OSW2021
OPENSOURCEWEEK

# AI: avere un buon modello è solo l'inizio

**Filippo Minutella**
AI Engineer, Larus Business Automation

#OSW2021

RIOS RETE ITALIANA OPEN SOURCE

OSW2021
OPENSOURCEWEEK

# Introduzione MLOps

RIOS
RETE ITALIANA
OPEN SOURCE

# DevOps e MLOps

RETE ITALIANA OPEN SOURCE

## DevOps

Insieme di pratiche ben consolidate che si preoccupano di gestire il ciclo build-deploy-monitor di un progetto

## MLOps

Estensione delle metodologie di DevOps per gestire assets come modelli di ML, algoritmi di Data Science

Nonostante l'MLOps venga descritta come un'estensione del DevOps, **ci sono molte differenze**.

Quando una pipeline di DevOps viene lanciata? Quando una di MLOps?

- Nelle pipeline di MLOps bisogna monitorare **modello e dati**

OSW2021 OPENSOURCEWEEK

Le tecniche e tecnologie di MLOps **sono necessarie** e sono in continua evoluzione:

l'**87%** dei progetti di Machine Learning non raggiunge la produzione (fonte: venturebeat.com)

Solo il **22%** delle aziende che utilizzano il machine learning sono riusciti a mettere con successo un modello online (fonte: algorithmia.com)

OSW2021
OPENSOURCEWEEK


# MLOps → Prima (e unica per molti) fase

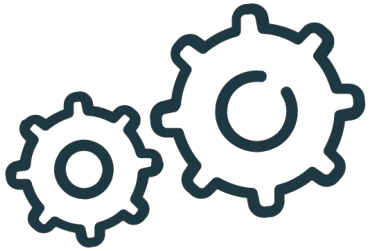
# Addestramento e test del modello

RETE ITALIANA OPEN SOURCE

## DATI

- Data Cleaning
- Data Exploration
- Data Preprocessing
- Labelling
- Feature Extraction
- Train/Dev/Test Splitting

