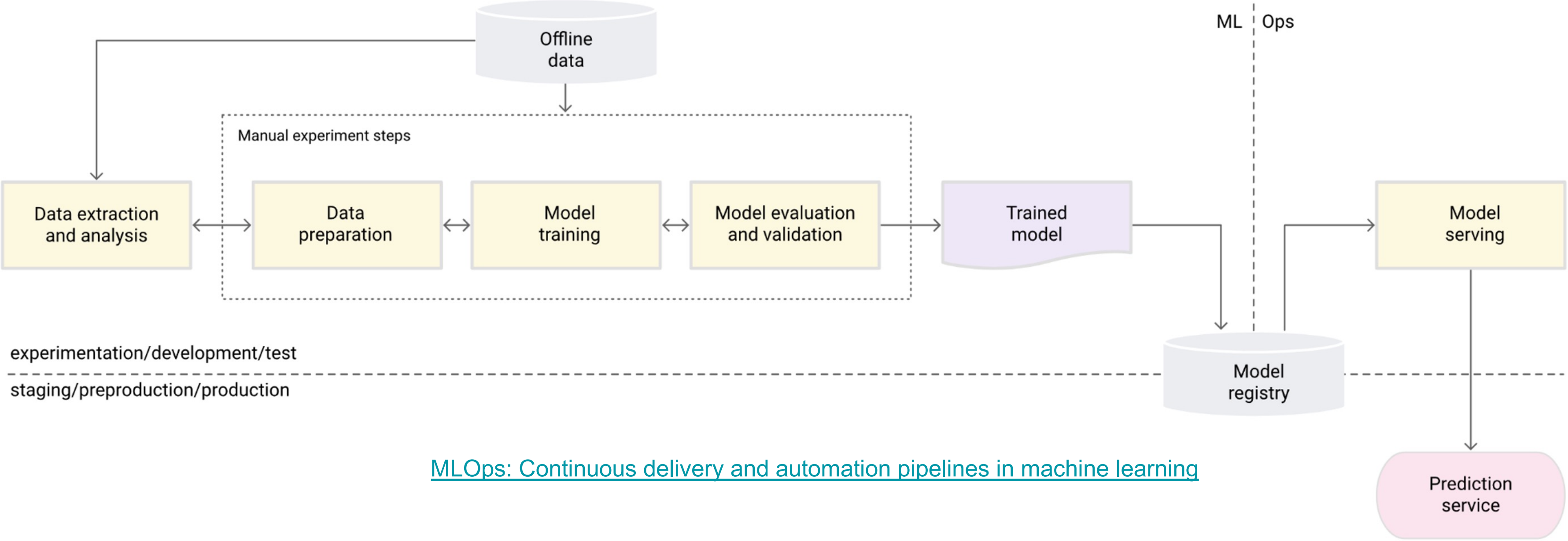
Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Sculley, David, et al. "Hidden technical debt in machine learning systems." Advances in neural information processing systems 28 (2015).

MLOps comes to the rescue!

