Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Scanflow

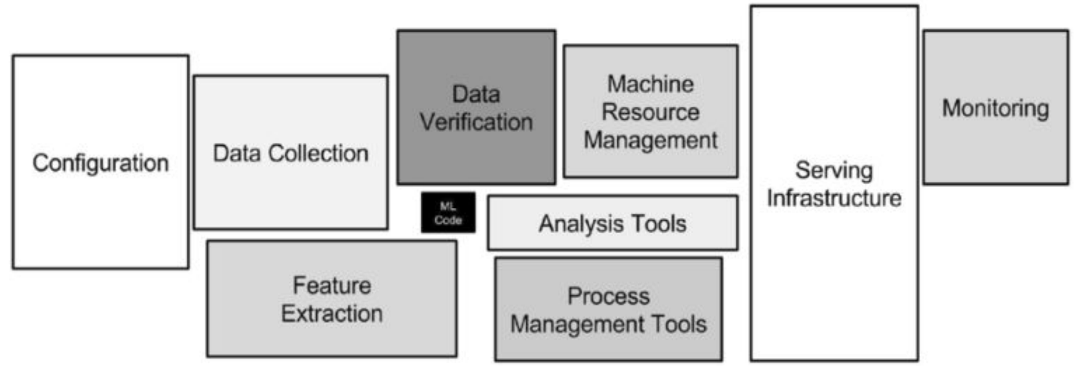
Scalable library for end-to-end ML workflow management

Lenovo-BSC collaboration

January 2020

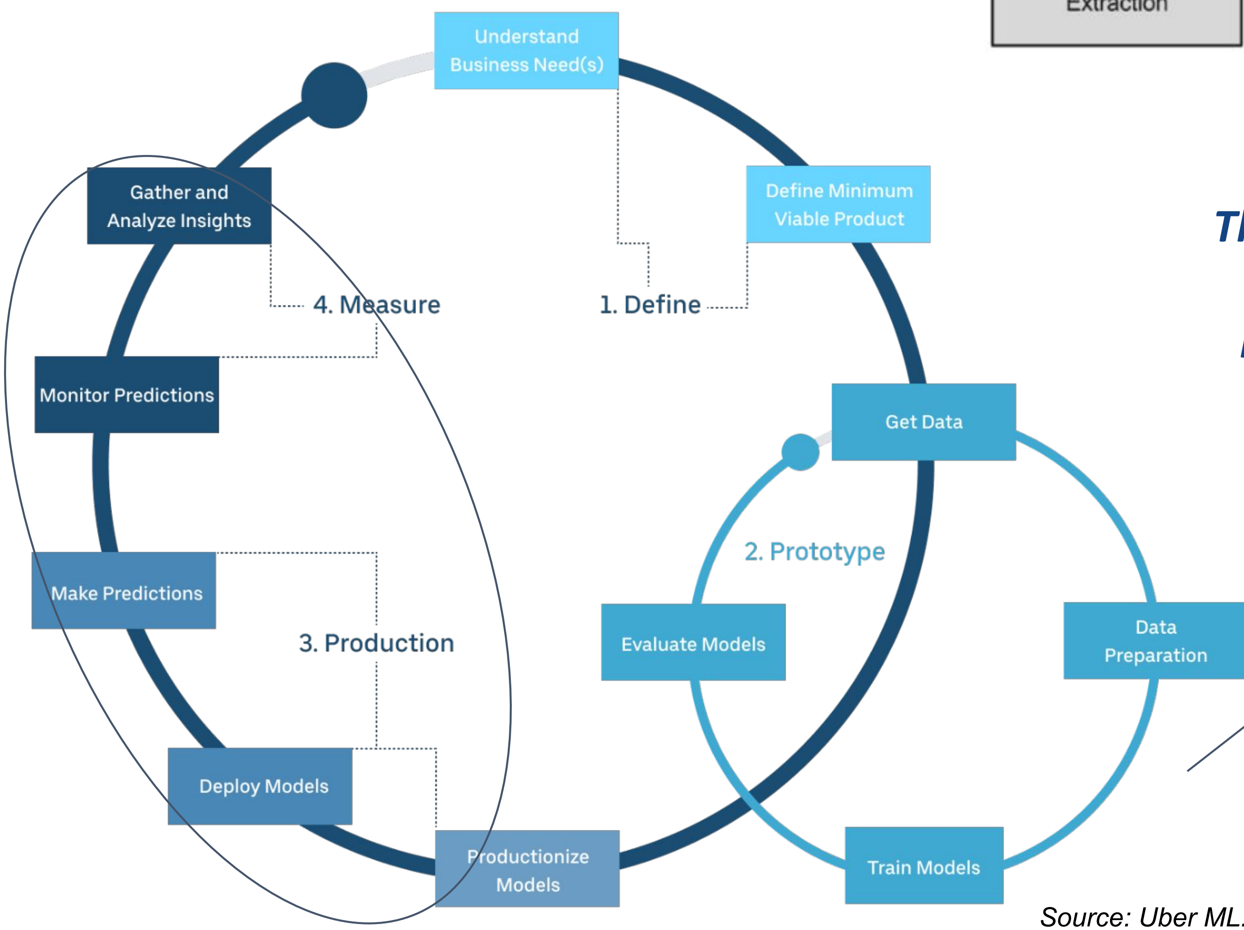
Applied Machine Learning: Workflow

Deployment: The ML code is a small part of the complete pipeline. More steps are needed to get it working on production.



Source: Sculley D., Holt. G.

The modeling phase is not the end of a workflow, more steps are needed.



Source: Uber ML.

Applied ML workflow: Packaging ML workflows for later use will decrease the prototyping and deployment time .

Applied Machine Learning: Issues

- Moving a complete workflow from development platform to another new platform can break things, e.g, operating system, libraries, dependencies, etc.
- Controlling a myriad of pipelines manually might be hard.
- Some steps in a workflow need different amount and type of computational resources, e.g, RAM, Storage, CPU, GPU.
- The complete workflow might scale from a single node to a cluster.
- The dataset distribution might change (normal, poison, etc, or different patterns). It is called distribution drift. An anomaly detector might help this.
- Some feature levels and balance of classes might change (categories, e.g, before {red, blue}, after {red, blue, black}. Classes, e.g, before {30% men, 70%women}. after {60% men, 40%women}).

Data scientists require tools for ease of deployment, tracking and reproducing experiments.

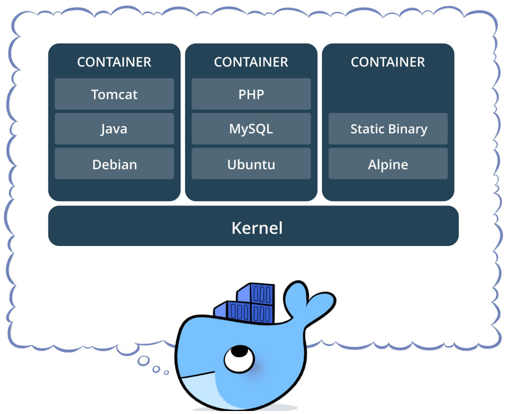
In production, many unexpected situations might arise such as drift distributions, unformatted data, etc.

How can the agent (model) and human collaborate to solve problems?