## TRAIN



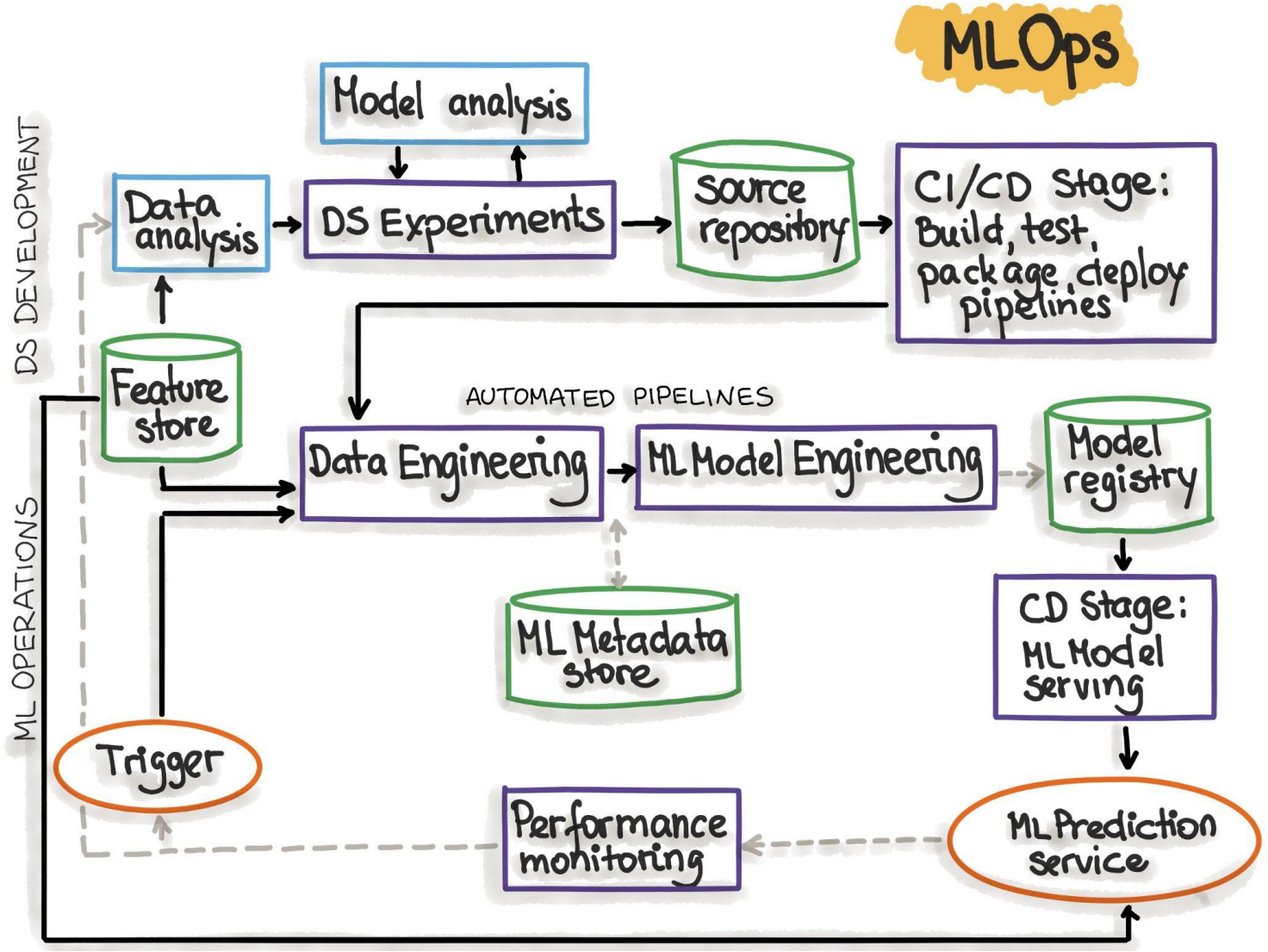
## TEST

- Training
- Diversi Modelli
- Hyperparameter Tuning
- Model Evaluation
- Model Selection (Single Evaluation Metric)