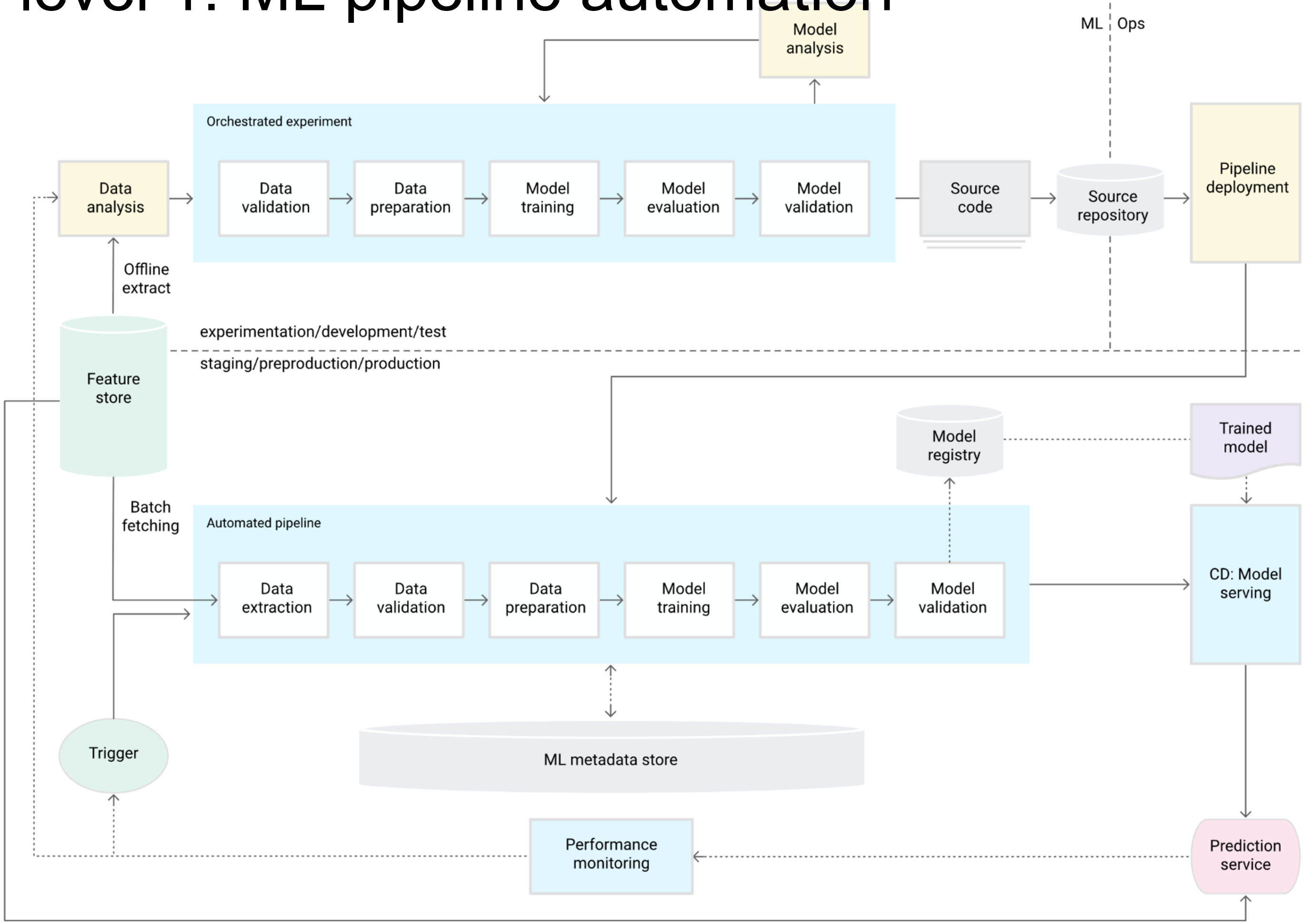
- As you can imagine, this complexity needs to be managed and automated as much as possible
- **MLOps** is a set of practices that aims to streamline and optimize the end-to-end machine learning lifecycle, from model development to deployment and monitoring.
- MLOps enhances agility, allowing teams to iterate quickly and deploy models faster, aligning with the dynamic nature of machine learning.
- Depending on the companies machine learning maturity, there are a few MLOps levels defined by Google