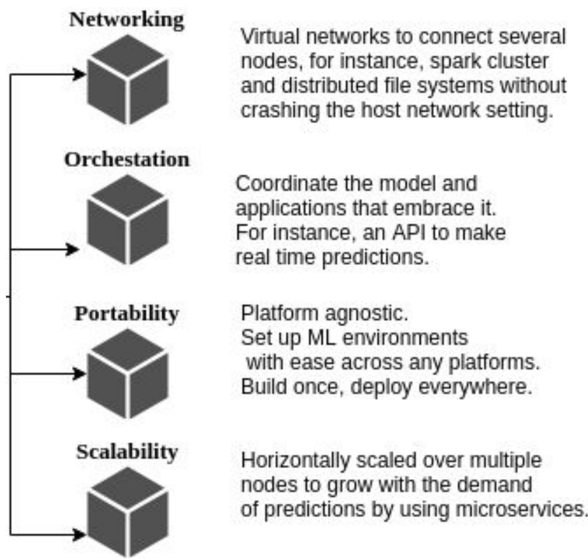
We need Interactive Machine Learning tools

Applied Machine Learning: Recipes

Docker: Isolate environments, portability, scalability, affinity, etc.



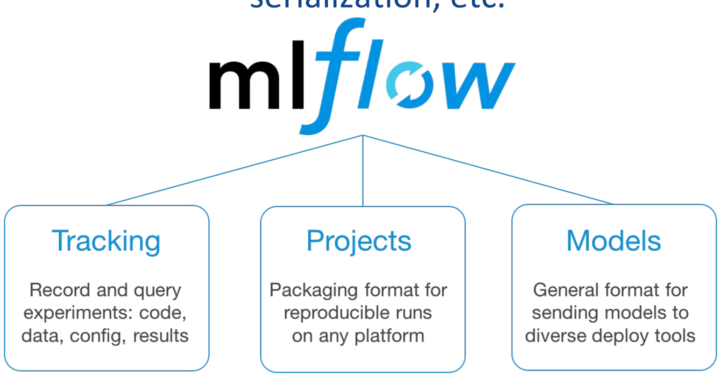
Source: arquitectoit.com



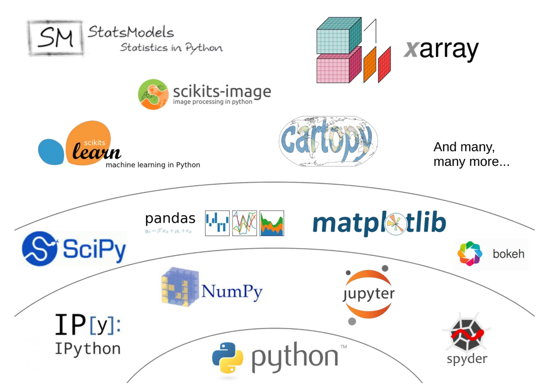
<u>Scanflow</u> is a high-level library that is built on top of these tools to manage and supervise workflows efficiently.

Python: Fast prototyping and expressiveness. Robust AI ecosystem.

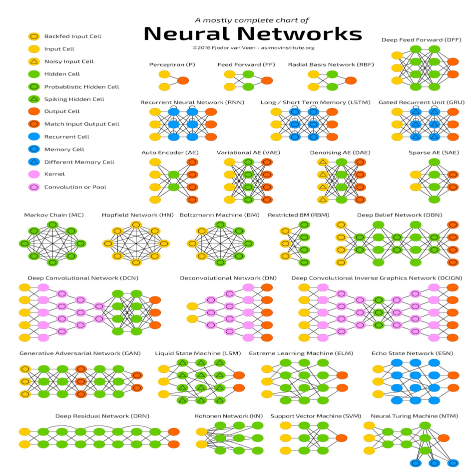
MLflow: Track metrics, organize projects, model versioning and serialization, etc.