OSW2021 OPENSOURCEWEEK


OSW2021
OPENSOURCEWEEK

# Arriviamo in produzione




fonte: ml-ops.org

OSW2021
OPENSOURCEWEEK

# RIPETIAMO: una volta che sono "contento" e ho un buon modello, non è finita!

RIOS
RETE ITALIANA
OPEN SOURCE

OSW2021
OPENSOURCEWEEK

# Percorriamo la pipeline completa: I dati sono l'inizio di tutto

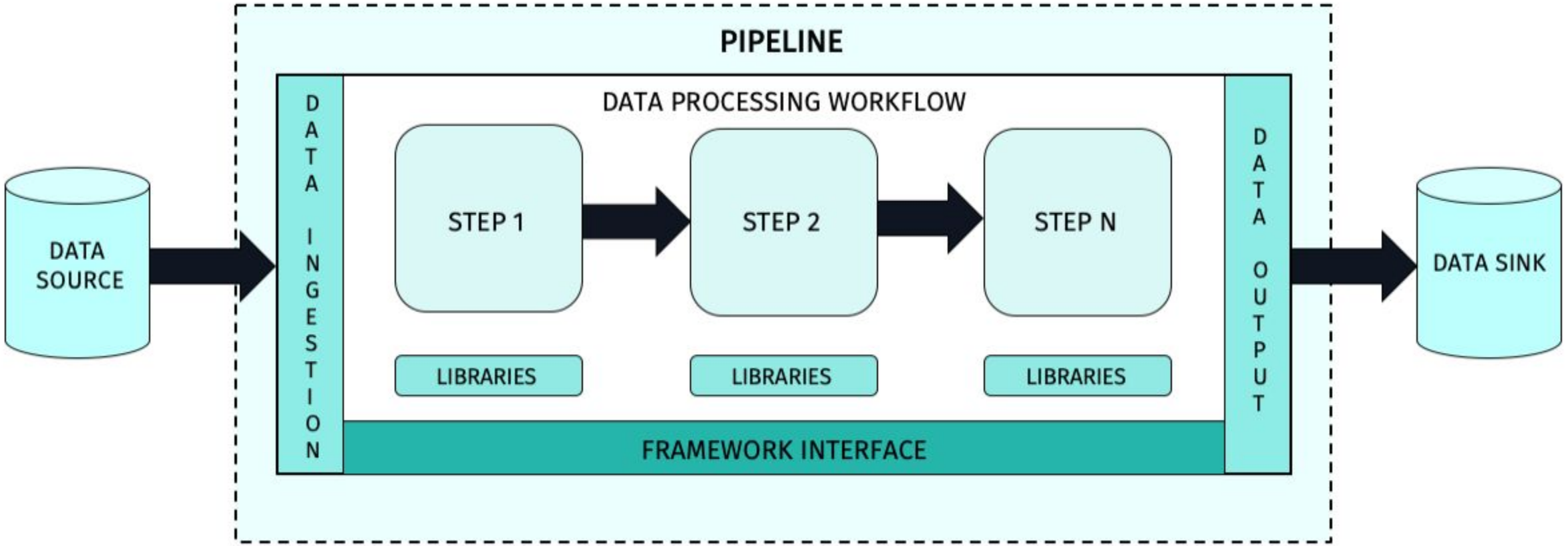
RIOS
RETE ITALIANA
OPEN SOURCE

# Data-Centric AI

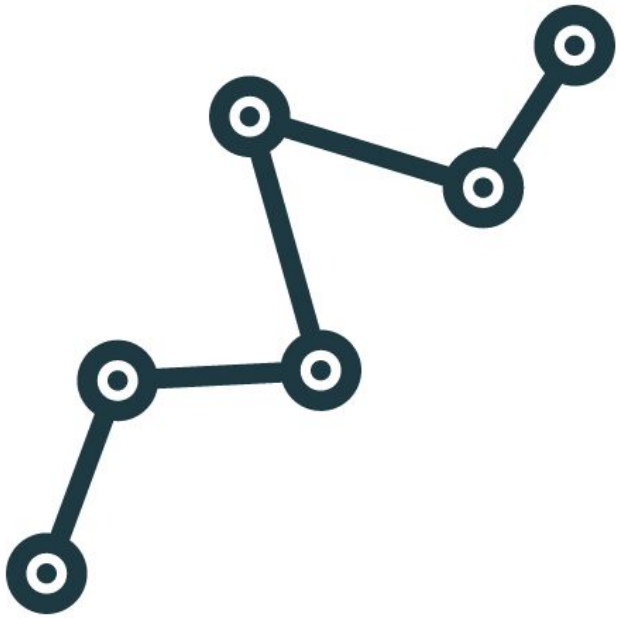
Spostiamo il focus dal modello ai dati (si ottengono risultati più velocemente!):

1. Decidere le priorità e le metriche per AI team e Business
2. Arrivare ad una baseline il prima possibile
3. Fare error analysis e capire quali errori sono collegati ai dati
4. Data Augmentation
5. Feature Extraction/Selection
6. Label Consistency
7. Data pipeline: Data provenance and Data Lineage

# Data Pipeline



# Data Provenance e Data Lineage



**Monitorare** e **tracciare** il percorso dei **dati** è importante per individuare possibili problemi nella nostra data pipeline

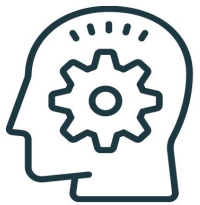
fonte: ml-ops.org

# Data Labelling

RIOS RETE ITALIANA OPEN SOURCE

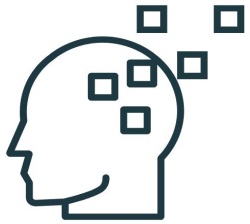
**Definire chiaramente il valore delle label** (in contesti audio, video, sentiment analysis potrebbe non essere semplice)

## SME



Subject Matter Experts ci danno label di alta qualità

## NOT SME



Non SME, cioè persone non esperte, ma che possono essere “trovate” a basso costo: avere più label per ogni esempio