MLOps level 0: Manual process

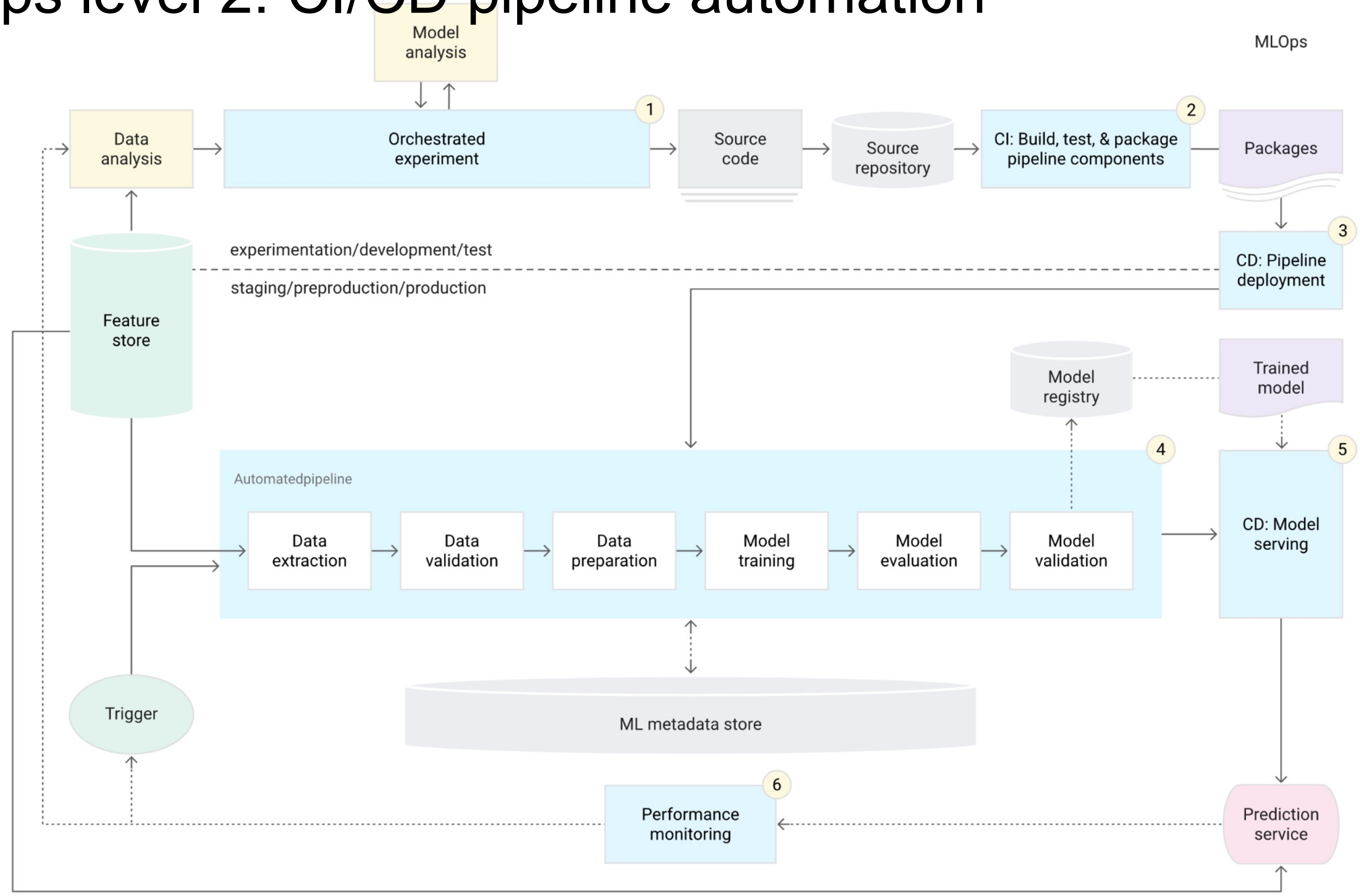


MLOps: Continuous delivery and automation pipelines in machine learning

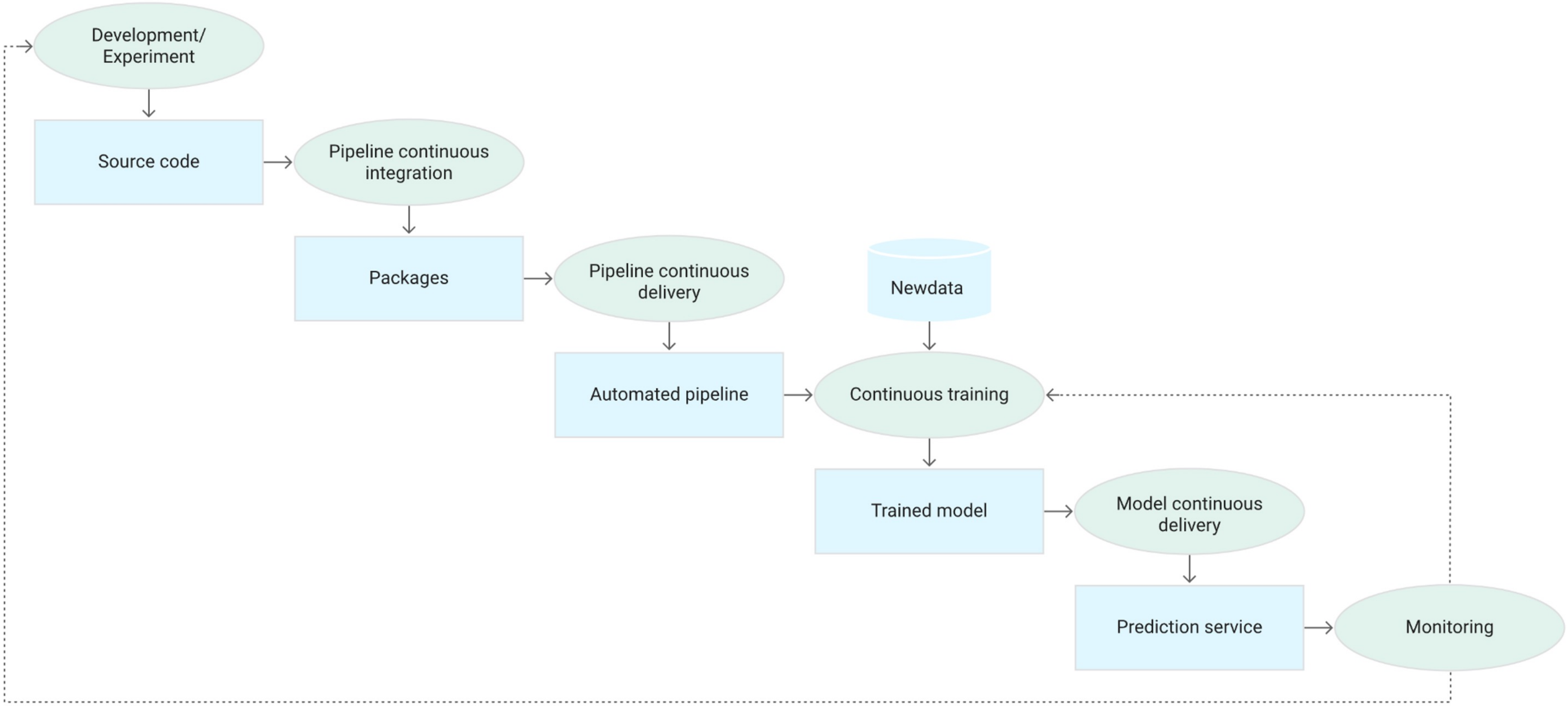
MLOps level 1: ML pipeline automation



MLOps level 2: CI/CD pipeline automation

