ML REPA

Инструменты автоматизации экспериментов и деплоя моделей

Mikhail Rozhkov

What you'll learn

- Путь от экспериментов в продакшн
- Обзор инструментов
- MLflow в экспериментах и production

Обо мне

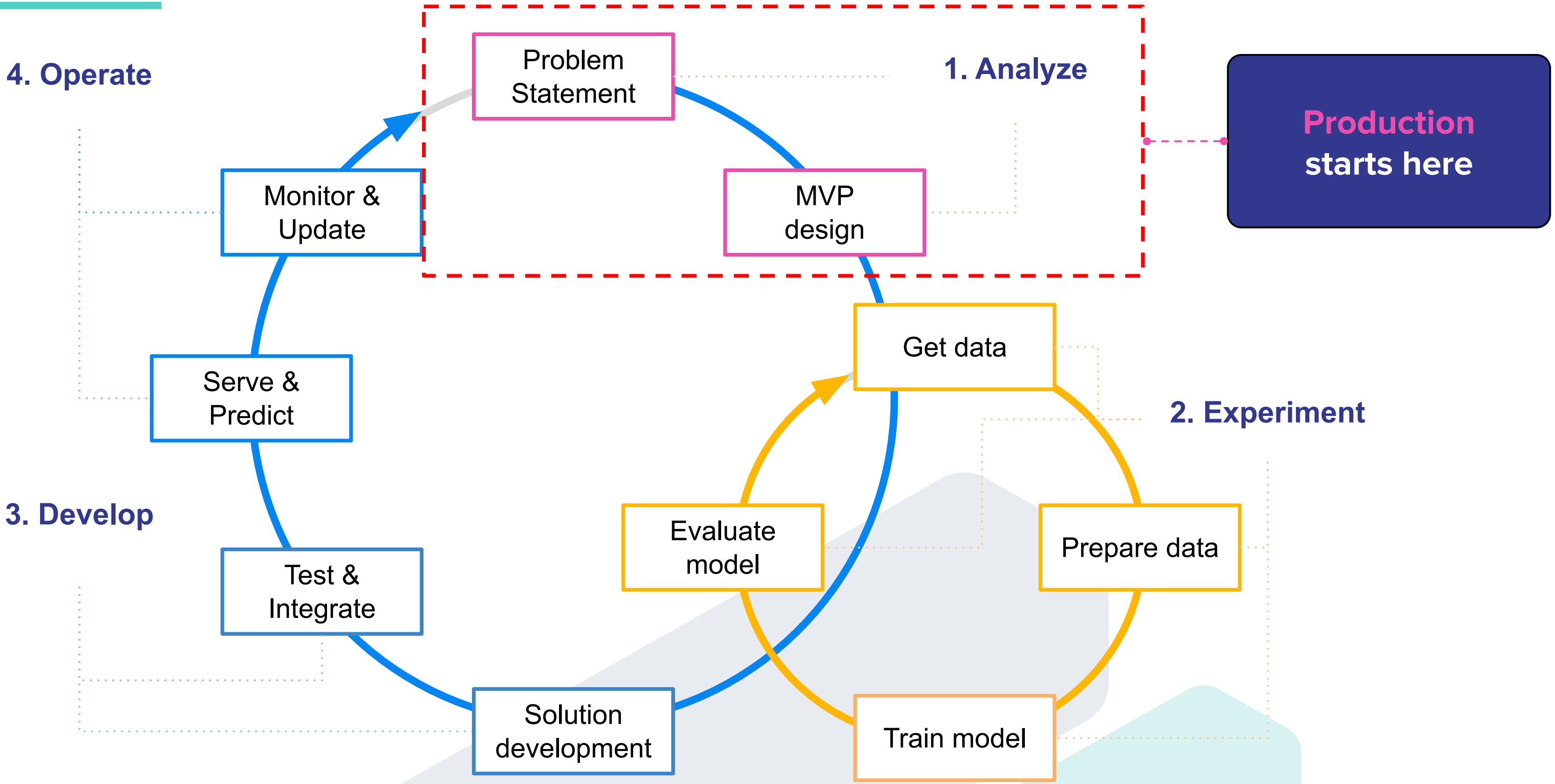
Website: mlrepa.ru
Telegram/ODS Slack: #mlrepa

Михаи Рожков

- **Creator ML REPA**: Machine Learning REPA: Reproducibility, Experiments and Pipelines Automation
- **Автор курсов:**
  - Machine Learning experiments reproducibility and engineering with DVC
  - ML REPA course: Reproducibility, engineering and MLOps from A to Z
- **5+ лет в Machine Learning & Data Science** (7labs.ru, Raiffeisenbank, Банк Открытие)

Путь от экспериментов в продакшн

Какое ТЗ, такой и Production!



Inspired by Uber's workflow of a machine learning project diagram. Scaling Machine Learning at Uber with Michelangelo https://eng.uber.com/scaling-michelangelo/

У каждого свой “production”

Production ML

- Бизнес-кейс
- Данные
- Процессы (Training / Deployment / Monitoring)
- Инфраструктура

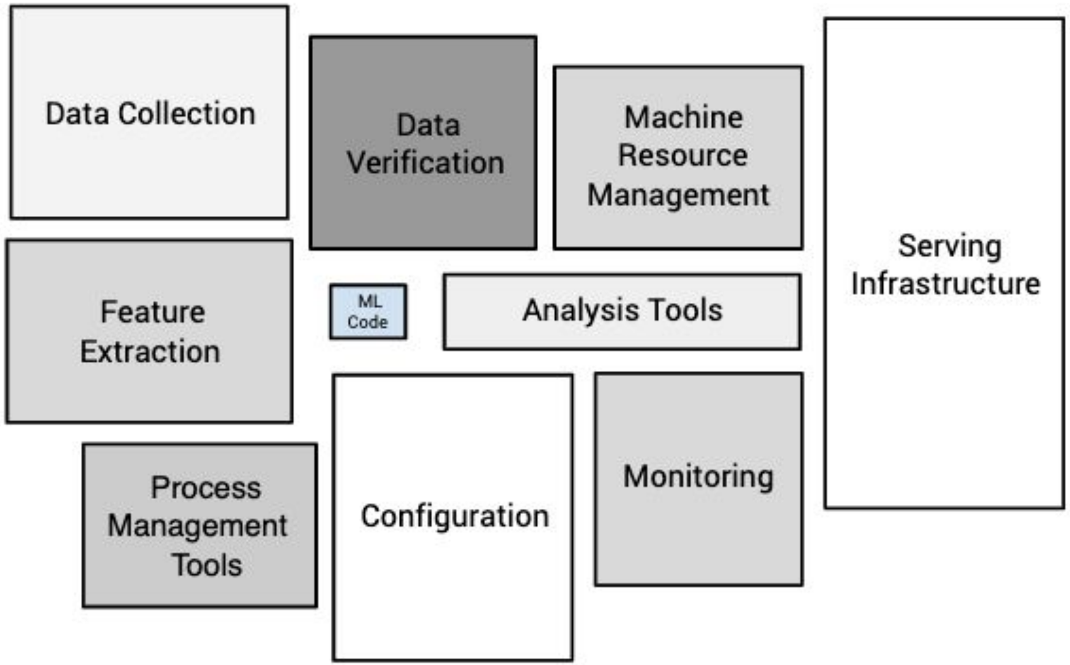
Кейс 1: Склонность клиентов к оттоку

Бизнес-кейс	• **Склонность клиентов к оттоку**
Данные	• Профиль клиента, транзакции • Много источников, большая глубина истории в данных
Процессы (Training / Deployment / Monitoring)	• Batch Training • Batch Scoring • Batch Monitoring
Инфраструктура	• Распределенные вычисления, большие объемы данных • Hadoop, Spark, Airflow

Кейс 2: Рекомендация в E-Commerce

Бизнес-кейс	• **Рекомендация в E-Commerce**
Данные	• Профиль клиента • Просмотры / поведение на сайте
Процессы (Training / Deployment / Monitoring)	• Batch Training • Online Scoring • Online/Batch Monitoring
Инфраструктура	• Масштабируемость, высокая скорость ответа • Kubernetes

Технические задачи “production” во всех случаях очень похожи

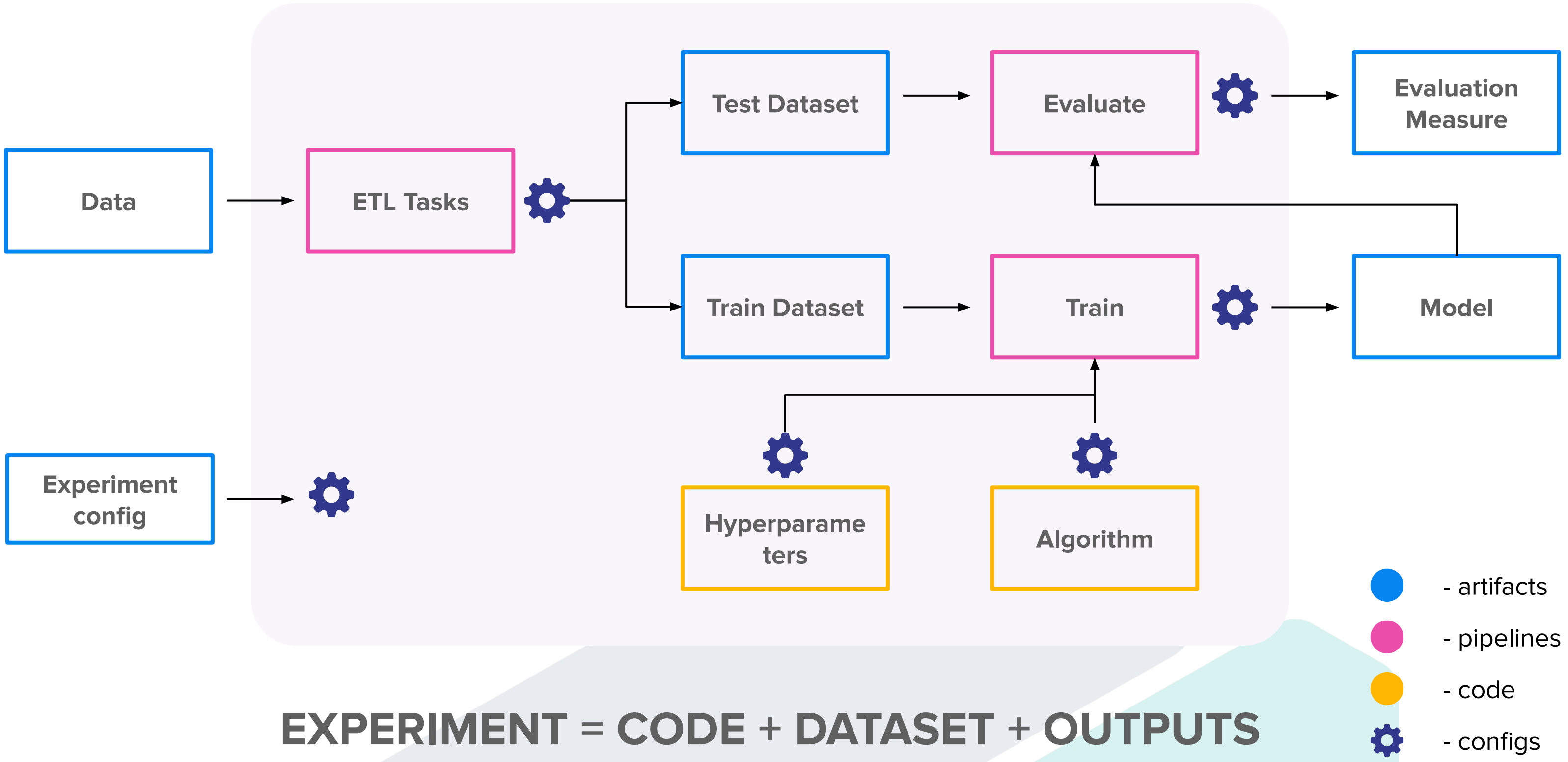


Source: https://developers.google.com/machine-learning/crash-course/production-ml-systems