Source: mlflow.org

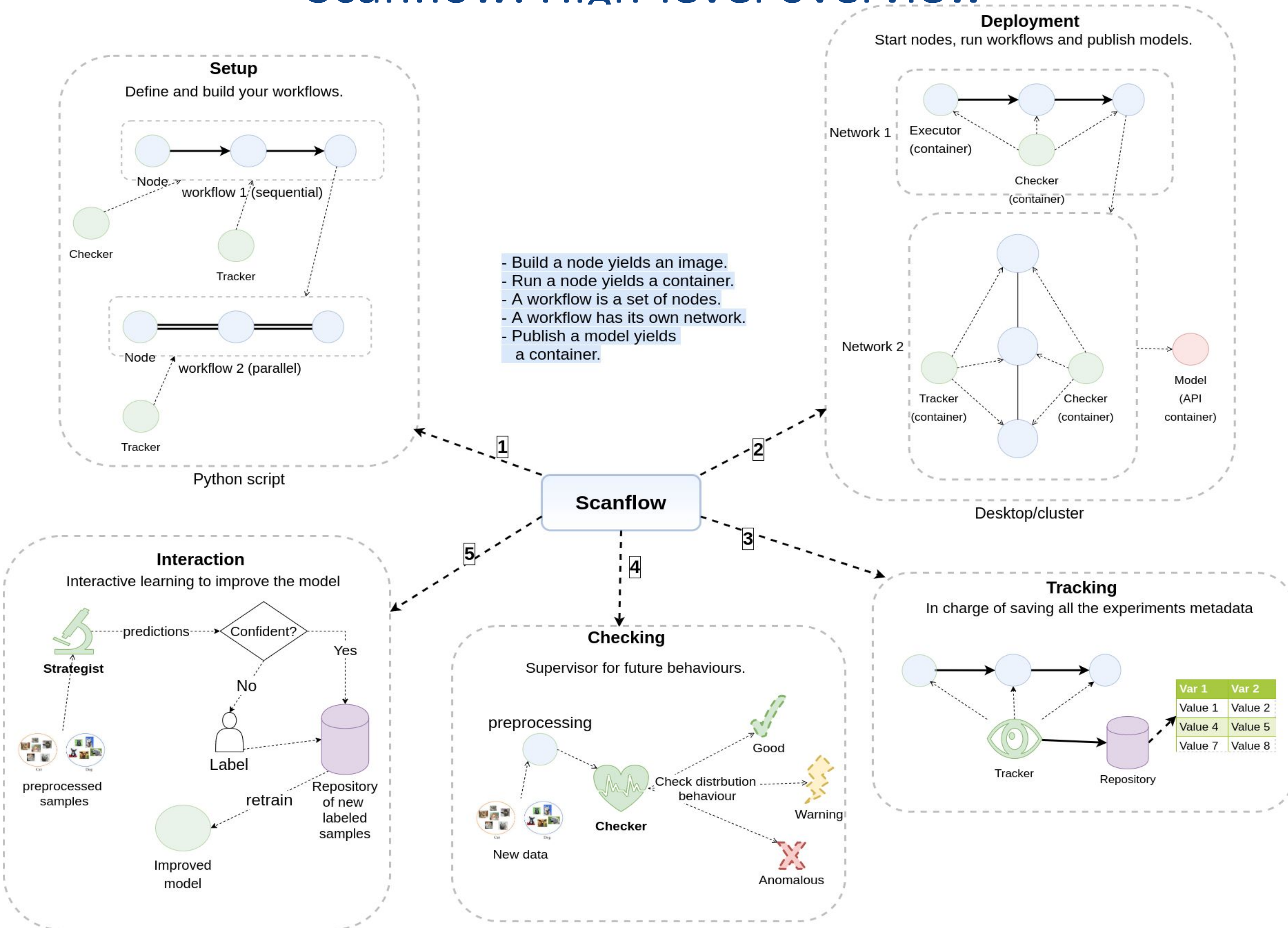


Source: leblancfg.com

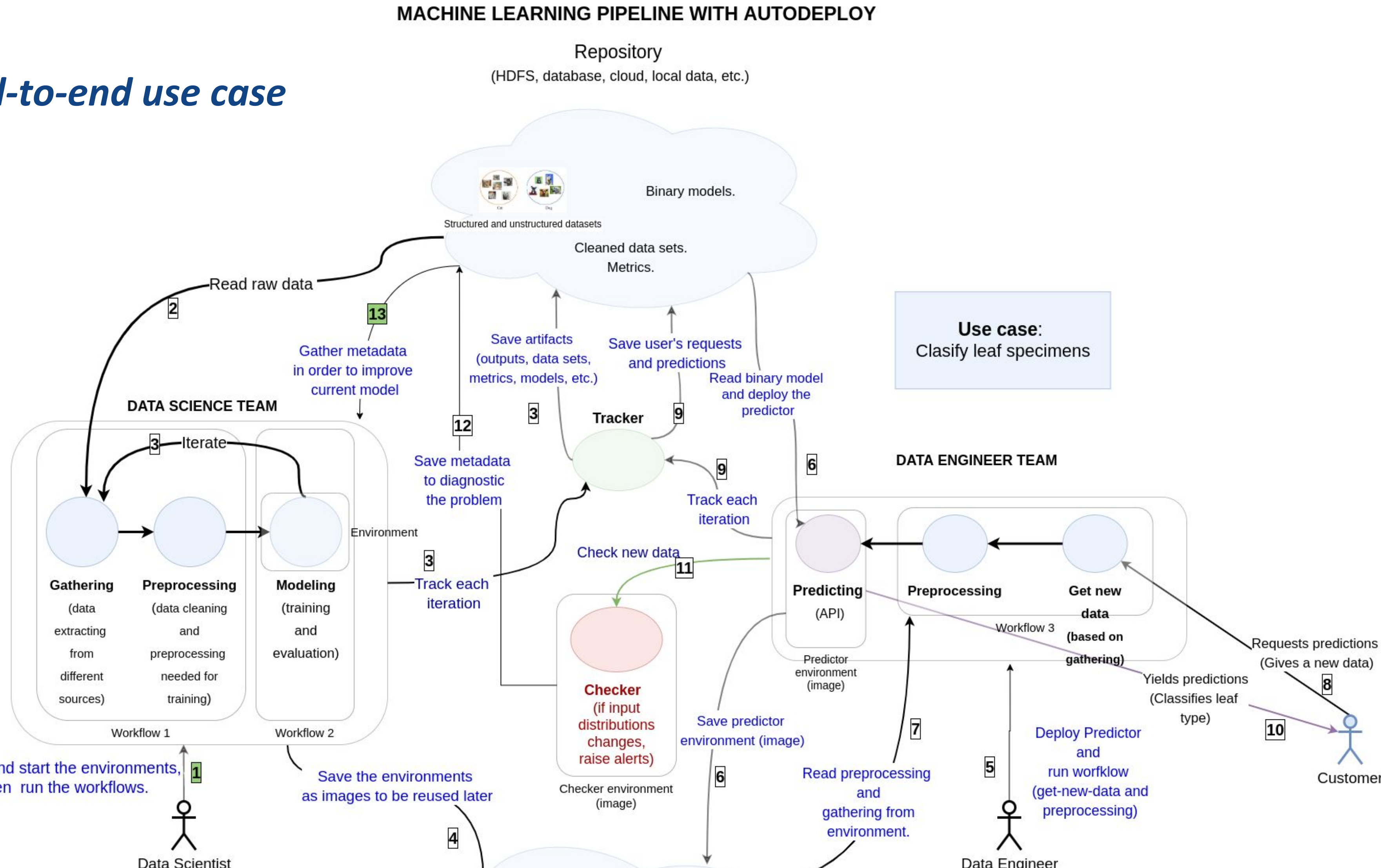


Source: fjodor van veen - asimovinstitute.org

Scanflow: High-level overview

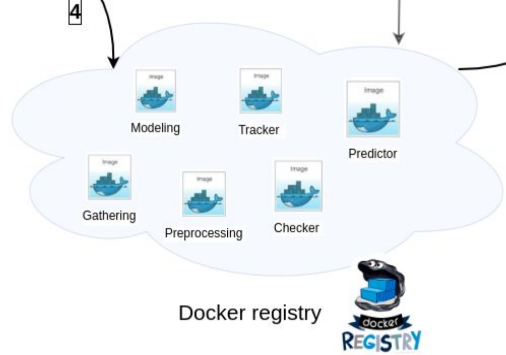


End-to-end use case



Benefits:

- Reproducible pipelines on any platform.
- Version control for each run.
- Track pipelines to see performance degradation.
- Easy communication between pipelines.
- Easy to program (just python code).
- Modularity, easy to add ML plugins.
- Integration with orchestration tools.