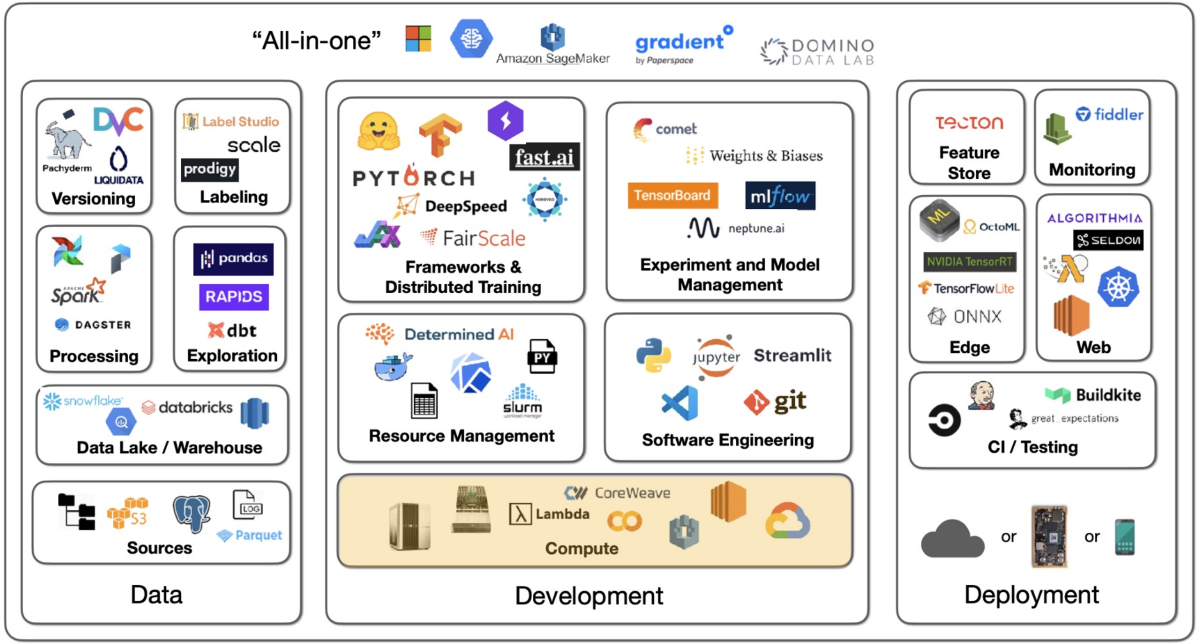


MLOps level 2: CI/CD pipeline automation



Tools & Technologies:

- Implementing these processes independently demands a significant amount of time; fortunately, we can leverage the expertise of industry leaders.