Управление экспериментами и воспроизводимость

ML Experiments

take long time and produce mess of metrics and artifacts

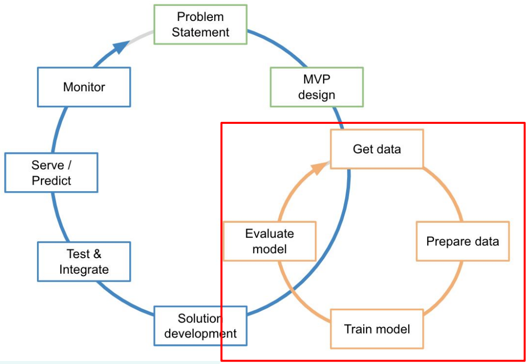


EXPERIMENT = CODE + DATASET + OUTPUTS

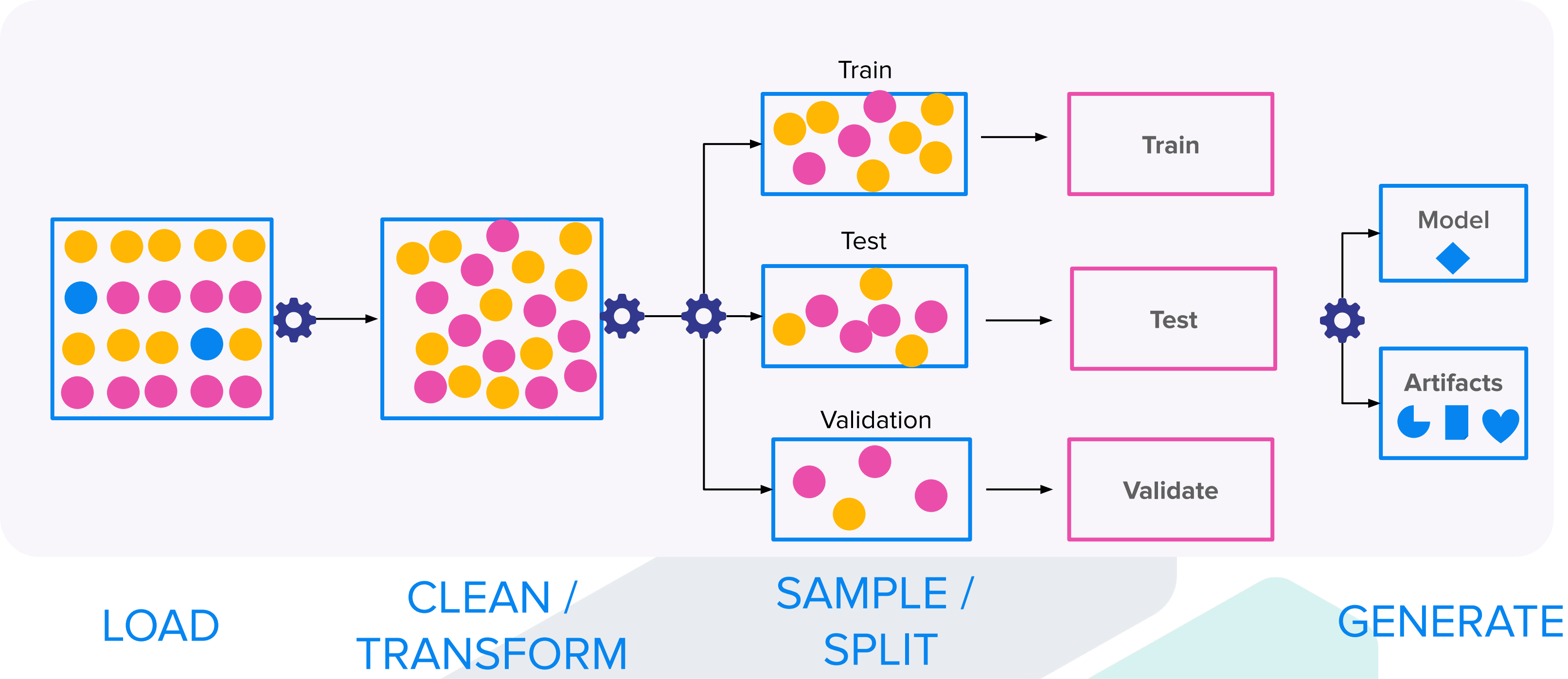
Experiments management

Tasks

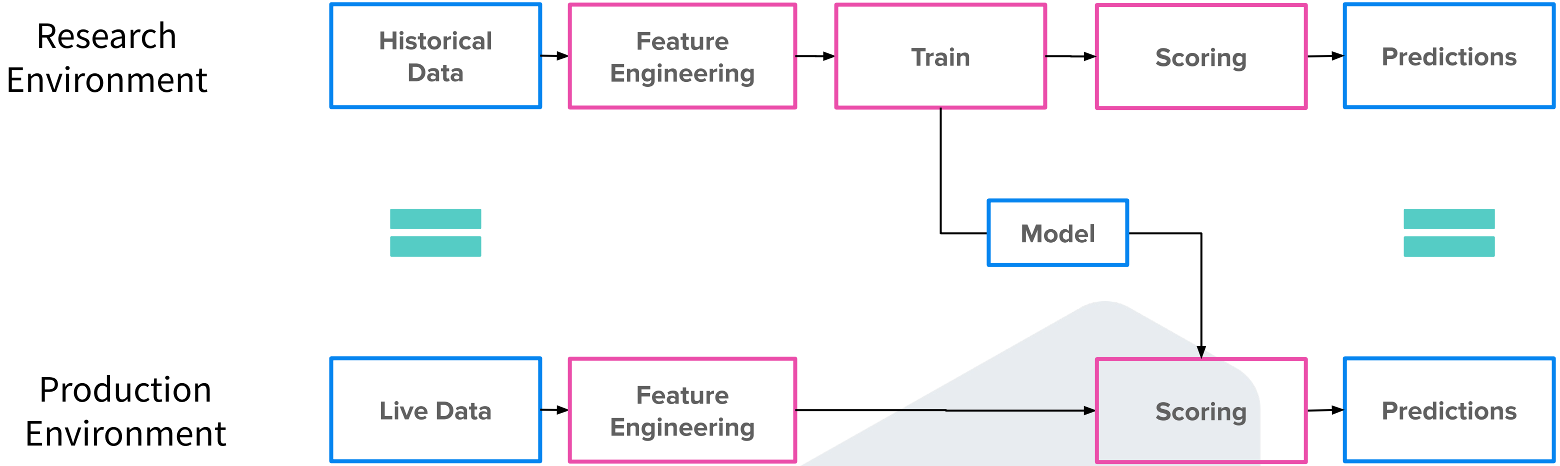
- prepare data
- prepare experiment config
- run experiments
- browse history
- compare results
- share results
- documentation



Data dependency is complicated



Priority 1: Reproducibility

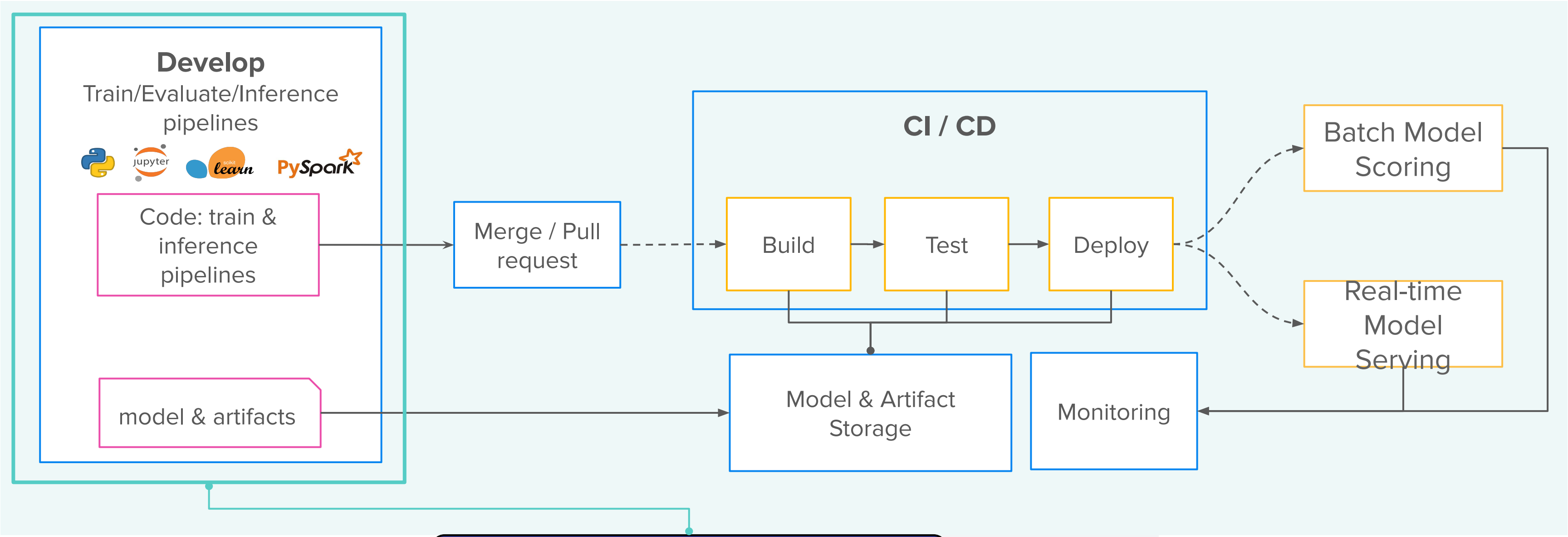


ML Reproducibility checklist for DS

1. Environment dependencies control ✓
2. Code version control ✓
3. Control run params ✓
4. Automated pipelines ✓
5. Control execution DAG ✓
6. Artifacts version control ✓
7. Experiments results tracking ✓
8. Automated CI
9. Automated CD