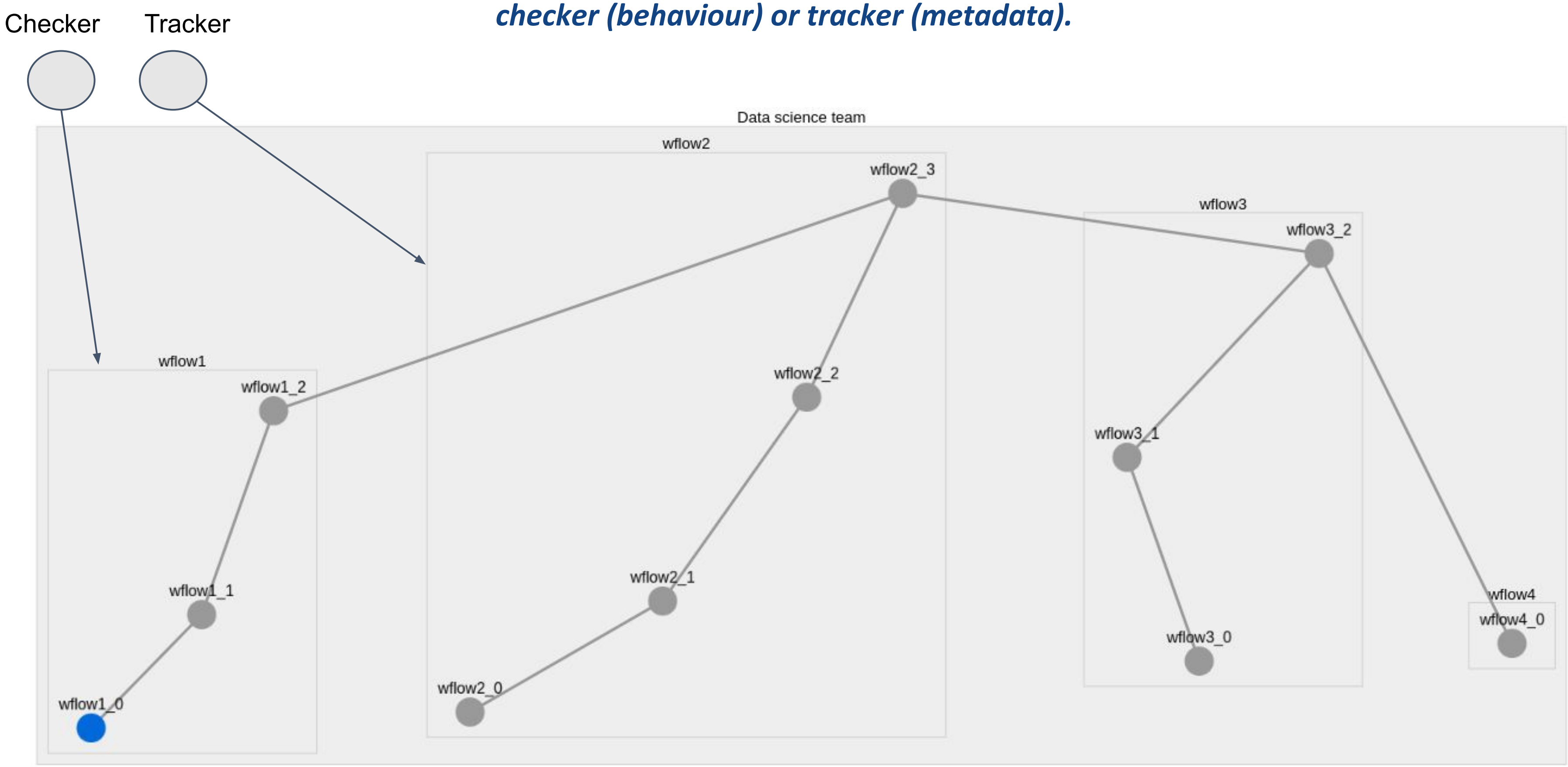


Docker own features for ML

- Faster time to market (faster delivery)
- Developer productivity. Dependencies
- Deployment velocity.
- IT infrastructure reduction (workload density)

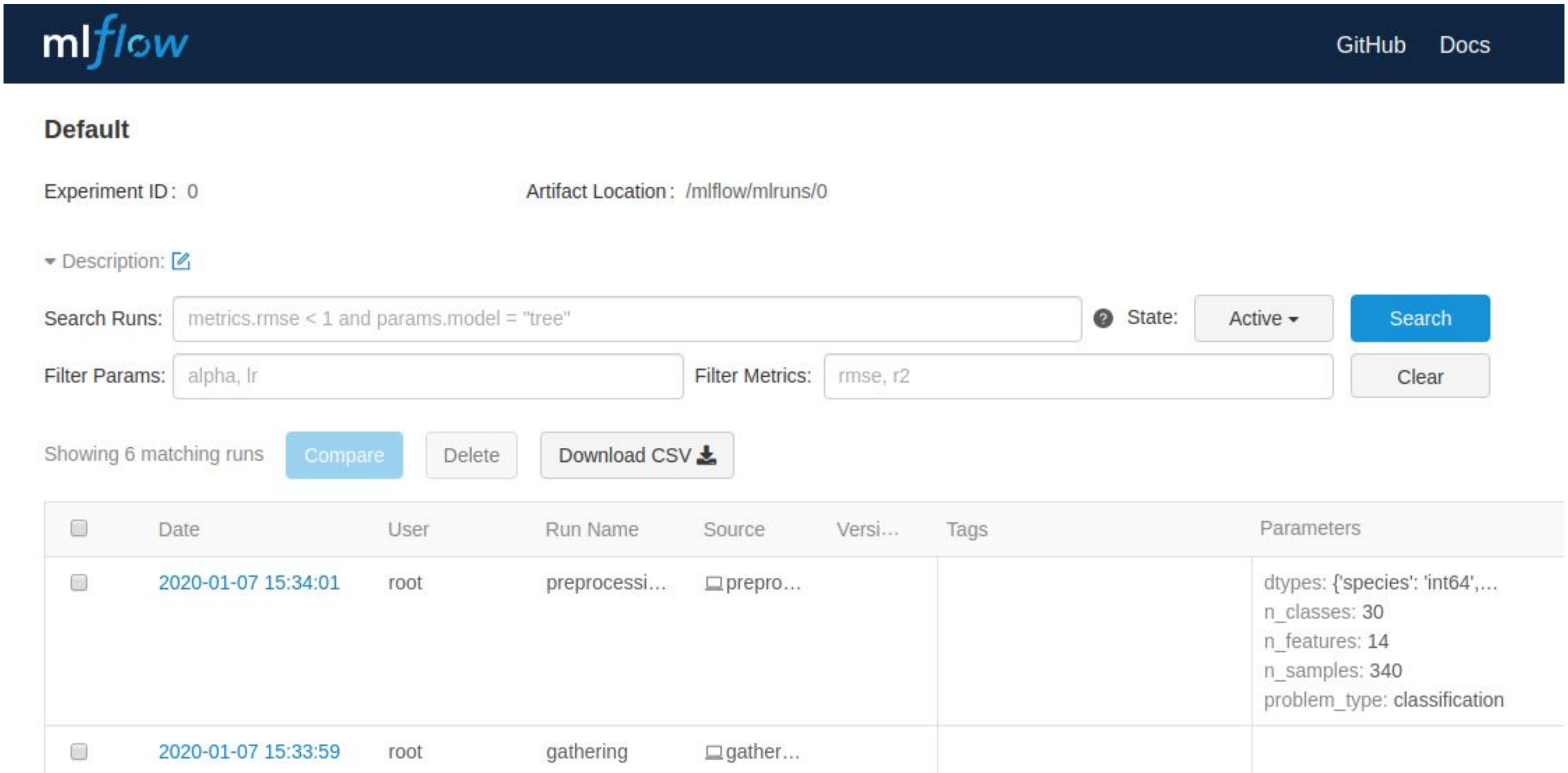
Setup and deployment

Design nested and parallel workflows: *Each node is a computation function, e.g preprocessing or modeling. They can be run following the user ordering design. Besides, each node is a docker container (environment) inside a workflow (rectangle in the following picture). Finally, each workflow has its own checker (behaviour) or tracker (metadata).*



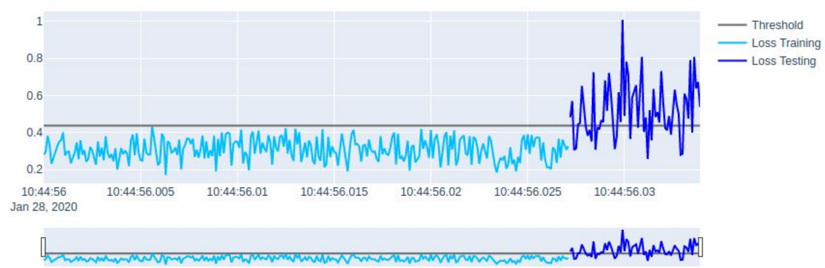
Tracking: Save any workflow metadata for future analysis

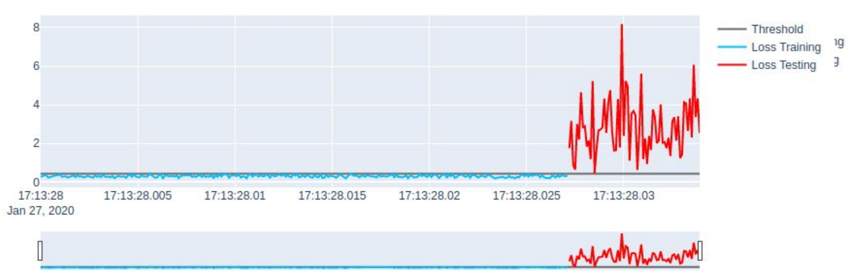
This module logs intermediate results belonging to a workflow such as, settings, metrics, statistics, scores, etc.



Checking: supervise workflow's behaviour in production

Early drift distribution detector (built-in function)





Custom plugins can be added to check the behaviour of any workflow: for sanity, integrity, anomaly, interpretability, etc.