## ADVANCED LABELLING

- Active Learning
- Weak Supervision
- Semi-supervised Learning

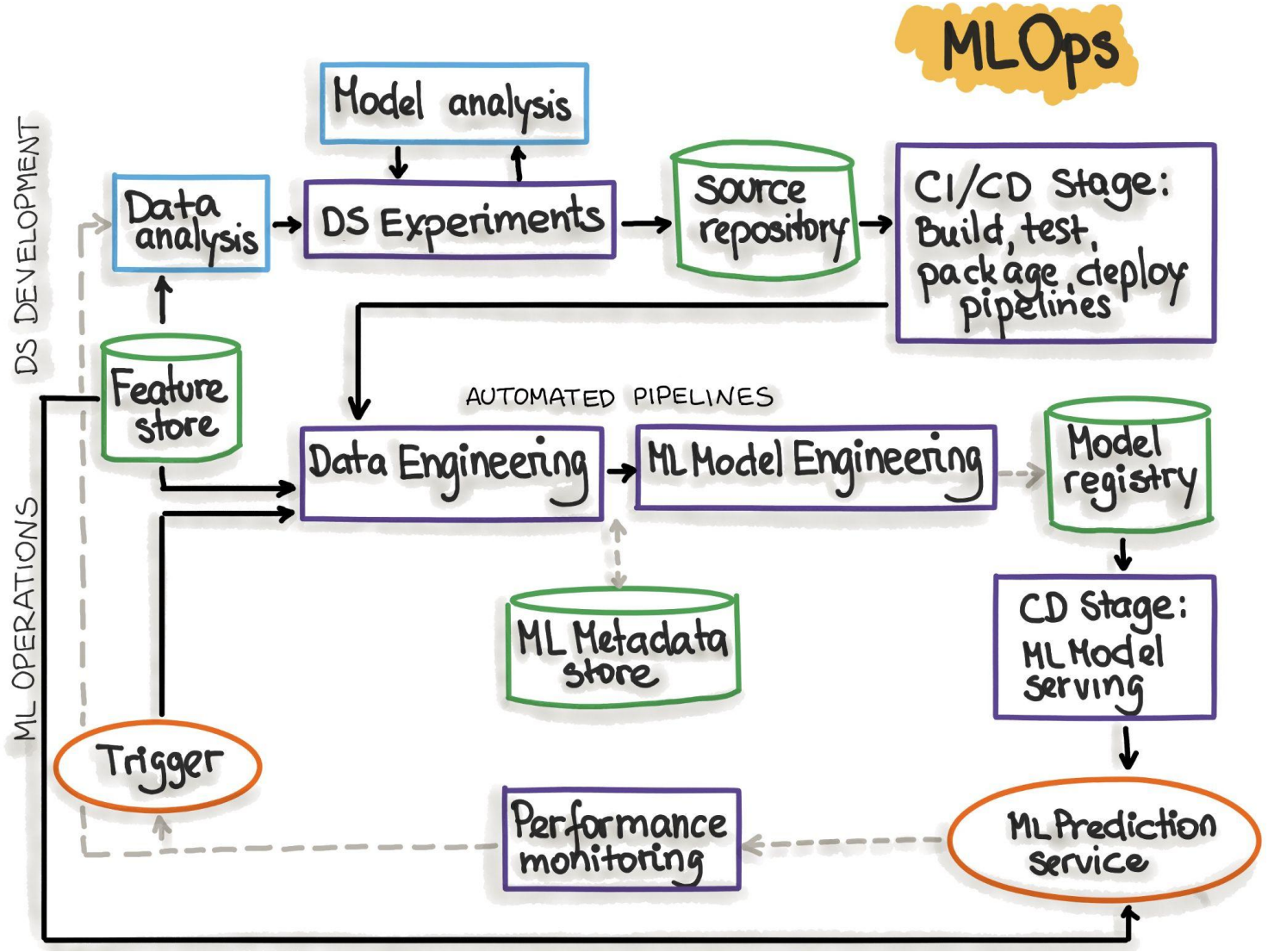
OSW2021 OPENSOURCEWEEK

# Concept and Data drift

**Concept drift** è quando il significato della variabile target cambia. Per esempio cambia il concetto di cosa è fraudolento o meno

**Data drift** è quando