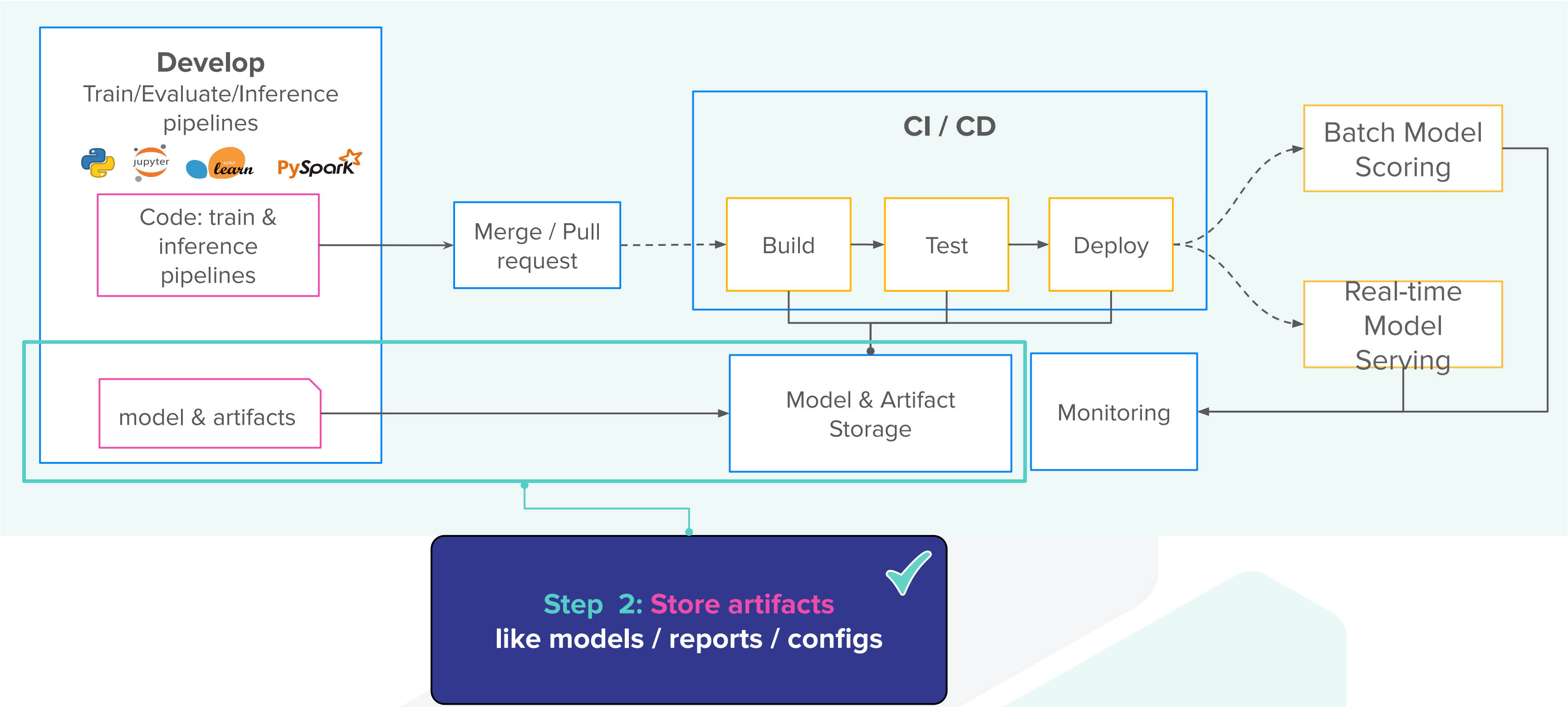
Процесс MLOps

Start with code & artifacts management

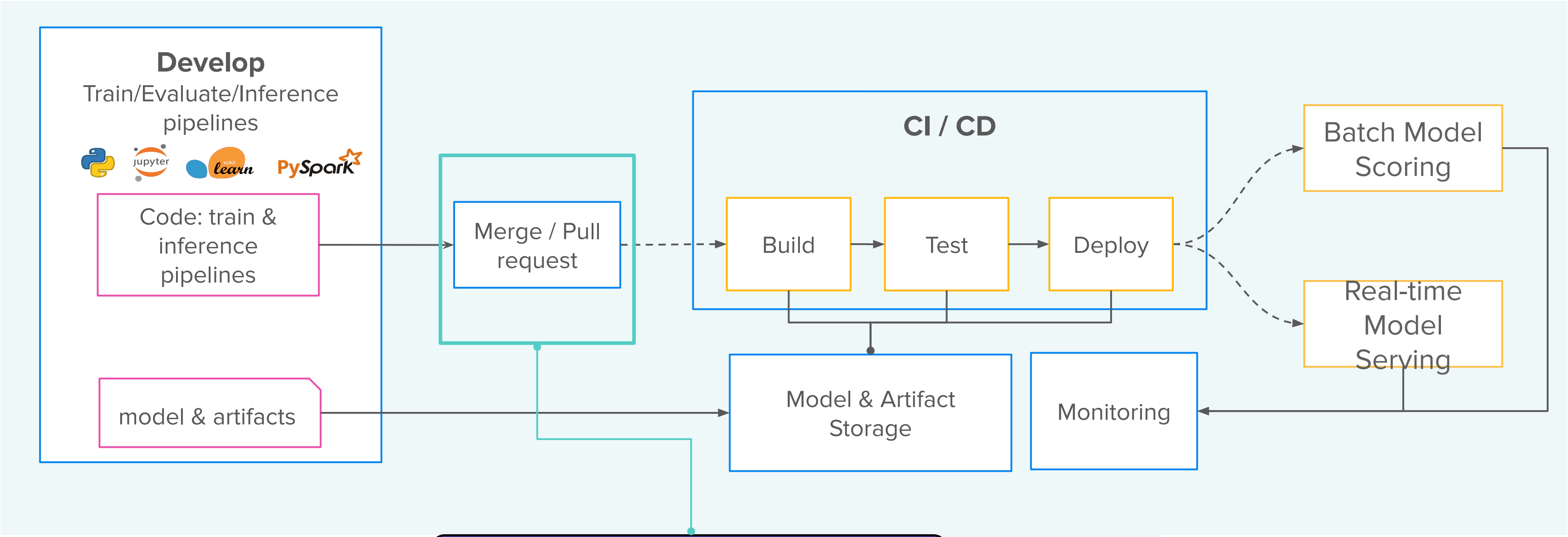


Step 1: Develop code for train/evaluate/inference pipelines

Start with code & artifacts management

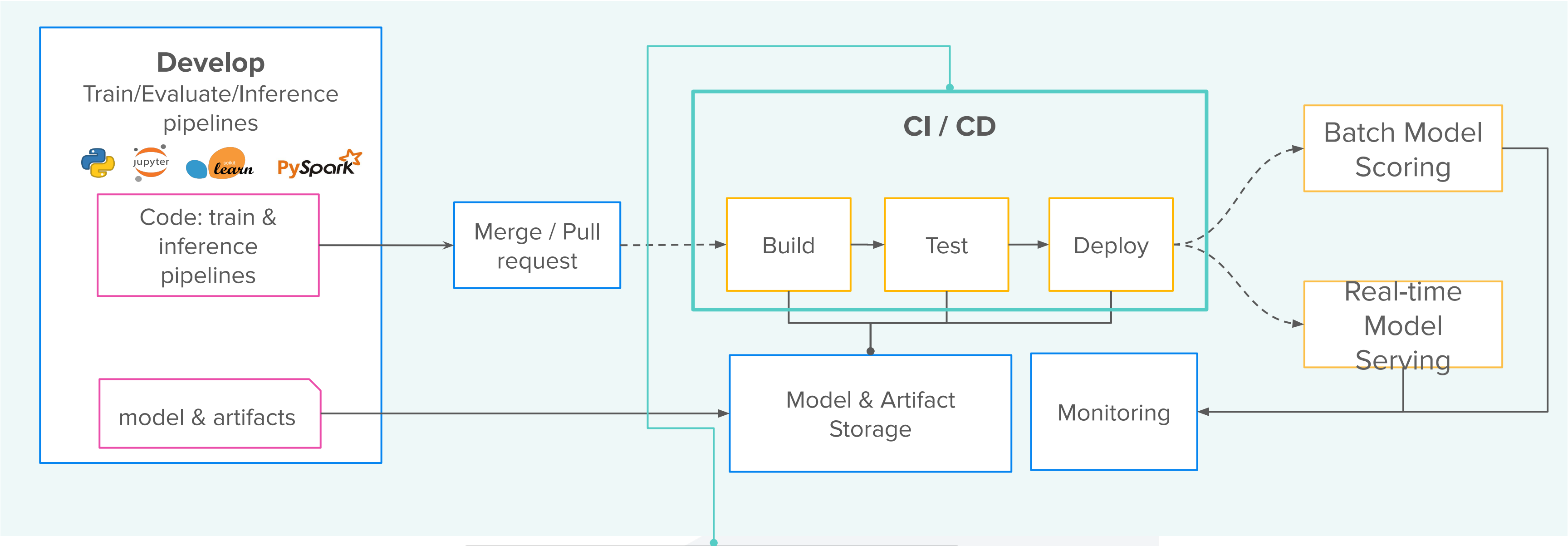


Automate CI/CD



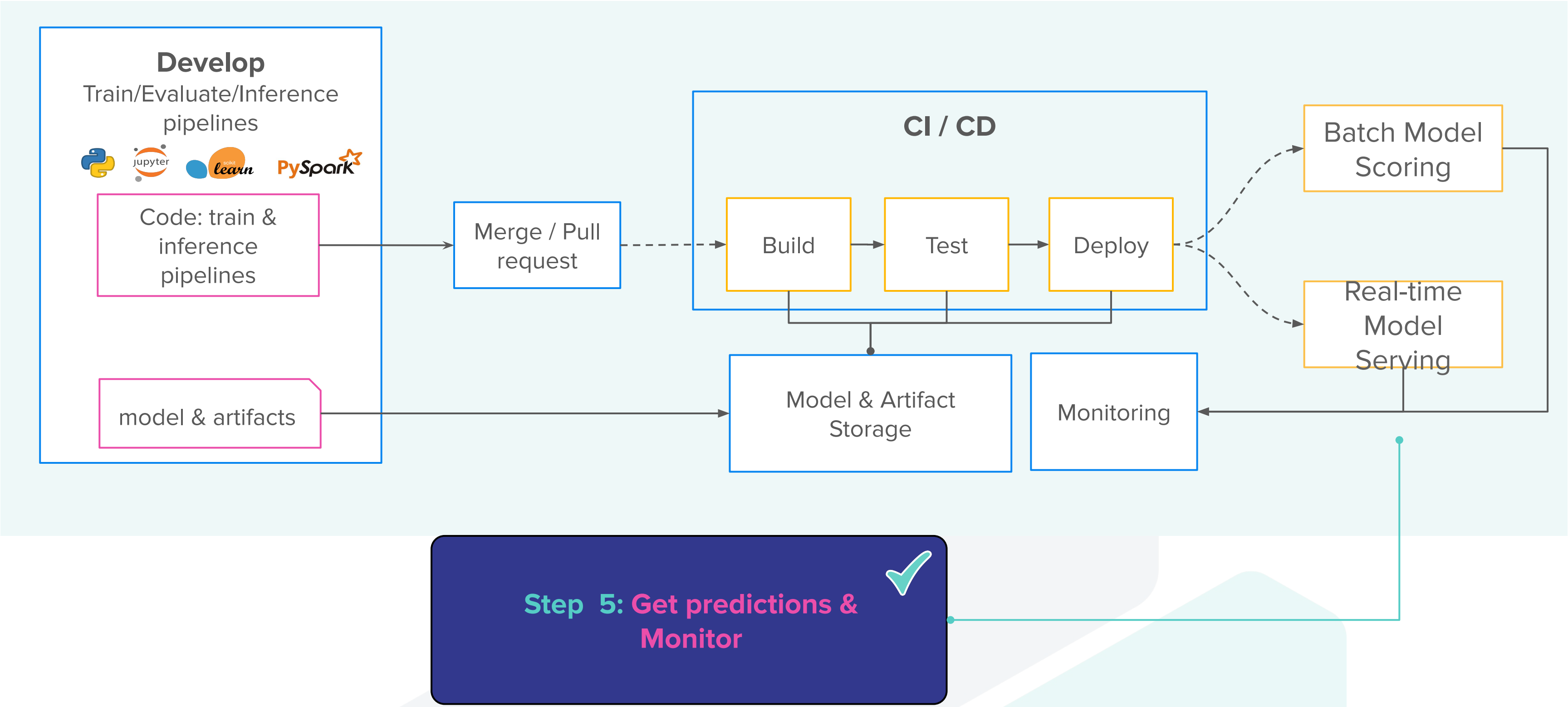
Step 3: Merge / Pull request by the engineer (or data scientist) starts CI / CD process