“All-in-one”

Versioning

Labeling

Processing

Exploration

Data Lake / Warehouse

Sources

Data

Frameworks & Distributed Training

Experiment and Model Management

Resource Management

Software Engineering

Compute

Development

Feature Store

Monitoring

Edge

Web

CI / Testing

or

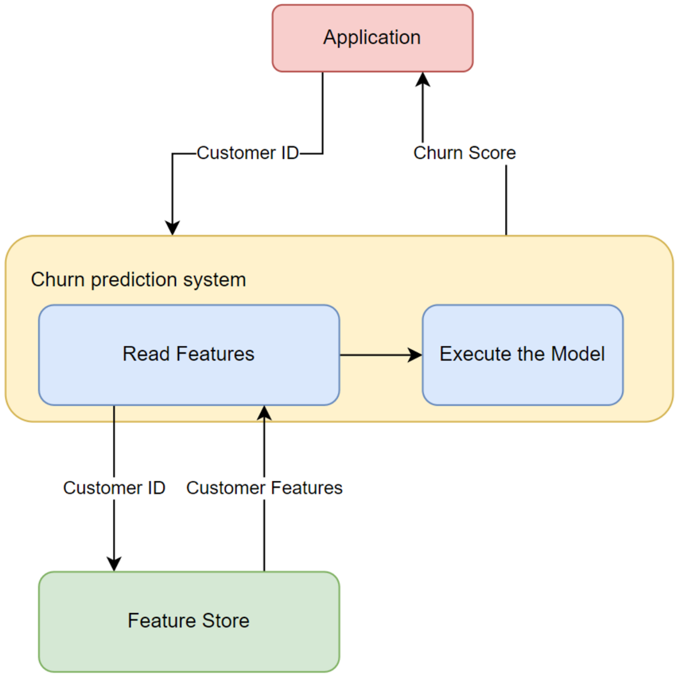
or

Deployment

Choosing Between Real-Time and Batch Prediction Serving

Real-Time Predictions: *Instant Insights, Swift Action*

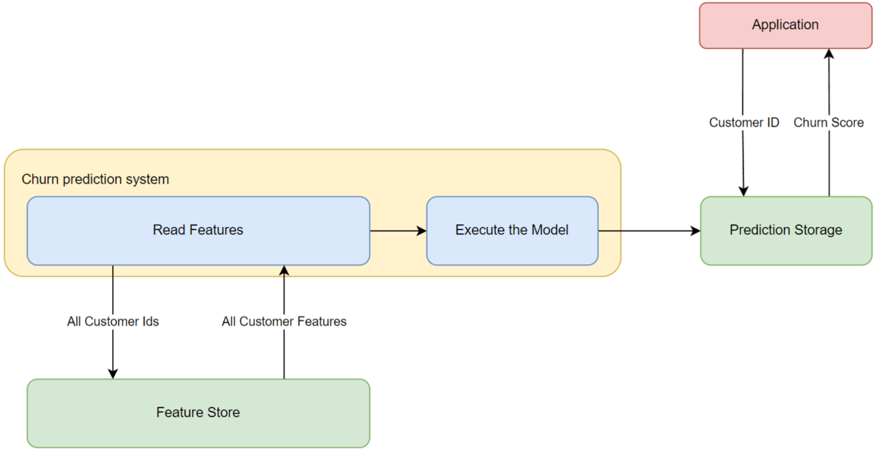
- Overview:
  - Predicting data as it arrives, providing immediate results.
  - Well-suited for applications demanding low-latency responses.
- Pros:
  - Swift decision-making.
  - Enhanced user experience
- Cons:
  - Higher infrastructure and processing costs.
  - Complexity in handling and managing real-time data streams.
- Use Cases:
  - Autonomous Driving, Recommendation Systems, Language Translation