le proprietà delle variabili indipendenti (input) cambiano (seasonality, covid, ...)

# Arriviamo in produzione




fonte: ml-ops.org

# ML Metadata

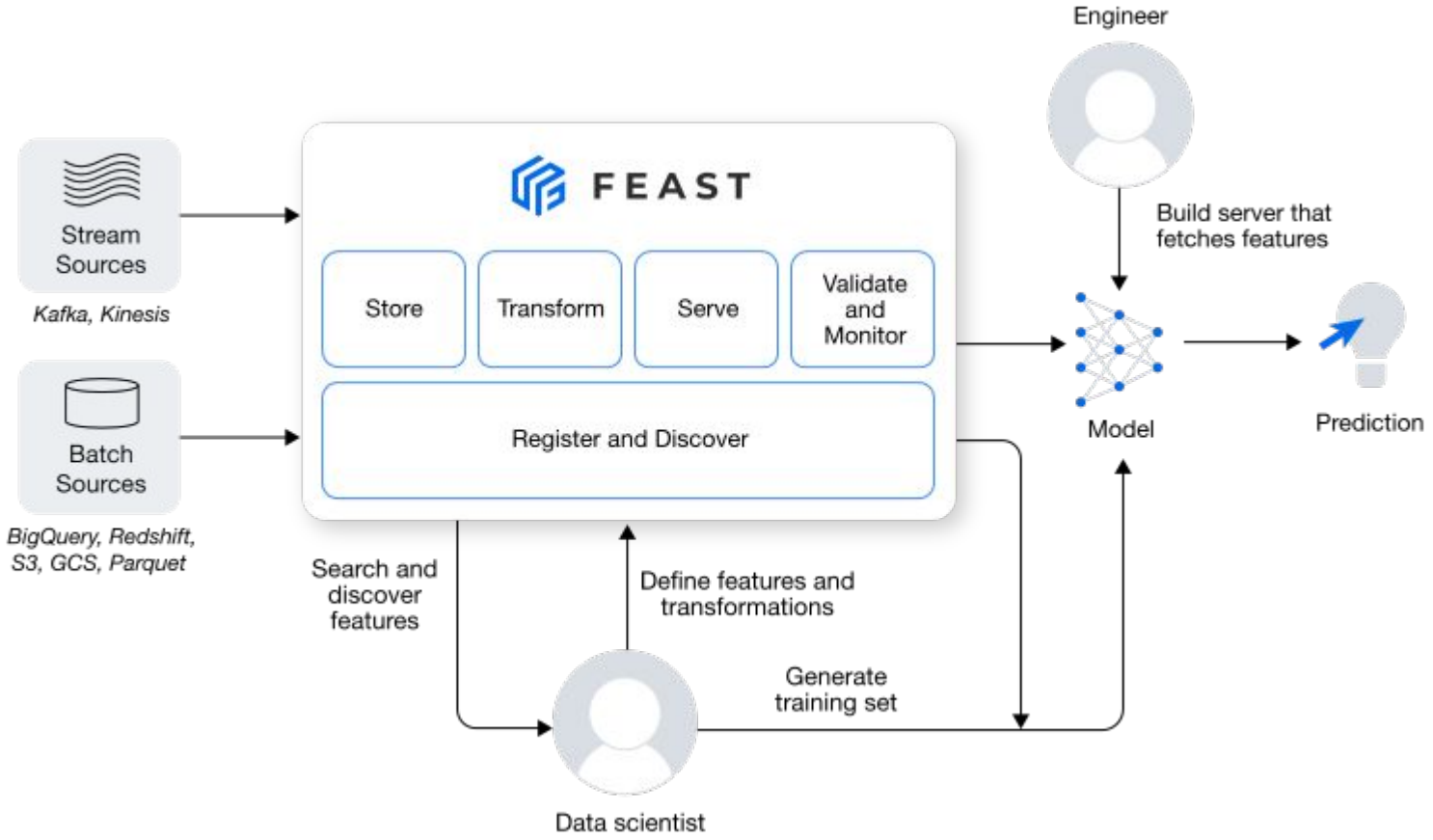
Durante il processo di addestramento, testing, messa in produzione dei modelli, durante quasi ogni step della pipeline vengono generati metadata.