Automate CI/CD



Step 4: Build / Test / Deploy
ML solution: service or pipeline

Monitor your models and pipelines

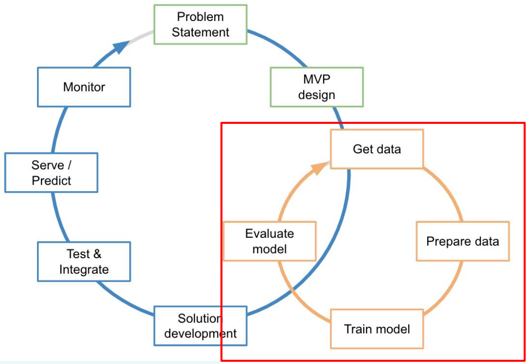


Инструменты управления экспериментами

Experiments management

Tasks

- prepare data
- prepare experiment config
- run experiments
- browse history
- compare results
- share results
- documentation



params.yaml

```yaml
...

train:
 cv: 3
 estimator_name: logreg
 estimators:

   logreg: # sklearn.linear_model.LogisticRegression
     param_grid: # params of GridSearchCV constructor
       C: [0.001]
       max_iter: [100]
       solver: ['lbfgs']
       multi_class: ['multinomial']

evaluate:
 metrics_file: metrics.json
 confusion_matrix_png: confusion_matrix.png
 classes_path: classess.csv
```

DVC links data and code

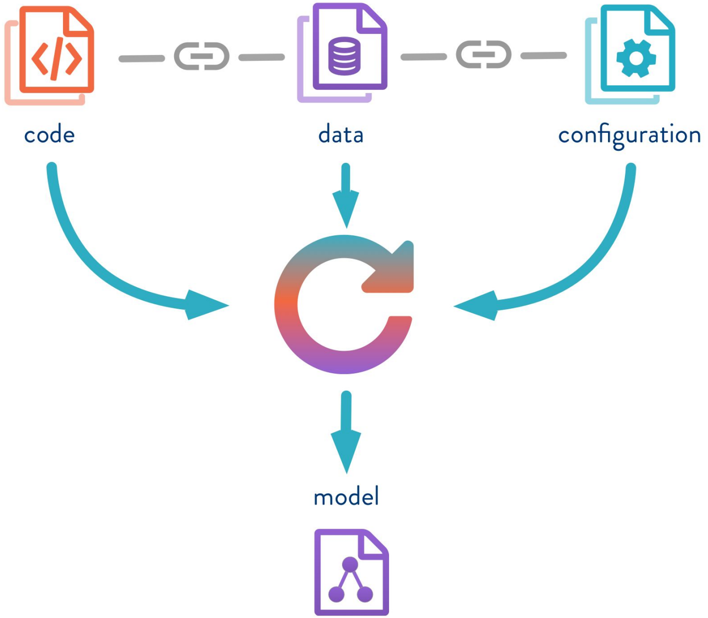
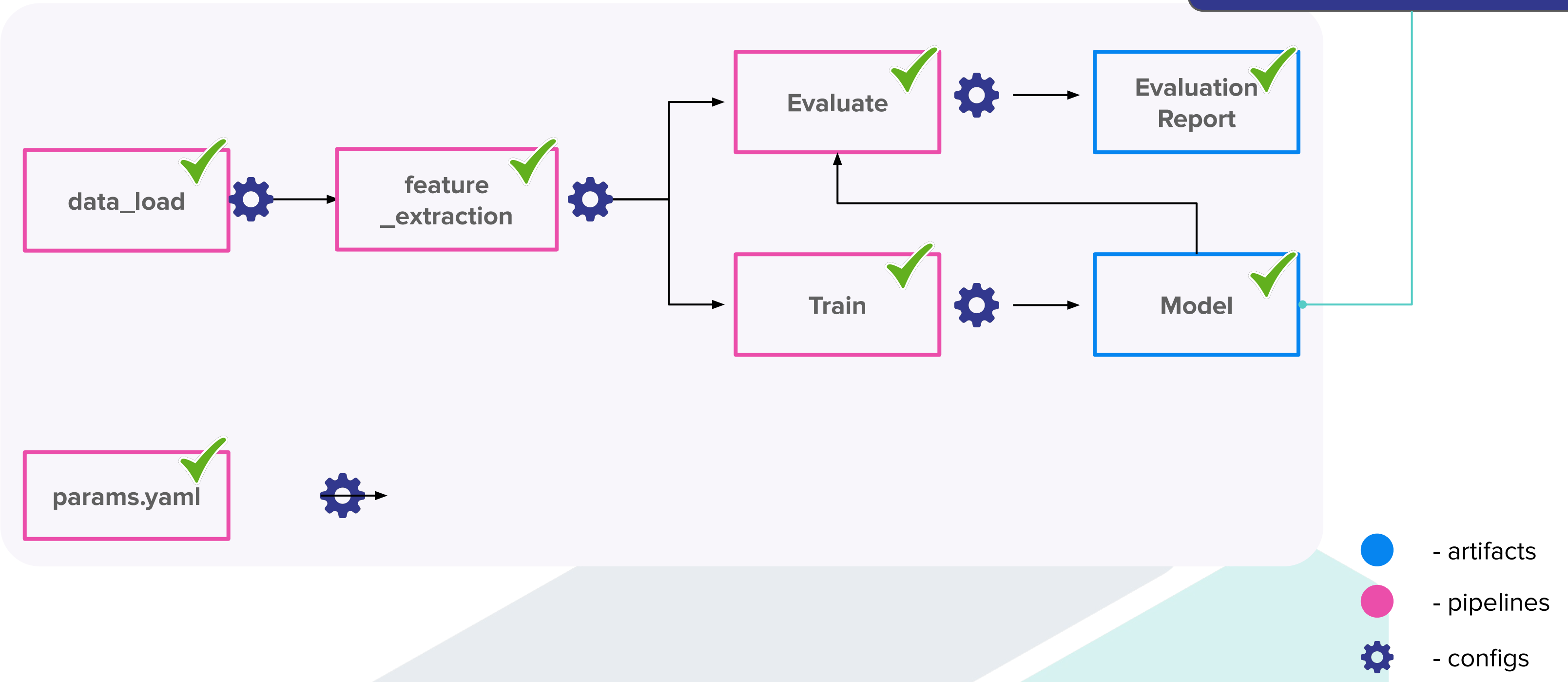


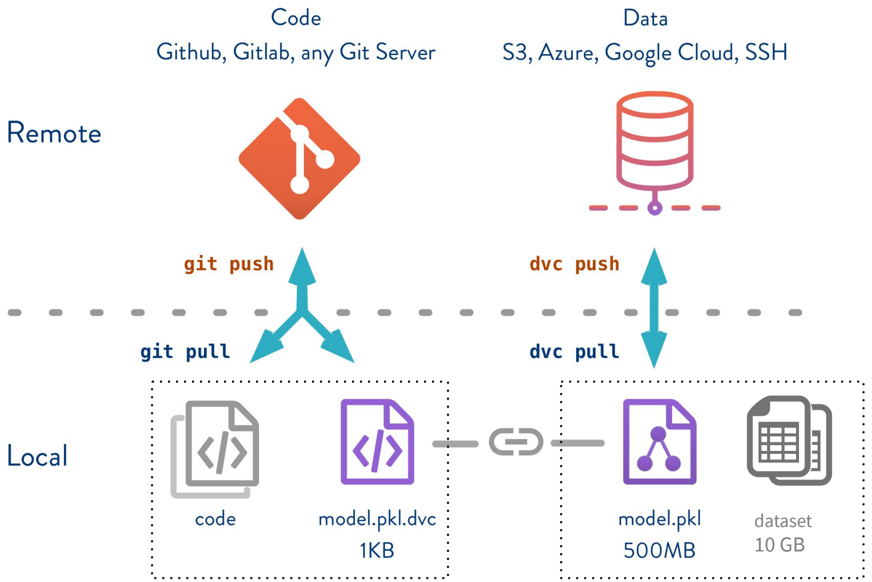
Image source: https://dvc.org/doc/tutorial

Automate experiments with DVC

Run with 1 command:
dvc repro



How DVC works with data?