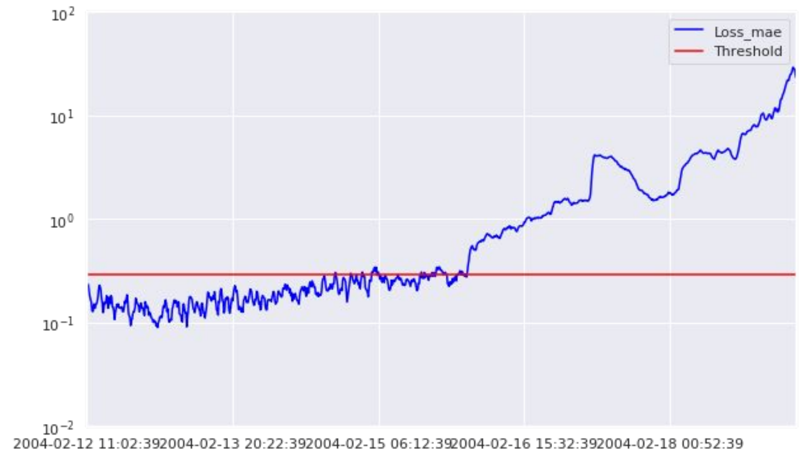
Any plugin is supposed to be run in real time

Early drift distribution detector: architecture

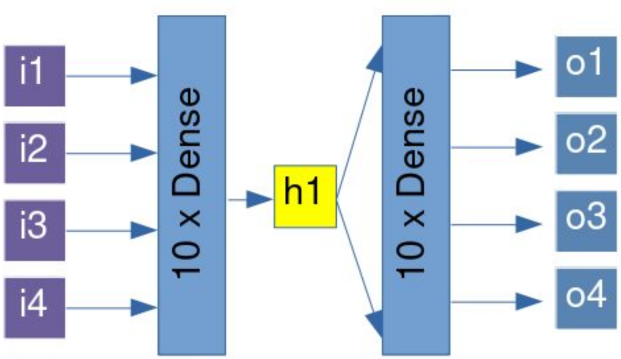
Reconstruction error is calculated to measure how different is a new distribution from the original one

Input

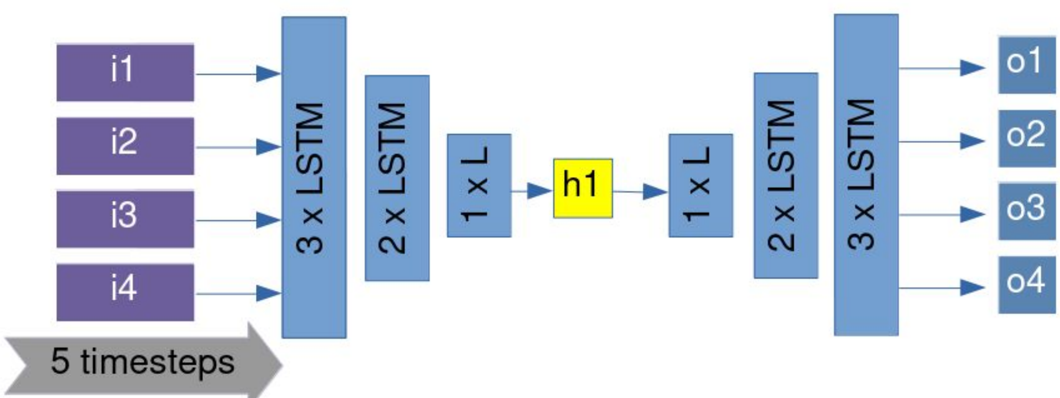
Output



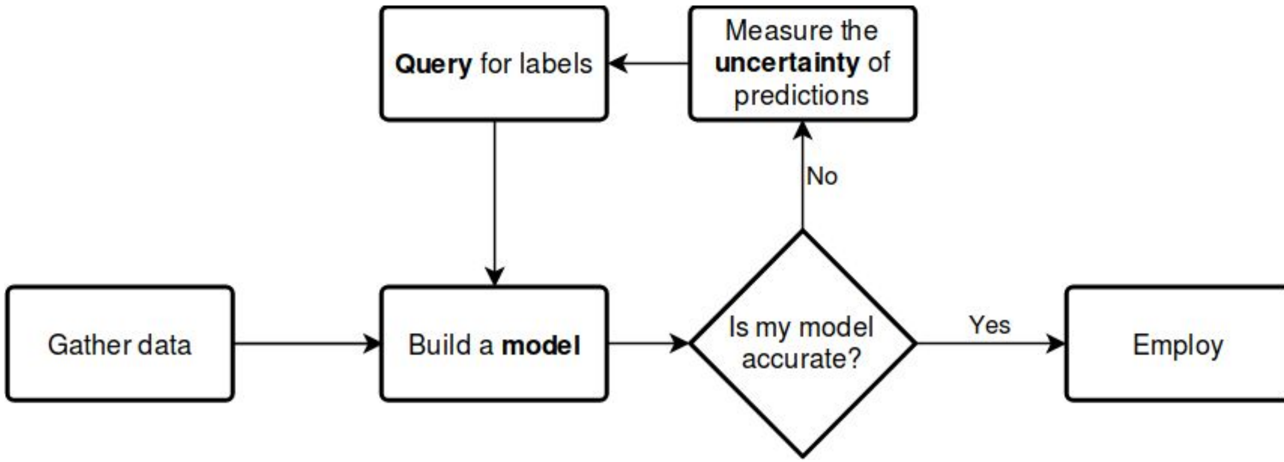
Naive Autoencoder



LSTM Autoencoder



Interact: Interactive learning for improving the model

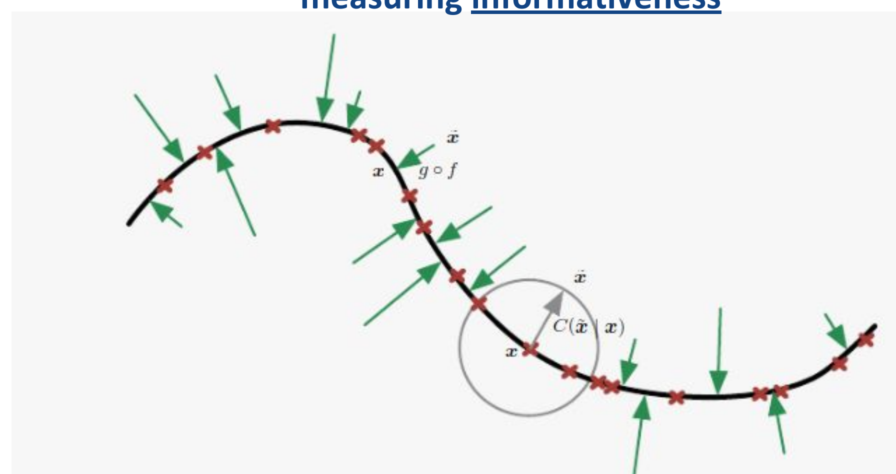


Source: modAL

Our objective is to find out the __best strategy__ to measure __uncertainty__. It means, find the samples that are more informative to the model

Our first approach is to build a __deep denoise autoencoder__ that can select the samples that have __more entropy__