Questi sono importanti per gestire il flusso e per scambiarsi i dati da una parte del flusso ad un’altra.

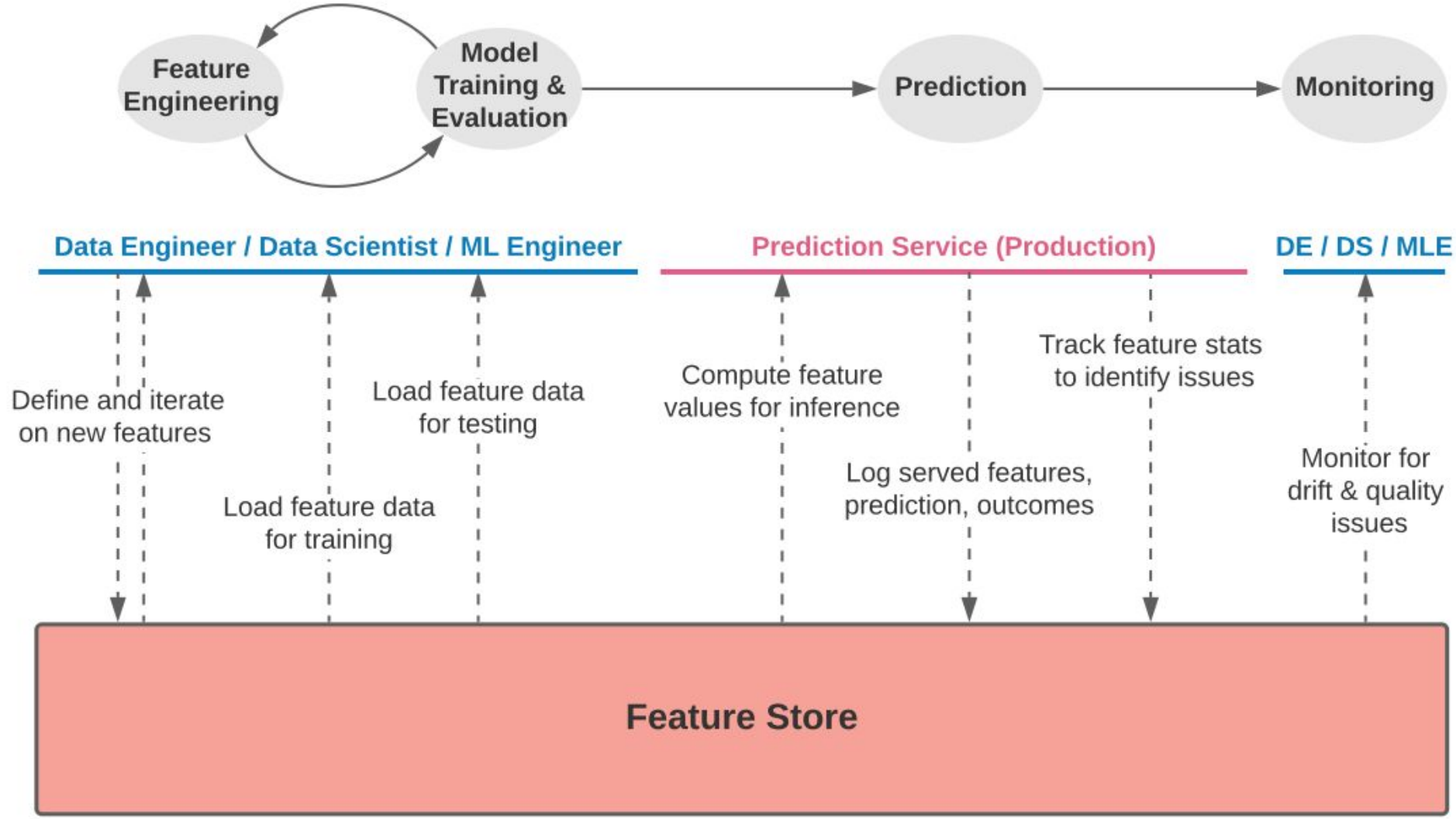
Tensorflow mette a disposizione ML Metadata Store che si collega a diversi database, ci sono diversi servizi che offrono questo tipo di soluzioni