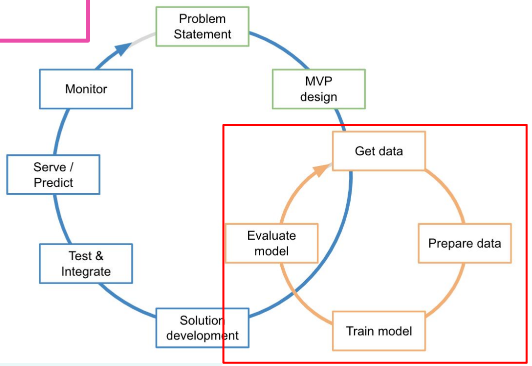


Original image source: https://dvc.org/doc/use-cases/data-and-model-files-versioning

Experiments management

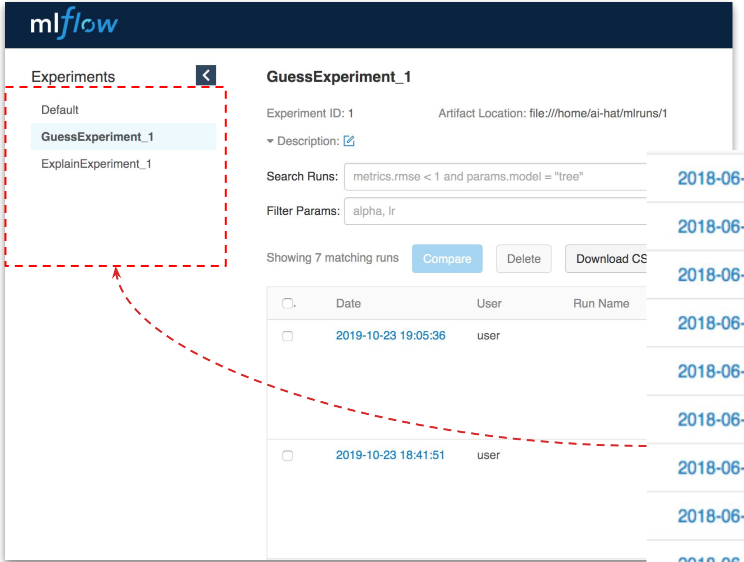
Tasks

- prepare data
- prepare experiment config
- run experiments
- browse history
- compare results
- share results
- documentation



Manage metrics & artifacts with MLflow

Metrics tracking



	alpha	L1	mae	r2	rmse
2018-06-04 23:00:10	1	1	0.649	0.04	0.862
2018-06-04 23:00:10	1	0.5	0.648	0.046	0.859
2018-06-04 23:00:10	1	0.2	0.628	0.125	0.823
2018-06-04 23:00:09	1	0	0.619	0.176	0.799
2018-06-04 23:00:09	0.5	1	0.648	0.046	0.859
2018-06-04 23:00:09	0.5	0.5	0.628	0.127	0.822
2018-06-04 23:00:09	0.5	0.2	0.621	0.171	0.801
2018-06-04 23:00:09	0.5	0	0.615	0.199	0.787
2018-06-04 23:00:09	0	1	0.578	0.288	0.742
2018-06-04 23:00:09	0	0.5	0.578	0.288	0.742
2018-06-04 23:00:09	0	0.2	0.578	0.288	0.742
2018-06-04 23:00:08	0	0	0.578	0.288	0.742

Experiment params

MLflow Components

mlflow

MLflow Tracking

Record and query experiments: code, data, config, and results

Read more

MLflow Projects

Package data science code in a format to reproduce runs on any platform

Read more

MLflow Models

Deploy machine learning models in diverse serving environments

Read more

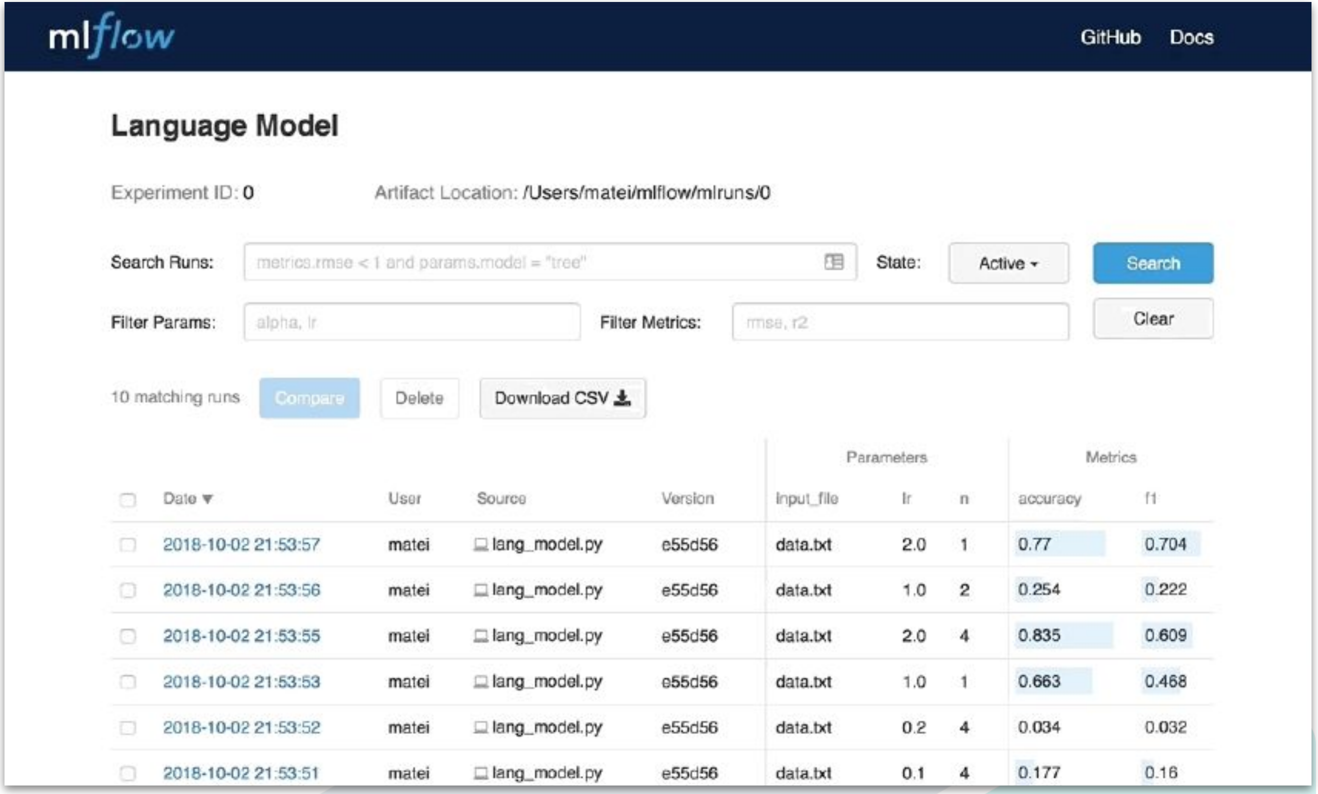
Model Registry

Store, annotate, discover, and manage models in a central repository

Read more

MLflow metrics tracking

Tracking UI


mlflow
GitHub
Docs
Language Model
Experiment ID: 0
Artifact Location: /Users/matei/mlflow/mlruns/0
Search Runs:
metrics.rmse < 1 and params.model = "tree"
State:
Active
Search
Filter Params:
alpha, lr
Filter Metrics:
rmse, r2
Clear
10 matching runs
Compare
Delete
Download CSV
Parameters
Metrics
Date
User
Source
Version
input_file
lr
n
accuracy
f1
2018-10-02 21:53:57 matei lang_model.py e55d56 data.txt 2.0 1 0.77 0.704
2018-10-02 21:53:56 matei lang_model.py e55d56 data.txt 1.0 2 0.254 0.222
2018-10-02 21:53:55 matei lang_model.py e55d56 data.txt 2.0 4 0.835 0.609
2018-10-02 21:53:53 matei lang_model.py e55d56 data.txt 1.0 1 0.663 0.468
2018-10-02 21:53:52 matei lang_model.py e55d56 data.txt 0.2 4 0.034 0.032
2018-10-02 21:53:51 matei lang_model.py e55d56 data.txt 0.1 4 0.177 0.16

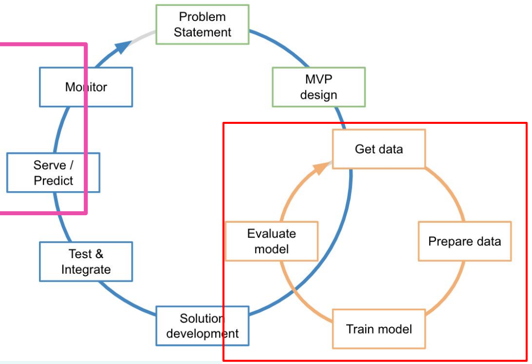
MLflow Tracking API: Simple & Pythonic

```
import mlflow
import mflow.tensorflow

# log model’s tuning parameters
with mlflow.start_run() as run:
  mlflow.log_param("layers", layers)
  mlflow.log_param("alpha", alpha)

  # log metrics and model
  mlflow.log_metric("mse", model.mse())
  mlflow.log_artifact("plot", model.plot(test_df))
  mlflow.tensorflow.log_model(model)
```

Experiments management

Tasks

- prepare data
- prepare experiment config
- run experiments
- browse history
- compare results
- share results
- documentation



Collaborate on ML Experiments with DVC

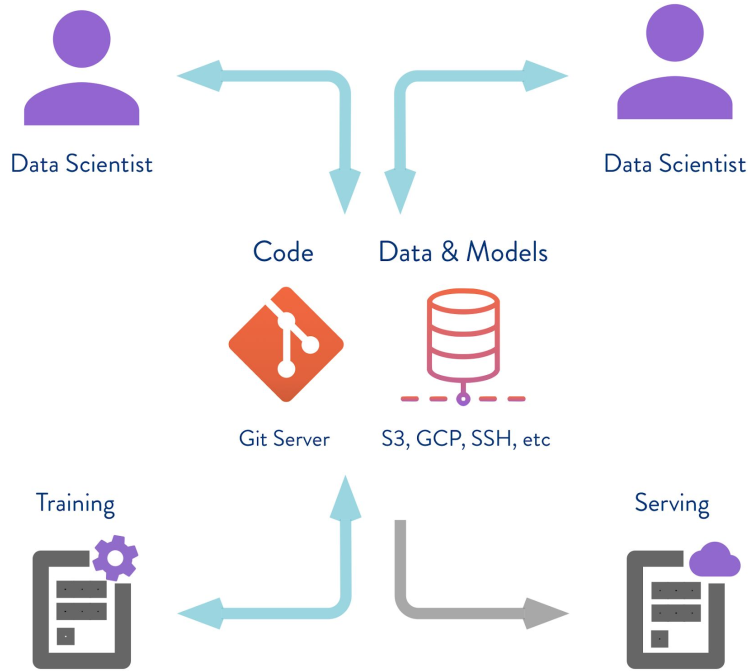
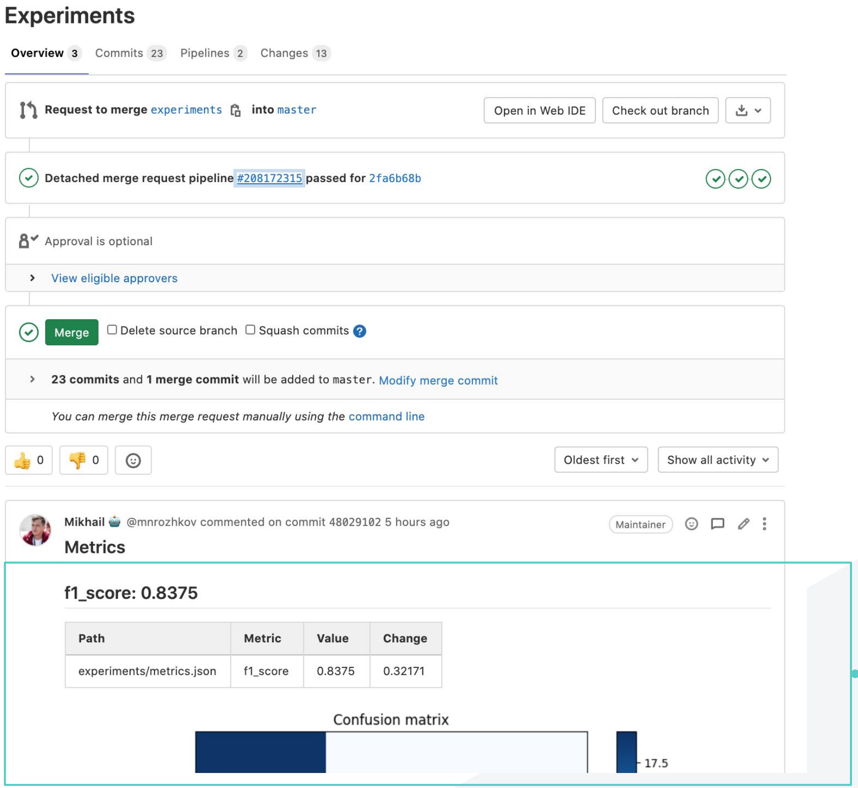
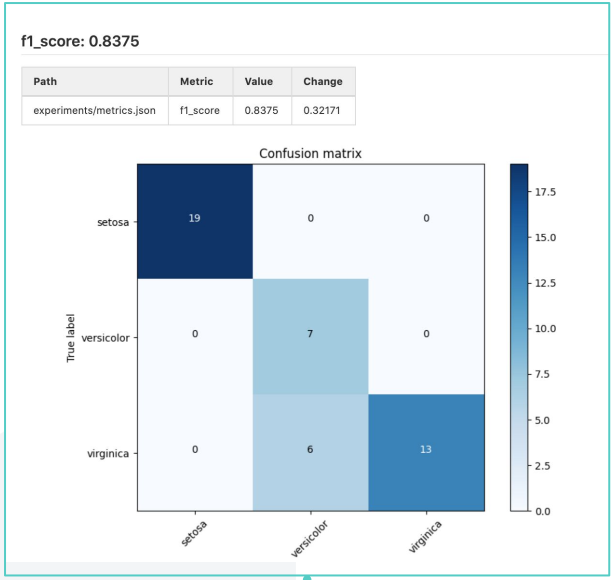