Choosing Between Real-Time and Batch Prediction Serving

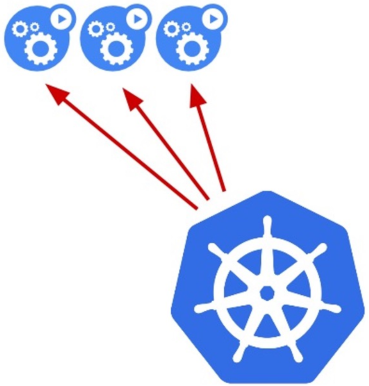
Batch Predictions: *Efficiency and Scalability*

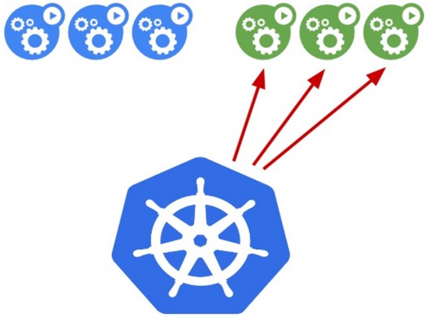
- Overview:
  - Collecting and predicting data in predefined intervals or batches.
  - Suitable for tasks where immediate results are not critical.
- Pros:
  - Efficient for large-scale data processing.
  - Cost-effective for non-time-sensitive analyses.
- Cons:
  - Latency in obtaining results.
  - Features and predictions might get stale
- Use Cases:
  - Daily stock market predictions
  - Quarterly retail inventory predictions



Deployment Strategies

- Transitioning from model development to deployment is a critical phase.
- Deployment strategies provide a structured approach to manage this transition seamlessly.
- Deployment strategies help identify and mitigate potential risks before they impact the operational environment.
- Minimizing the chances of downtime, data inconsistencies, and performance issues.
- Ensures optimal utilization of infrastructure, minimizing operational costs.