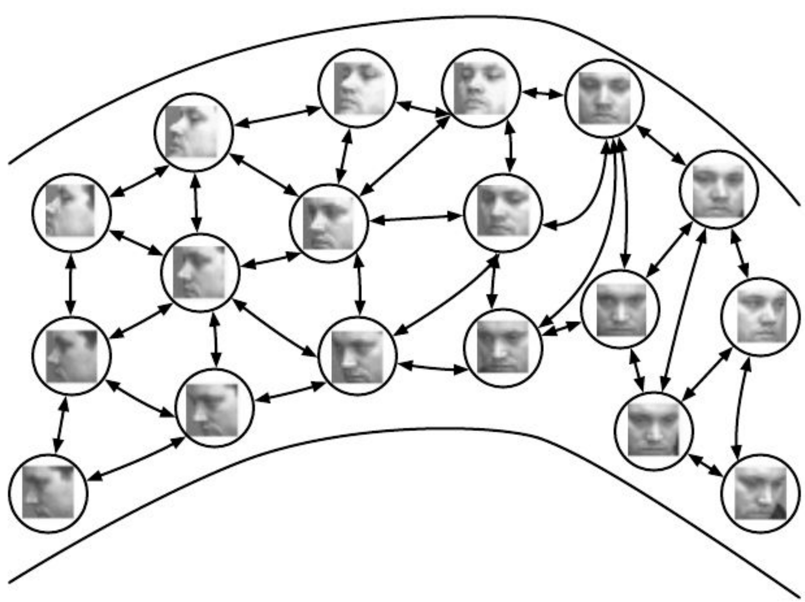
Here we learn a latent space that can get the more __useful features__ from the original distribution to the aim of measuring __informativeness__

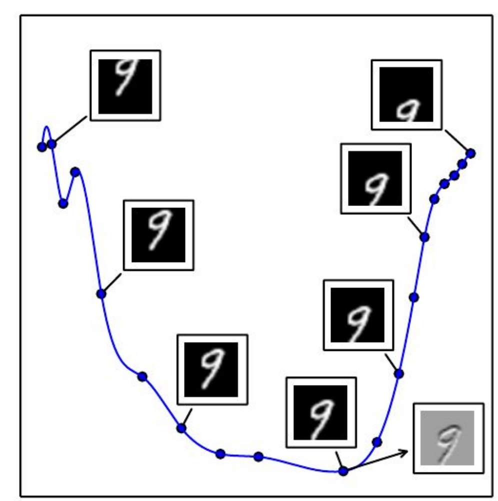


Source: Deep learning book, autoencoders. Ian Goodfellow at el.

Interact: Interactive learning for improving the model

Having a good latent space means that we have learnt a low dimensional space that best represents the expected behaviour. E.g, in the picture, each face belongs to the same man, even if they have different representations (e.g, rotations). With this we can detect, in inference stage, those samples that are far from this space. Our hypothesis is that these points add more information to the current model.

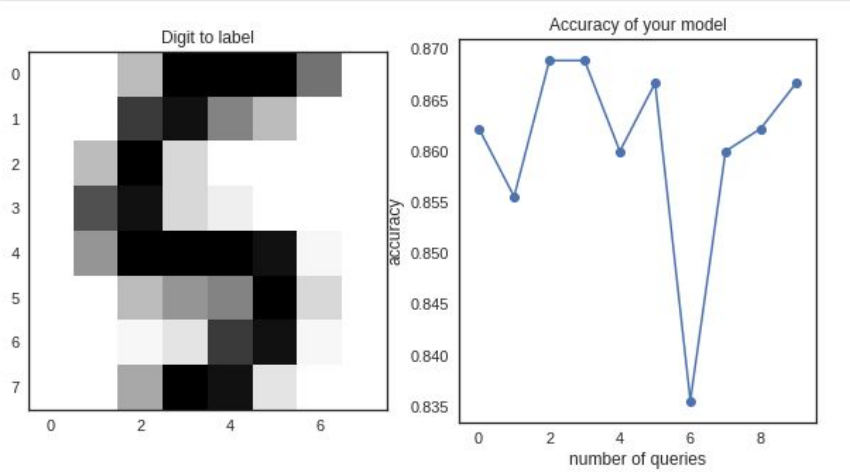




Source: Deep learning book, autoencoders. Ian Goodfellow at el.

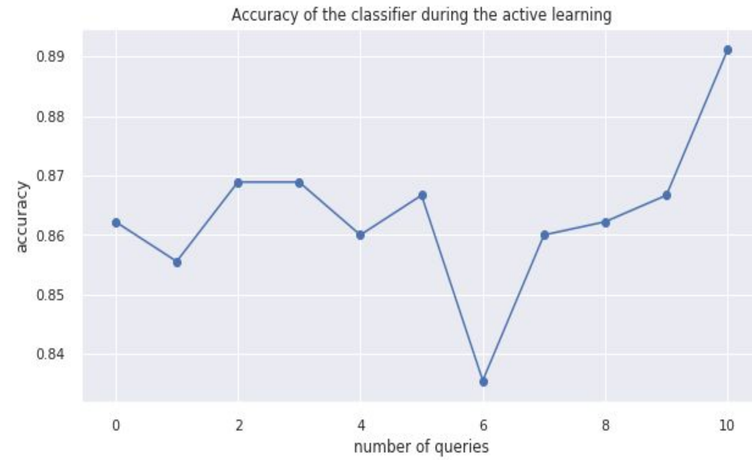
Interact: Use case I

Example using the MNIST dataset



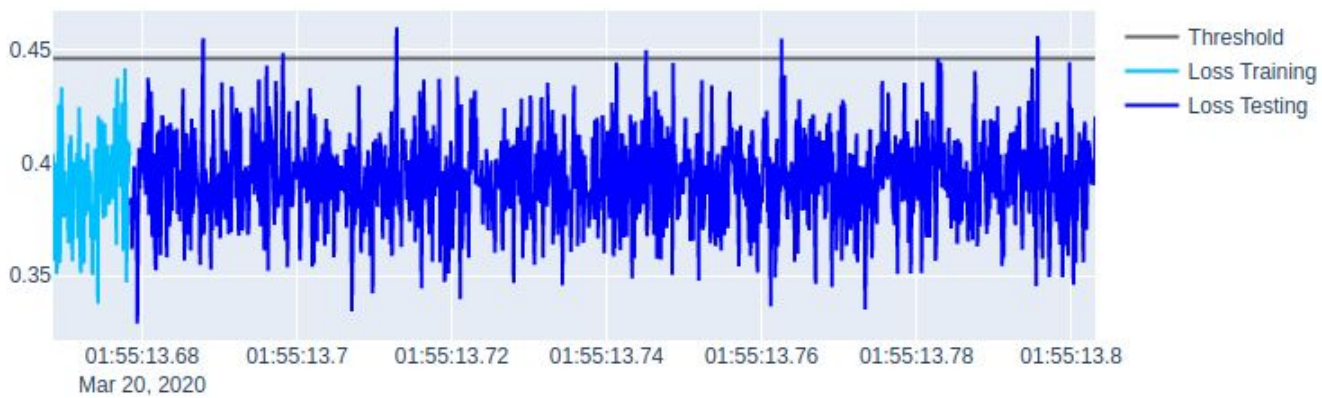

```
Which digit is this?
 5
```

Label just the set of samples that brings more information to the model



After labeling, the model has filled out gaps in the feature space, therefore, it's improved.

The points above the threshold (gray line) are the chosen.



Interact: Use case II

Current model: A model built with only pedestrian scenario as a input. Other scenarios will be anomalous.