# Feature Store

Amazon SageMaker Feature Store, DataBricks Feature Store, Feast, tecton (fsaas)…
Solitamente hanno internamente 2 tipi di DB (row-oriented e column-oriented)



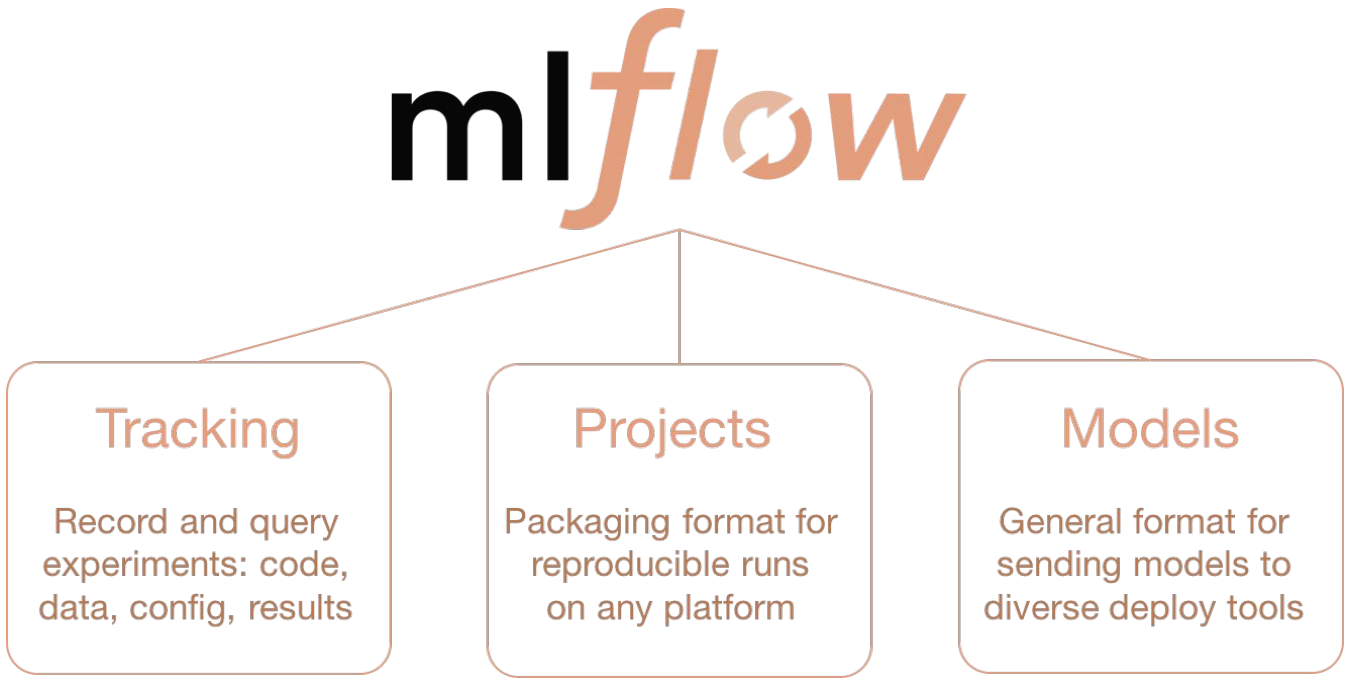
# Feature Store



# Model Registry / Experiment Tracking

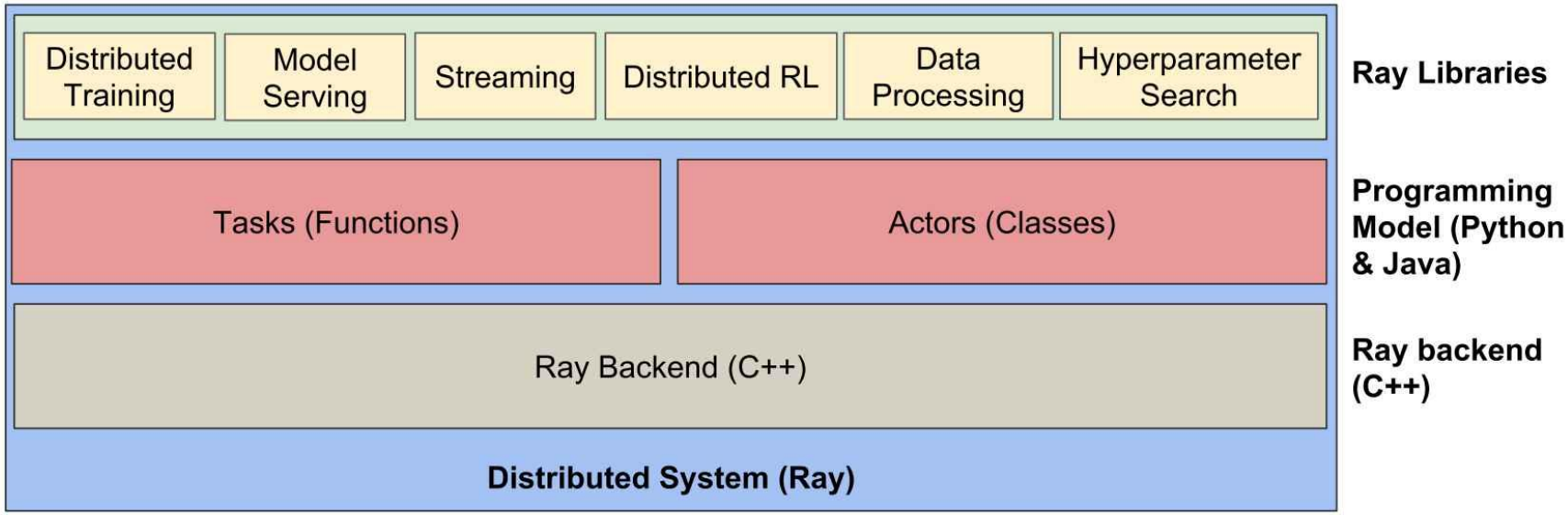
RETE ITALIANA OPEN SOURCE

Aiutano ad organizzare modelli e esperimenti, facilitando la possibilità di avere modelli riproducibili (mlflow, WandB, Neptune.ai, ...)