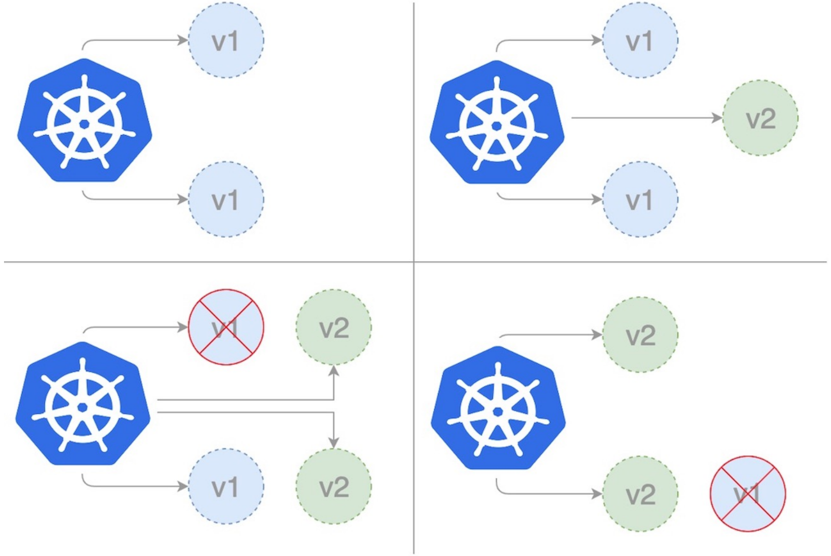




https://auth0.com/blog/deployment-strategies-in-kubernetes/

Deployment Strategies

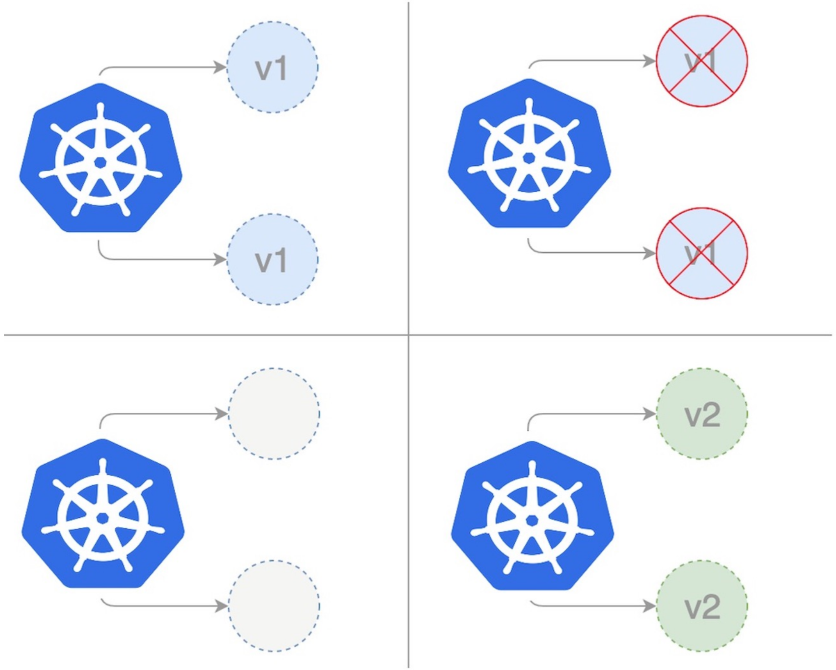
- Rolling Strategy:
  - Slowly replace instances of the previous version of an application, and test a small portion of the traffic before changing all instances



https://auth0.com/blog/deployment-strategies-in-kubernetes/

Deployment Strategies

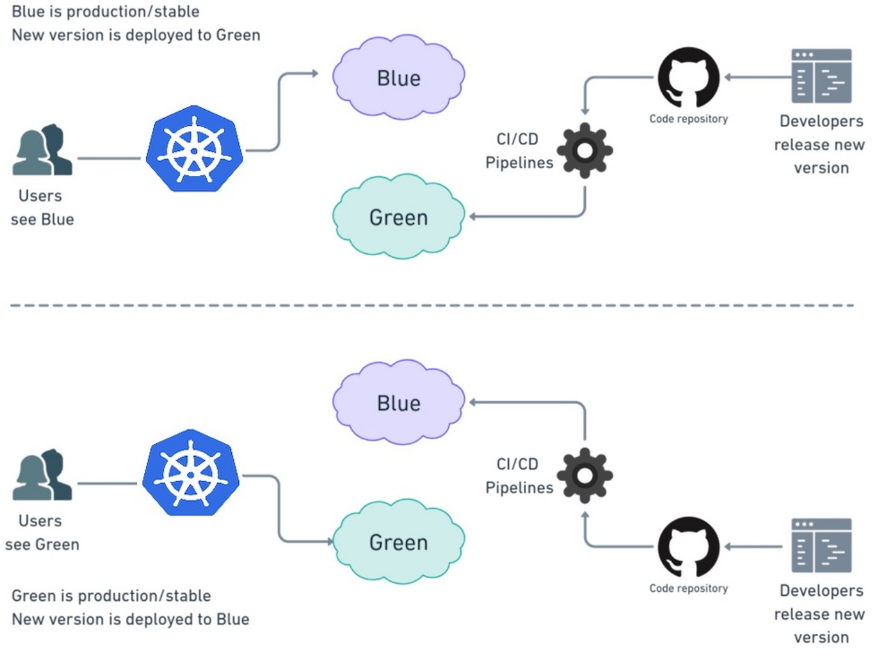
- Recreate Strategy:
  - Entirely replace existing environment, this might be necessary if you can’t maintain two versions of the machine learning model at the same time



https://auth0.com/blog/deployment-strategies-in-kubernetes/

Deployment Strategies

- Blue-Green Deployment using Routes:
  - Maintains two identical environments: blue (active) and green (inactive). Smoothly transitions traffic between environments using route switches.


Blue is production/stable
New version is deployed to Green
Blue
Code repository
Developers release new version
CI/CD Pipelines
Users see Blue
Green
Blue
CI/CD Pipelines
Users see Green
Green
Code repository
Developers release new version
Green is production/stable
New version is deployed to Blue

https://auth0.com/blog/deployment-strategies-in-kubernetes/

Summary

- As the usage of machine learning applications expands, there is a growing demand to ensure their readiness for production.
- Maintaining a machine learning model in production requires managing many components and infrastructure.
- MLOps is a set of practices that aims to streamline and optimize the end-to-end machine learning lifecycle.
- There are many tools for helping us in ML lifecycle. Feature stores, workflow orchestrators, model training and evaluation libraries, hyperparameter optimization tools and ecosystem is constantly evolving.
- Models can be deployed in Real-time or Batch and they have their pros and cons.
- Deployment strategies provide a structured approach to manage this transition seamlessly.