Normal Clip

Pedestrians

Abnormal Clip

A car

Abnormal Clip

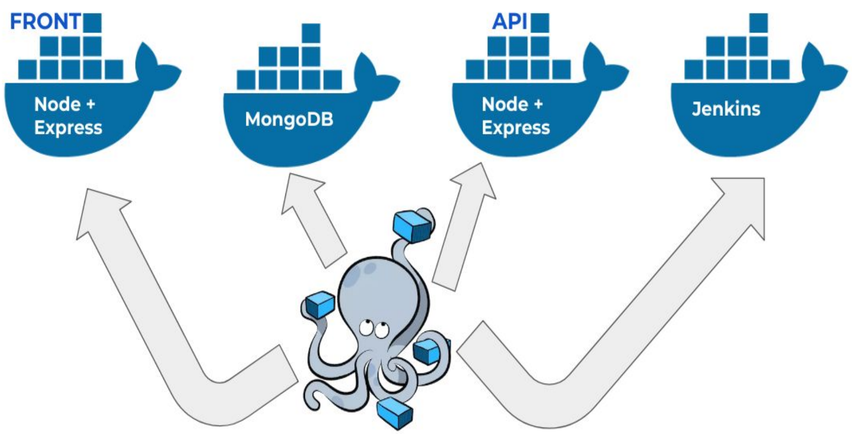
A bike

<u>Before labeling</u>. A bike is detected as anomalous. False positive. Score: 75%

<u>After labeling.</u> A bike is detected now as normal. Score: 90%

Deployment integration: Docker compose

Docker compose behaviour



Source: medium.com

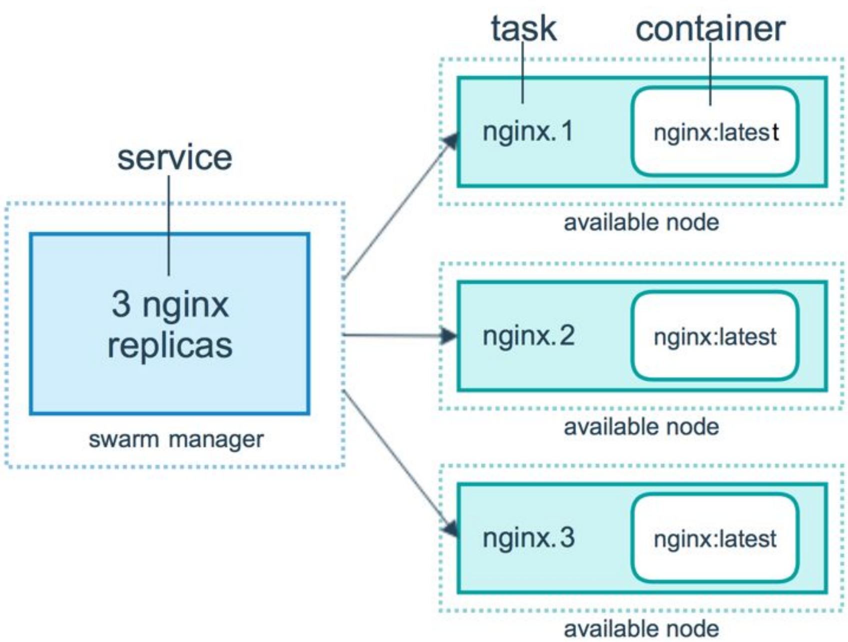
docker-compose.yml

```yaml
version: '3'
services:
  get_new_data:
    image: get_new_data
    container_name: get_new_data-20200126033621
    networks:
    - network-workflow3
    depends_on:
    - tracker-workflow3
    environment:
      MLFLOW_TRACKING_URI: http://tracker-workflow3:8003
    volumes:
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/:/app
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/ad-stuff/ad-tracker/tracker-workflow3:/mlflow
    tty: 'true'
  preprocessing_new_data:
    image: preprocessing_new_data
    container_name: preprocessing_new_data-20200126033621
    networks:
    - network-workflow3
    depends_on:
    - tracker-workflow3
    environment:
      MLFLOW_TRACKING_URI: http://tracker-workflow3:8003
    volumes:
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/:/app
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/ad-stuff/ad-tracker/tracker-workflow3:/mlflow
    tty: 'true'
  tracker-workflow3:
    image: tracker-workflow3
    container_name: tracker-workflow3-20200126033621
    networks:
    - network-workflow3
    volumes:
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/ad-stuff/ad-tracker/tracker-workflow3:/mlflow
    ports:
    - 8008:8003
networks:
  network_workflow3: null
```

Once finished defining the workflows, they can be saved as a docker-compose file

Deployment integration: Docker Swarm

Docker Swarm behaviour



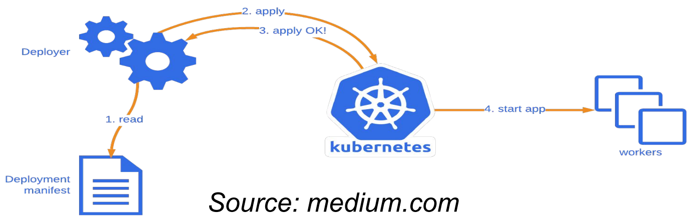
Source: filepicker.io

Scanflow provides a docker-stack file to deploy a swarm cluster to schedule the workflows.

Docker Swarm console

```
[xgbravo@nxt2027 compose_repo]$ docker service ls
ID             NAME                              MODE         REPLICAS   IMAGE                               PORTS
5r1l6yx27pfk   my_swarm_gathering                replicated   1/1        gathering:latest
xke5uf9aqdh4   my_swarm_modeling                 replicated   1/1        modeling:latest
ro9haibzt9ma   my_swarm_preprocessing            replicated   1/1        preprocessing:latest
pvdud6whi4pg   my_swarm_tracker_workflow1        replicated   1/1        tracker_workflow1:latest            *:8006->8001/tcp
alnxiq8y3tfs   my_swarm_tracker_workflow1_scale  replicated   0/5        my_swarm:latest
2dh4zpwu48od   my_swarm_tracker_workflow1_scale2 replicated   0/5        my_swarm_tracker_workflow1:latest
ilzhcp546xft   my_swarm_tracker_workflow1_scale3 replicated   5/5        tracker_workflow1:latest
ygwulqaah8a8   my_swarm_tracker_workflow2        replicated   1/1        tracker_workflow2:latest            *:8007->8002/tcp
```