Image source: https://dvc.org/doc/use-cases/data-and-model-files-versioning

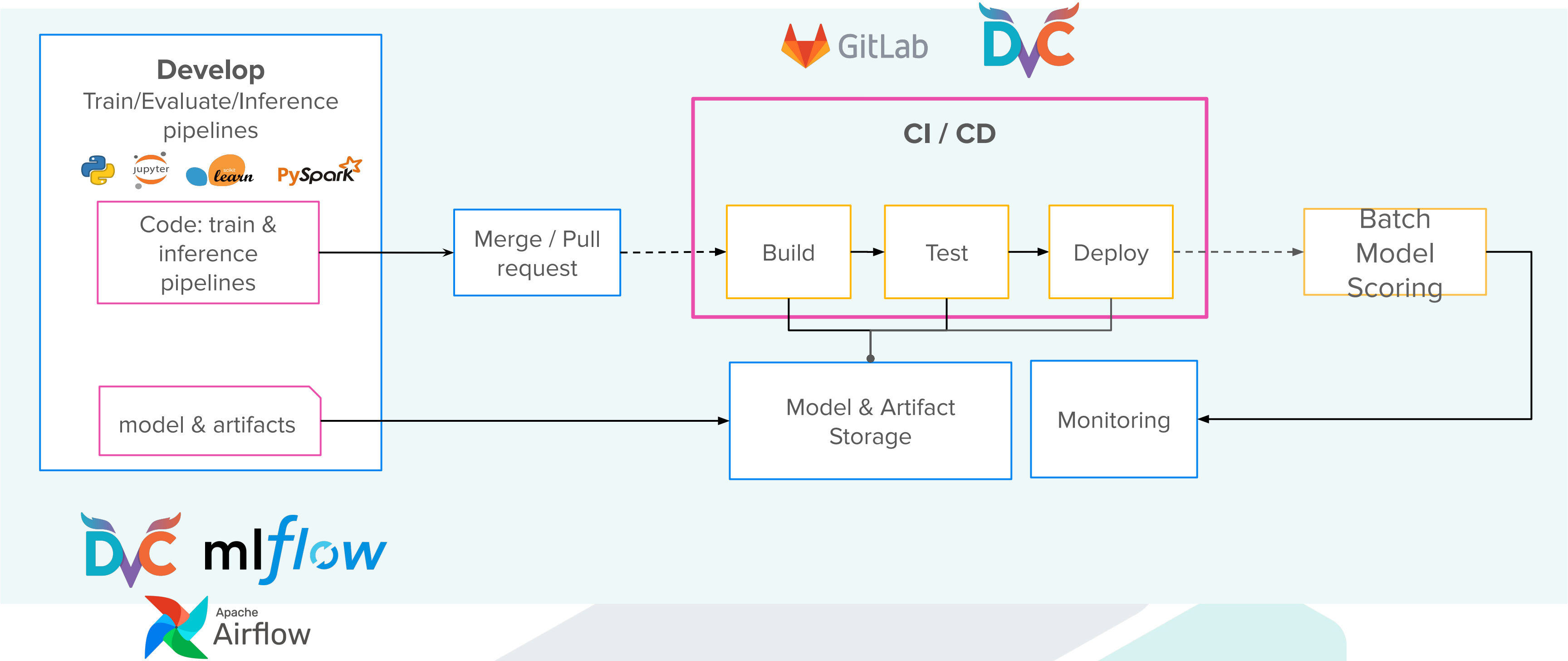
Auto reports for commits with GitLab & CML



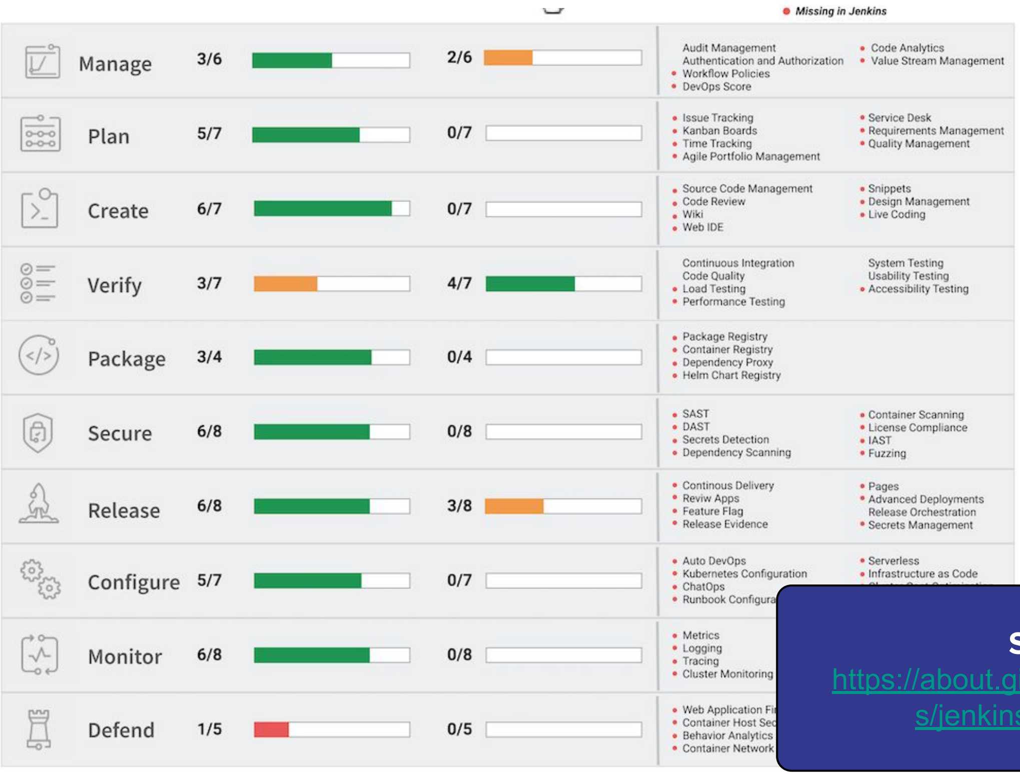


Инструменты вывода моделей в продакшн

Tools for CI/CD



Jenkins-GitLab Comparison Infographic

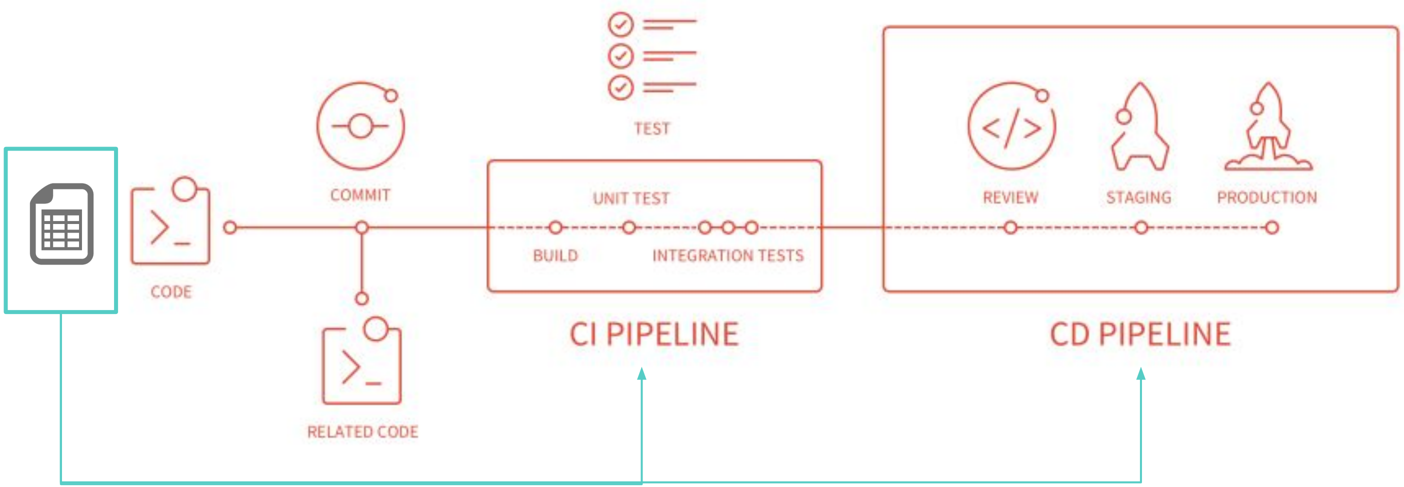


GitLab | Jenkins | Missing in Jenkins

	GitLab	Jenkins	Features	
Manage	3/6	2/6	Audit Management Authentication and Authorization Workflow Policies DevOps Score	Code Analytics Value Stream Management
Plan	5/7	0/7	Issue Tracking Kanban Boards Time Tracking Agile Portfolio Management	Service Desk Requirements Management Quality Management
Create	6/7	0/7	Source Code Management Code Review Wiki Web IDE	Snippets Design Management Live Coding
Verify	3/7	4/7	Continuous Integration Code Quality Load Testing Performance Testing	System Testing Usability Testing Accessibility Testing
Package	3/4	0/4	Package Registry Container Registry Dependency Proxy Helm Chart Registry	
Secure	6/8	0/8	SAST DAST Secrets Detection Dependency Scanning	Container Scanning License Compliance IAST Fuzzing
Release	6/8	3/8	Continous Delivery Reviw Apps Feature Flag Release Evidence	Pages Advanced Deployments Release Orchestration Secrets Management
Configure	5/7	0/7	Auto DevOps Kubernetes Configuration ChatOps Runbook Configura	Serverless Infrastructure as Code
Monitor	6/8	0/8	Metrics Logging Tracing Cluster Monitoring	
Defend	1/5	0/5	Web Application Fi Container Host Sec Behavior Analytics Container Network	