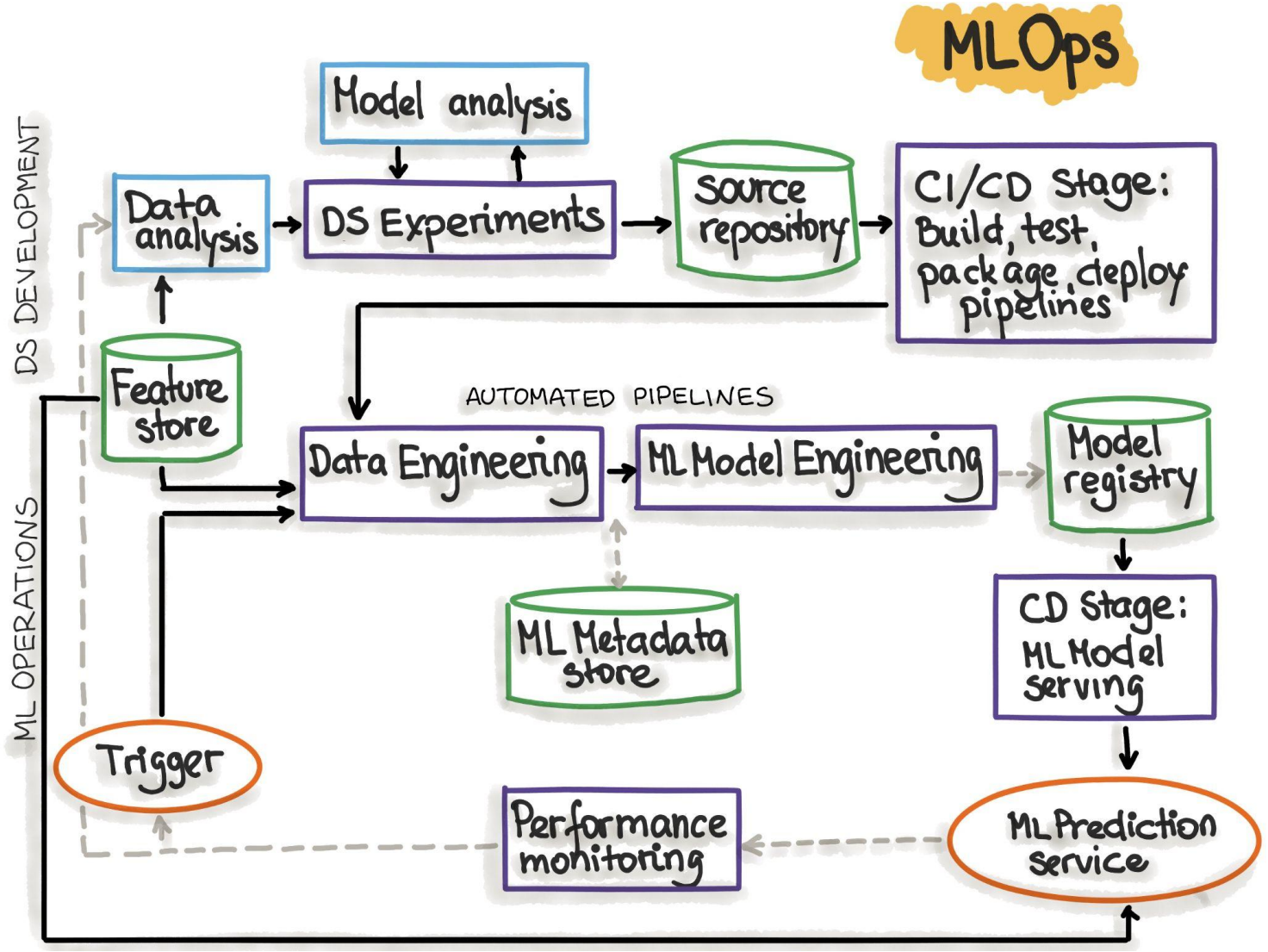



OSW2021 OPENSOURCEWEEK

# What is Ray?


Distributed Training
Model Serving
Streaming
Distributed RL
Data Processing
Hyperparameter Search
Ray Libraries
Tasks (Functions)
Actors (Classes)
Programming Model (Python & Java)
Ray Backend (C++)
Ray backend (C++)
Distributed System (Ray)

# Arriviamo in produzione




fonte: ml-ops.org

# Model Decay

Il mondo cambia continuamente, quindi il nostro modello peggiorerà gradualmente con il tempo.

# Model Deployment Strategies

RIOS RETE ITALIANA OPEN SOURCE

- **Recreate Deployment**: spengo il modello online e metto quello nuovo
  - Downtime
  - Forse sarebbe meglio monitorare il nuovo modello
- **Blue-Green Deployment**: Blue rappresenta il modello corrente e Green il nuovo modello, gradualmente trasferisco traffico dal vecchio modello al nuovo
  - Posso monitorare mano mano che sposto il traffico come si comporta il nuovo modello
  - Devo gestire due environment diversi
- **Canary Deployment**: Distribuire il nuovo modello solo a un subset di utenti o di server
  - Il test del modello avviene direttamente in produzione
  - Backward compatibility e slow rollouts
- **A/B Testing Deployment**, classico A/B Test tra due modelli:
  - Simile a Canary Deployment, ma quando non ho un modello corrente o non ho una versione migliore e la voglio scoprire in produzione con gli utenti (recommendation engine)
- **Shadow Deployment**
  - Il nuovo modello inizialmente non da mai risultati verso gli utenti, ma le predizioni vengono salvate per verificare come si comporta

# Model Remediation e Knowledge Distillation

Altri 2 concetti interessanti a proposito del modello:

- **Model Remediation**: insieme di tecniche per cercare di migliorare il mio modello quando questo lavora male solo su una porzione di dati
  - Migliorare l'input dei dati (anche Augmentation)
  - Agire sul modello (tecniche per migliorare la generalizzazione)
  - Post-processing delle previsioni: meno peso alle predizioni in certe situazioni
  - MinDiff: algoritmo che modificando la loss del modello cerca di far sì che le distribuzioni degli scores siano simili tra porzioni di dato differenti
  -
- **Knowledge Distillation**: distillare la knowledge da un modello più grande ad uno più piccolo
  - I modelli grandi consumano molte risorse e spesso non vengono pienamente utilizzati in produzione
  - Es: cross-entropy tra il modello più grande (y_hat) e il modello più piccolo (y). t è un parametro

$$E(\mathbf{x}|t) = -\sum_i \hat{y}_i(\mathbf{x}|t) \log y_i(\mathbf{x}|t).$$

# Alcune librerie/servizi/ … da citare

- **TFX**: offerto da Tensorflow, l'ecosistema Tensforflow per queste cose è davvero interessante, permette di creare Beam delle data pipeline scritte e eseguirle anche su Spark. Ha alcuni paletti nella scrittura e gestione del codice
- **I task orchestrators possono esserci utili** (Airflow, Kubeflow, Luigi, …)
- Ci sono diversi servizi cloud e è difficile fare una comparativa di quelli che offrono l'uno o l'altro

# Conclusioni



- Avere processi strutturati e tracciare gli esperimenti permette di non ribattere strade già battute e condividere facilmente i risultati
- Con processi e tecnologie di MLOps è più facile identificare problemi del modello e dei dati
- Le cose cambiano, non è pensabile che un modello messo in produzione funzionerà per sempre
- Gestire i dati è una parte fondamentale del processo, ricordiamo di andare in una direzione data-centric e non model-centric
- è importante definire scope e metriche di progetto sia con team di tecnici che business per far si di raggiungere il prima possibile i risultati





Grazie




**Filippo Minutella**
AI Engineer, Larus Business Automation