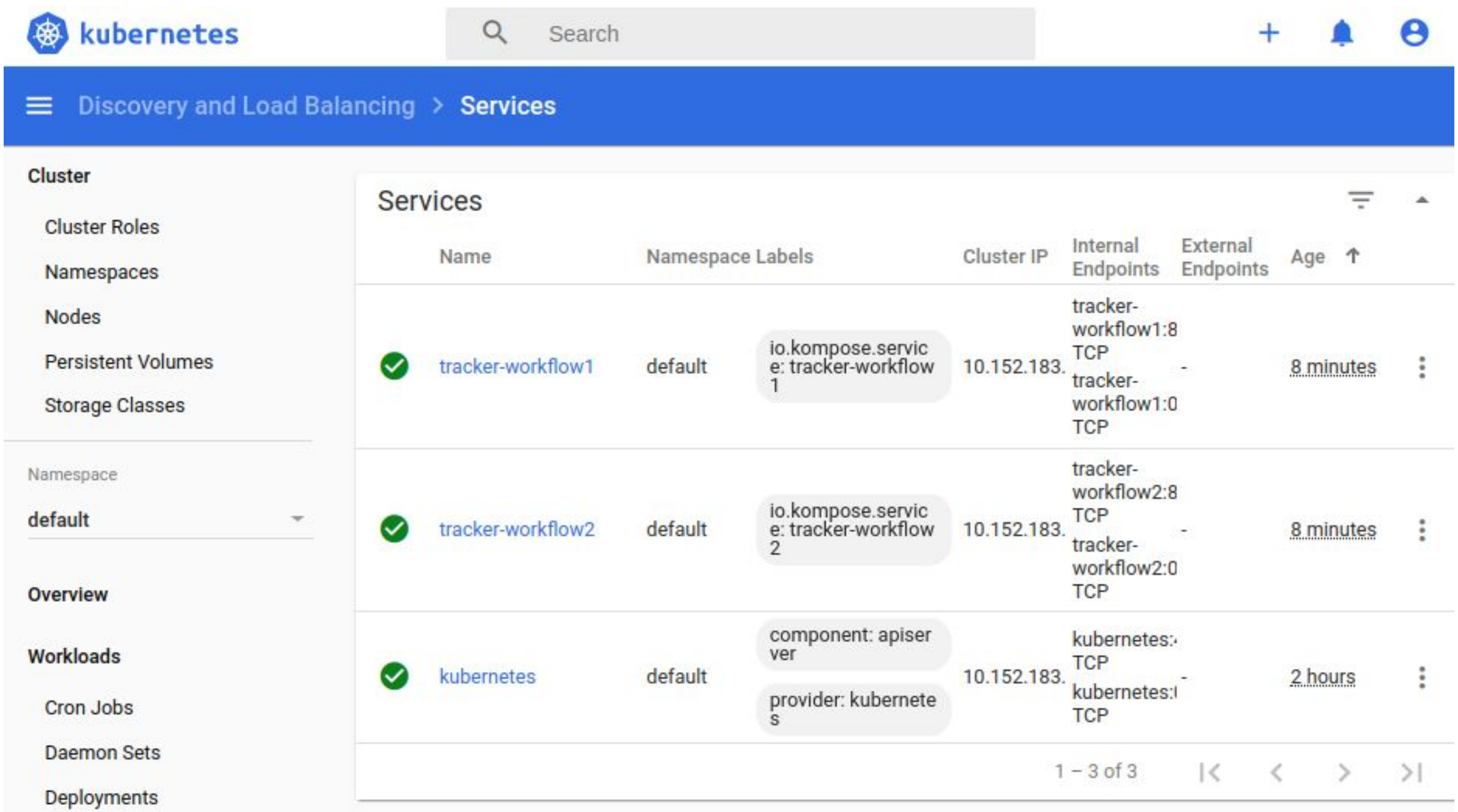
Deployment integration: Kubernetes



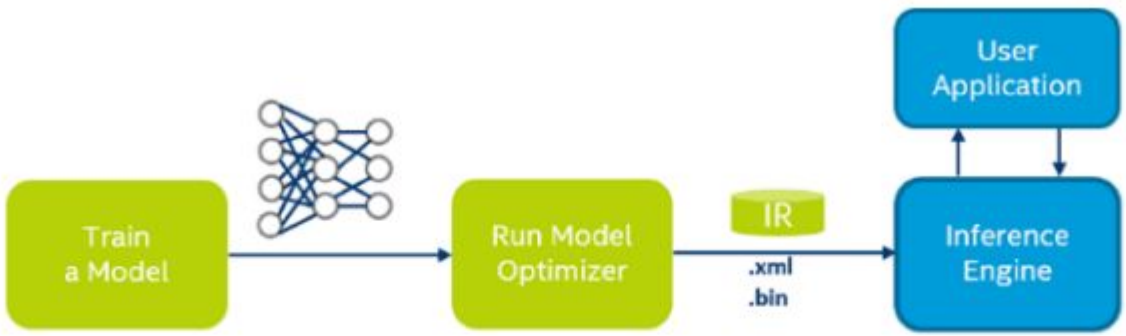
Source: medium.com

Scanflow provides a kube.config file to deploy a kubernetes cluster to schedule the workflows.

Kubernetes dashboard



Deployment integration: Intel OpenVINO



Intel tries to accelerate the model inference by lowering numerical precision training and inference

-However, adapting current working nodes and workflows to these optimizer nodes is not direct.

- Our proposal is to develop interfaces that facilitate this integration.

Proposal

TODO

- Transitional interfaces between states (e.g, the arrows in: model -> optimized_model -> inference), in order to make it simpler.

3. Transitional interfaces for [inference - Checking] and [inference - Interact] modules.

Benefits:

1. Standarization of communication between nodes, therefore, converting a model without much effort.
2. Flexibility to choose which workflow to run: a workflow that best fit a certain platform, such as desktop, car, raspberry pi, etc. For instance:

```
if device is typeA:
  run workflow A,
elif (device is typeB):
  run workflow B1 or run workflow B2
else
  run workflow C1 then run workflow C2 then workflow C3
```

3. Depending on traffic, switch models. For example:

```
if inference_time is critical:
  pick model A,
elif (accuracy is critical)
  pick model B
else pick model  C
```

Comparison ML workflow tools

	MLflow	Scanflow	Kubeflow/Airflow
Usability	Medium	**High**	Low
Built-in scheduling	No	No	Yes
Dynamic execution	No	**Yes**	Yes
Experiment tracking	Yes	Yes	Yes
Model versioning	Yes	Yes	Yes
Model checking	No	**Yes**	No
Orchestration-agnostic	Yes	**Yes**	No

Most appropriate tools for:

- Pre-deployment (steps required for getting a model): MLflow, Scanflow.
- Deployment (put a model into production): MLflow, Scanflow, Kubeflow/Airflow.
- Post-Deployment (check the model's health): Scanflow.
- Ease of use: Scanflow > MLflow > Kubeflow/Airflow.

Scanflow's main goals are usability, integration for deployment and real-time checking

Is it relevant?. AI predictions for 2020.

Creator of pytorch: … " place more value on AI model performance beyond accuracy. "

Celeste Kidd, psychologist at the University of California, Berkeley: … " increased awareness of the real-life implications of tech tools … "

Jeff Dean, Google AI chief: … " he wants to see less of an emphasis on slight state-of-the-art advances in favor of creating more robust models. "

Anima, Anandkumar, NVIDIA: … " self-supervision, and self-training methods of training models, which are the kinds of models that can improve through self-training with unlabeled data. "

Dario gil, IBM: …" focus on metrics beyond accuracy to consider the value of models deployed in production. Shifting the field toward building trusted systems instead of prioritizing accuracy above all else will be a central pillar to the continued adoption of AI. "

Keywords: **robust models, interpretable models, trusted models**.

Source: https://venturebeat.com/2020/01/02/top-minds-in-machine-learning-predict-where-ai-is-going-in-2020/

Current status

- Creation of nested workflows where each node can be an executor, tracker or checker. DONE
- Isolation of each workflow using docker, it comprises auto-creation of networks, volumes, docker files, images, containers and registries. DONE
- Tested integration with docker-compose.
- Ongoing development on swarm and kubernetes integration.
- Deploy ML models into services for prediction.
- Tracking module for saving any metadata. DONE
- Checking module (for now, drift distribution plugin) for post-deployment. DONE
- **Developing Interact module**.

Future work

- Interaction module: it will in charge of the interaction between the model and the human with the aim of make the former better. **Human-in-the-loop intelligent systems.**
- Lightweight integration with additional third-party tools.
- Enhance interface module for checking.
- Dashboard for checking module.
- Improve compatibility with docker-compose, swarm, kubernetes.
- Test more use cases.
- Start formalizing scanflow for writing a paper.
- Start writing documentation.
- Set up scanflow as a python package.
- Improve drift distribution checker.
- Add option for wipe out any metadata (tracking and checking).
- Add a new plugin for integrity checking.
- Add option to compress all the settings needed to transfer an application.
- Add interface to write on databases.
- Add option to plot nested workflows in jupyter notebooks.