Source:
https://about.gitlab.com/devops-tools/jenkins-vs-gitlab.html

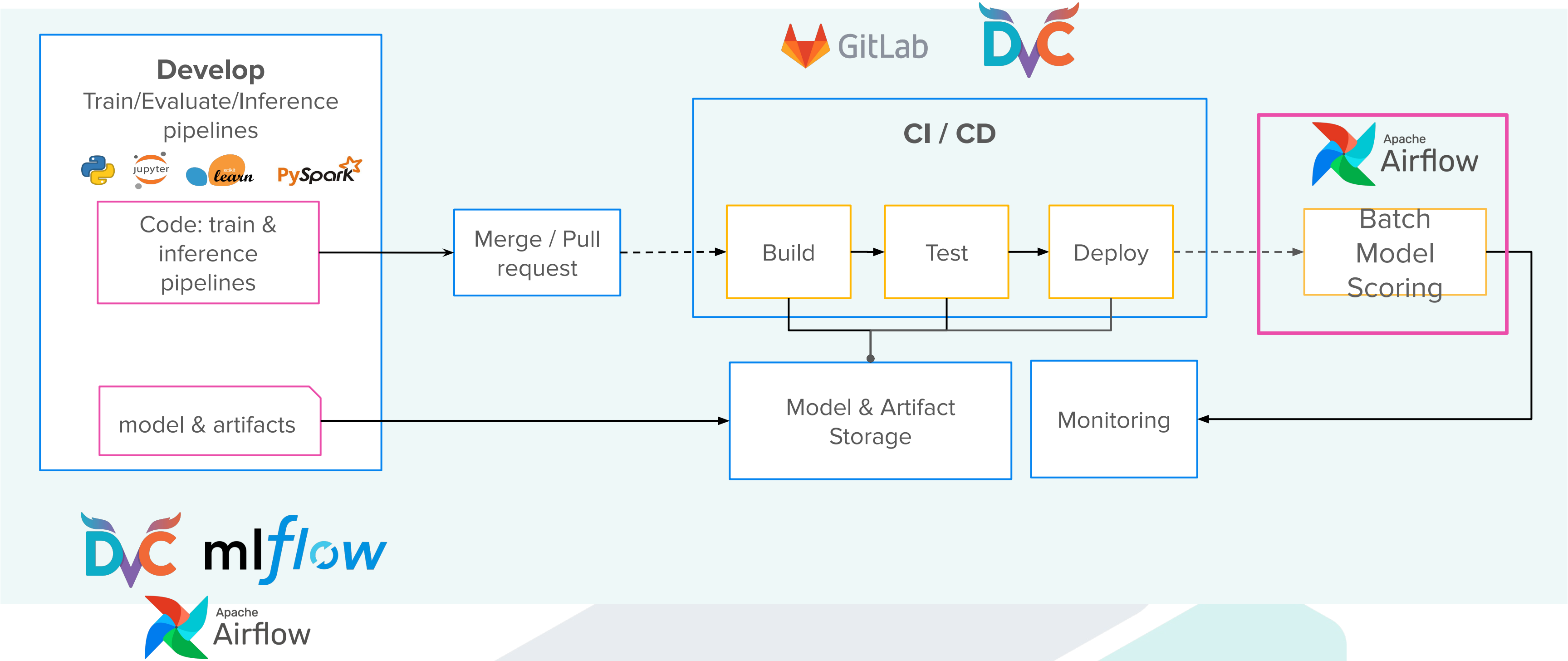
CI/CD pipeline



1. Store model / data / artifacts
2. Share access

References: https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/

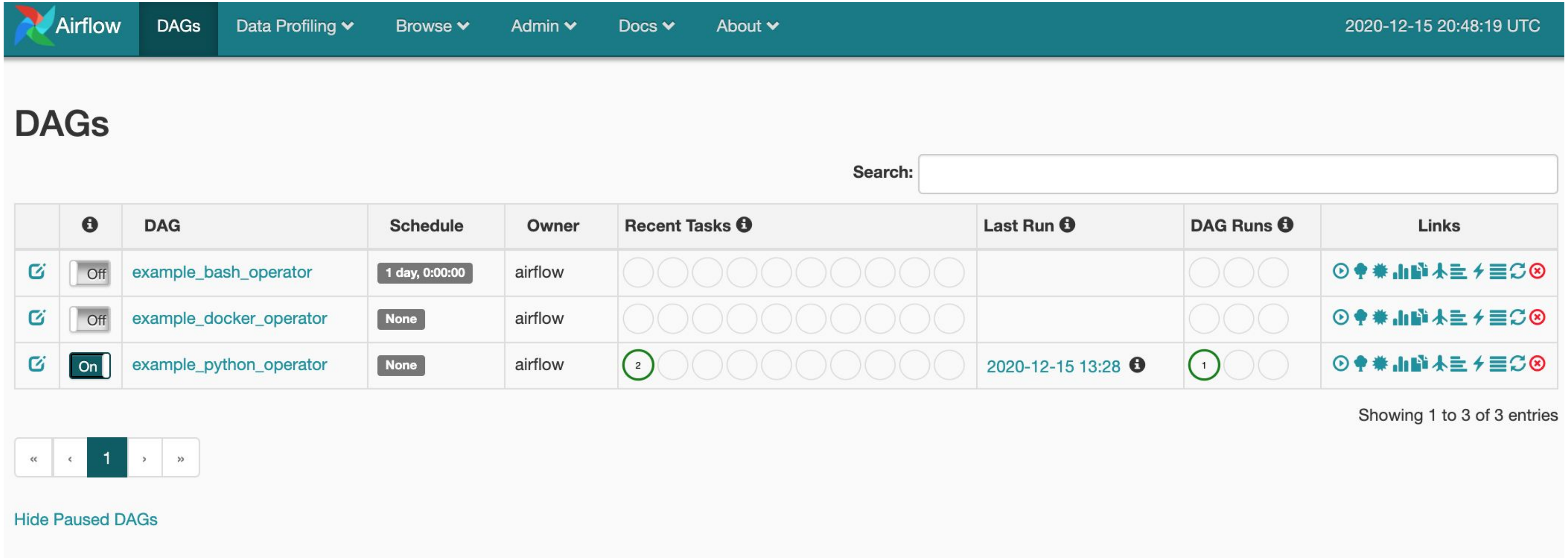
Tools for running Scoring



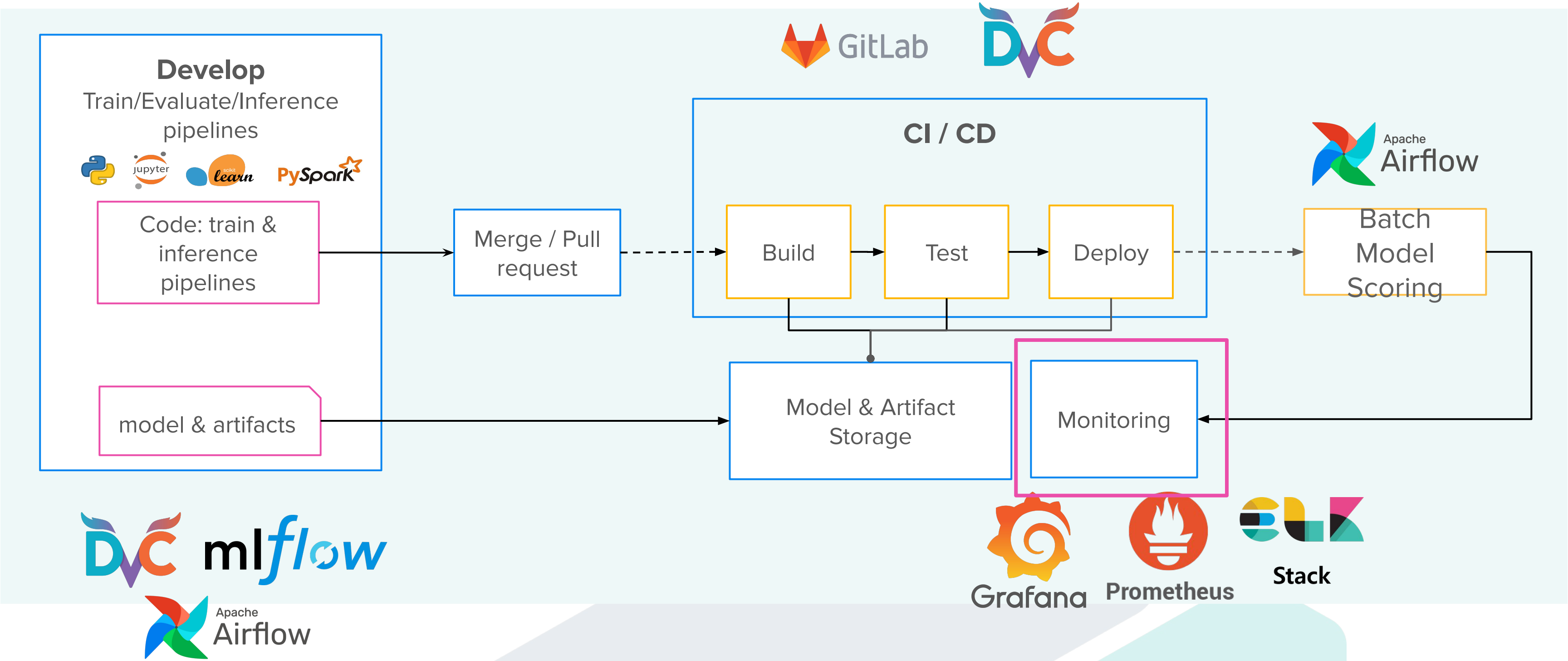
Airflow

- Airflow – это платформа для создания, мониторинга и оркестрации пайплайнов
- Python 2.7, 3.5, 3.6, 3.7 / Flask, Jinja / LDAP
- Запуск и мониторинг задач на неограниченном количестве машин
- Динамическая генерацией workflow из Python-кода
- Возможность связывать друг с другом любые базы данных и API с помощью как готовых компонентов, так и самодельных плагинов

Airflow UI


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2020-12-15 20:48:19 UTC
DAGs
Search:
DAG
Schedule
Owner
Recent Tasks
Last Run
DAG Runs
Links
Off
example_bash_operator
1 day, 0:00:00
airflow
Off
example_docker_operator
None
airflow
On
example_python_operator
None
airflow
2
2020-12-15 13:28
1
Showing 1 to 3 of 3 entries
1
Hide Paused DAGs

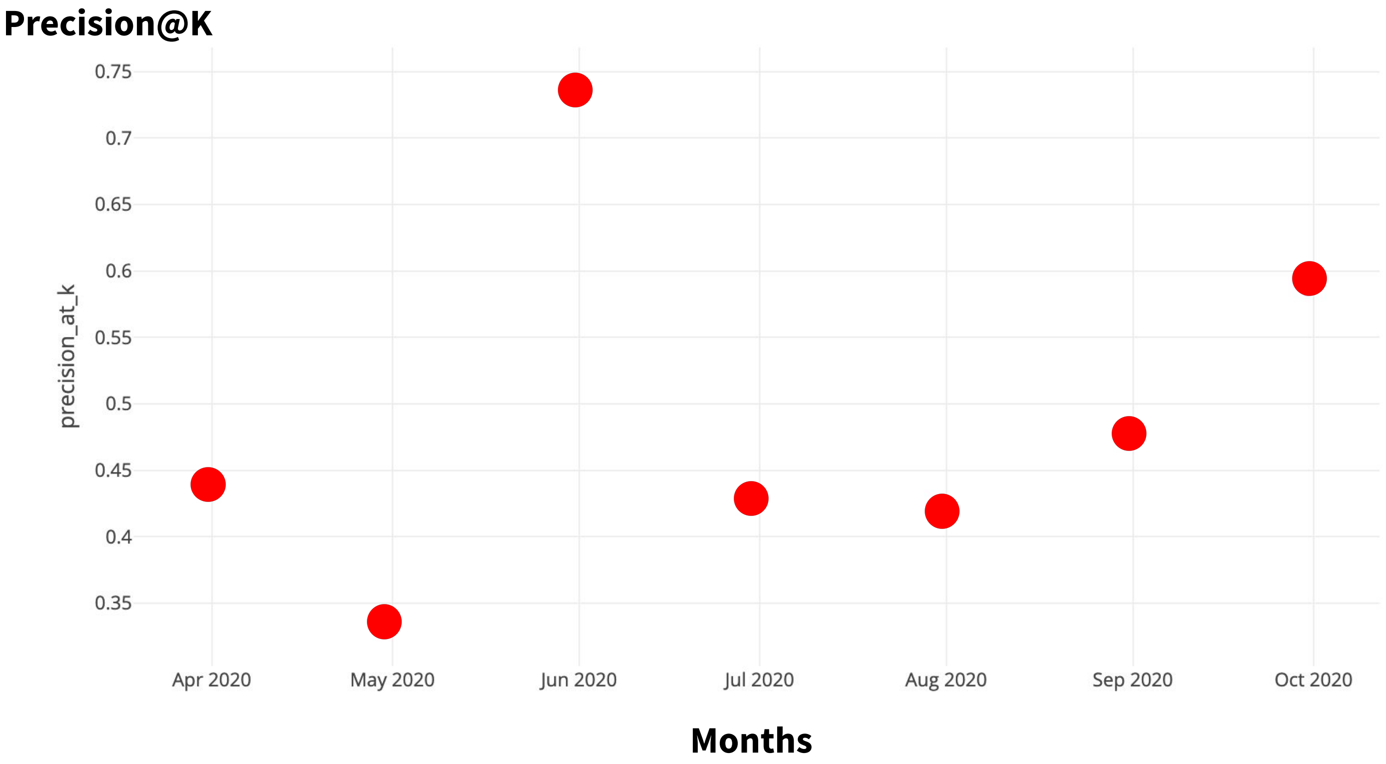
Tools for Monitoring



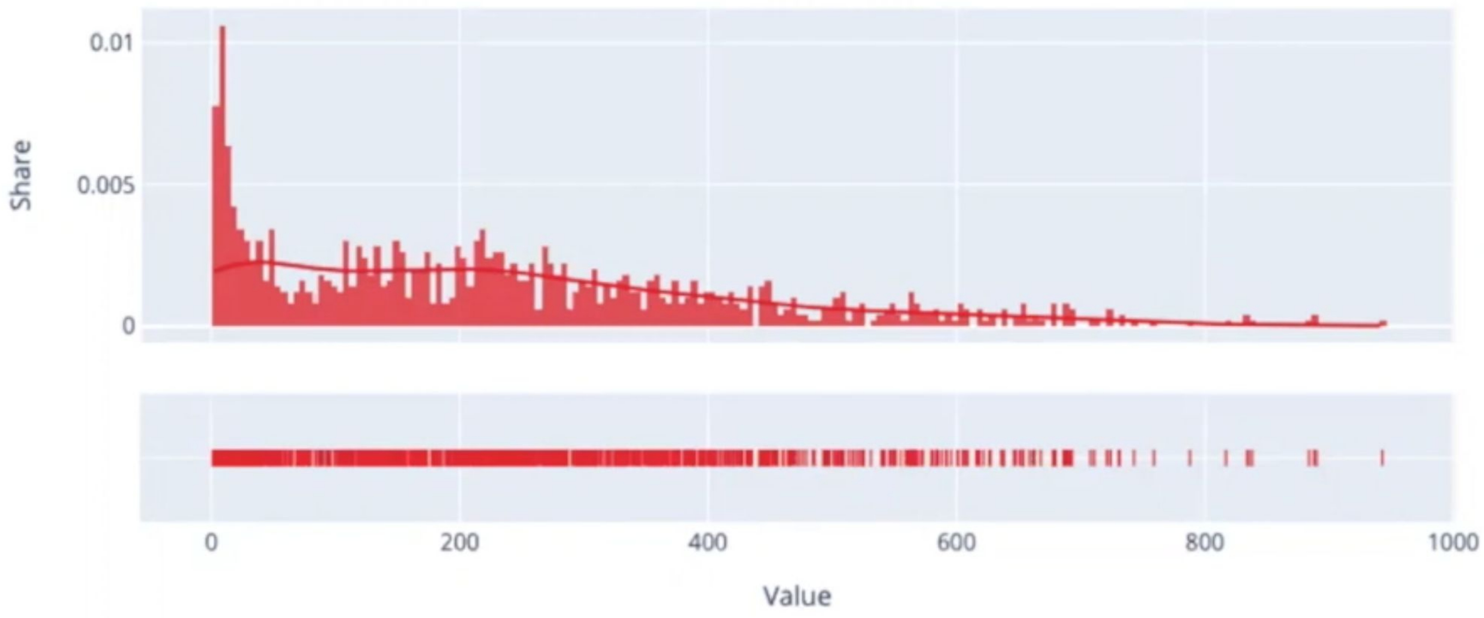
Monitor model, pipelines and data with Airflow, Grafana, Prometheus, ELK

Source: What's Next for Observability by Michelle Tan
https://grafana.com/blog/2019/10/21/whats-next-for-observability/

Metrics monitoring for each month

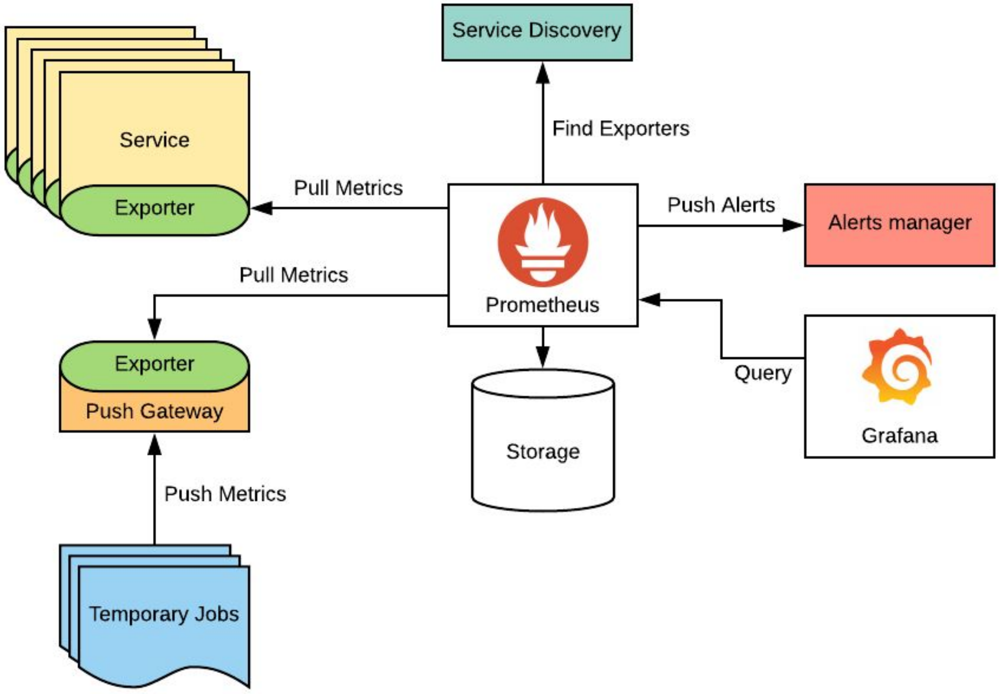


Model output distribution monitoring daily



References: Emeli Dral, The day after deployment: how to set up your model monitoring https://ml-repa.ru/datafest2020

Collect technical metrics with Prometheus



Prometheus

Grafana

Source: https://itnext.io/ephemeral-jobs-monitoring-using-prometheus-pushgateway-917b33486564

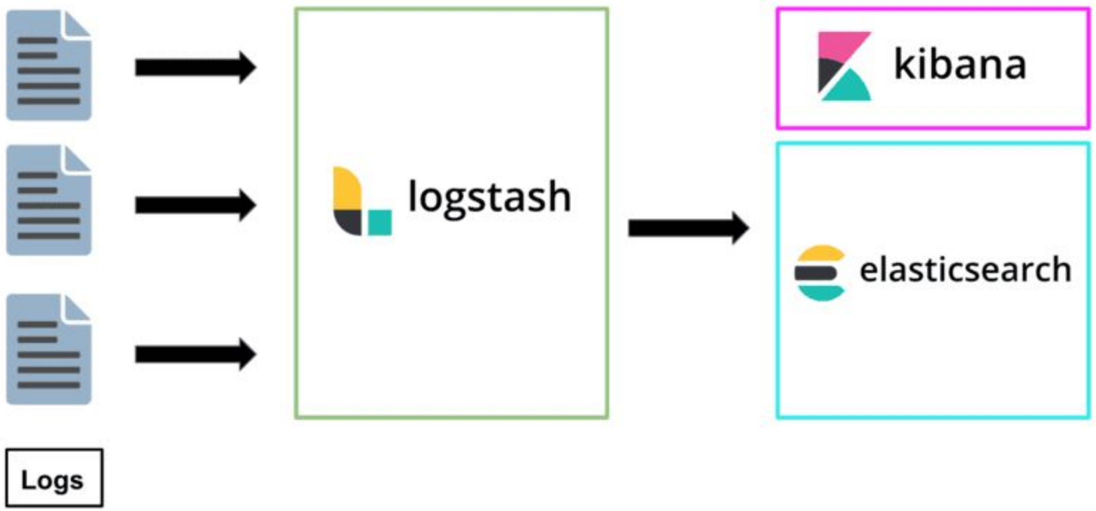
Visualize metrics with Grafana

Prometheus

Grafana

Source: https://itnext.io/ephemeral-jobs-monitoring-using-prometheus-pushgateway-917b33486564

Logs management with ELK Stack



Source: https://christophergs.com/machine%20learning/2020/03/14/how-to-monitor-machine-learning-models/

Пример 1:
MLflow для логирования метрик и артефактов экспериментов

https://github.com/mlrepa/mlflow-1-tracking

Thank you!

Time to start...