Machine Engineering Principles with Python and MLFLow

Intro

Natu Lauchande

About Me :

Principal Data Engineer @Data & Prediction team @ Jumo.World

What we will cover and not cover in this talk ?

In Scope:

1. ML Engineering **pain points**

2. Sample of relevant ML Pipeline **architectures** used in the on top tech companies

3. ML Engineering **principles** to address **some** of the pain points in the ML development process

4. **MLOps concepts and intro** and technical debt prevention/mitigation techniques

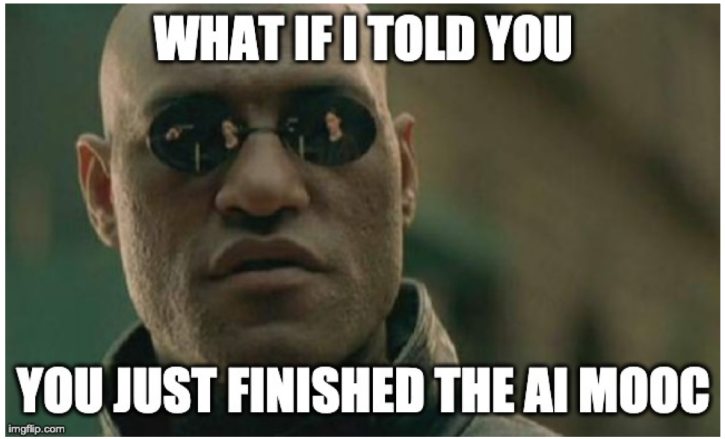
Out of Scope:

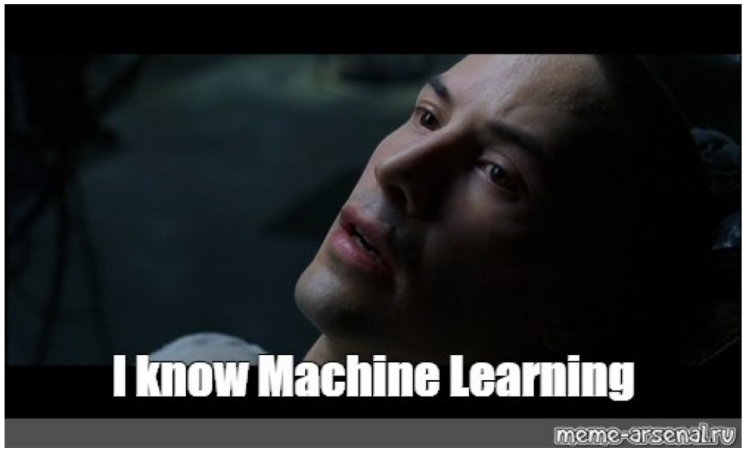
1. **Detailed review** of ML systems

2. Any specific **data science or algorithmic** solution

4. Tutorial on using any specific tool (eg: **MLFlow, etc.)**

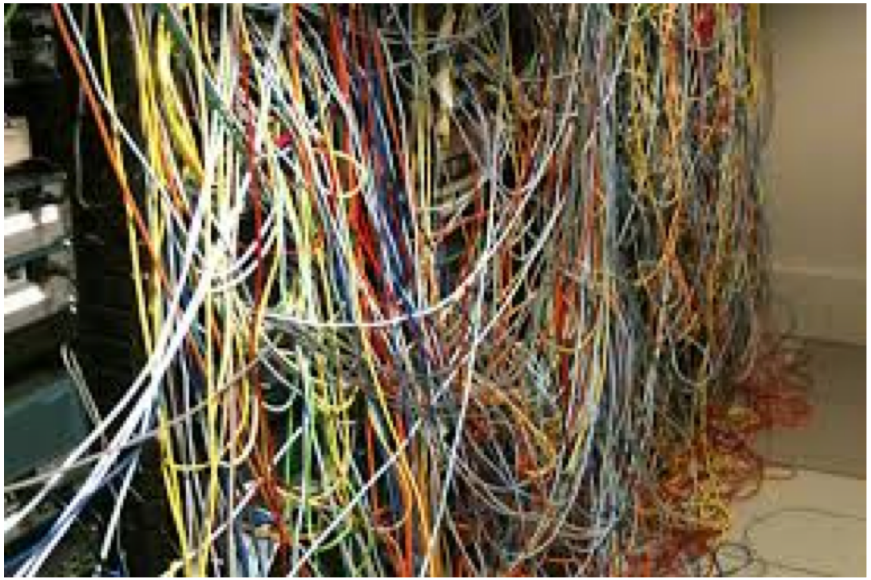
Machine Learning easily accessible





Pain points - ML systems don’t work the same in dev vs in prod

Pain points - Data Pipeline Entanglement

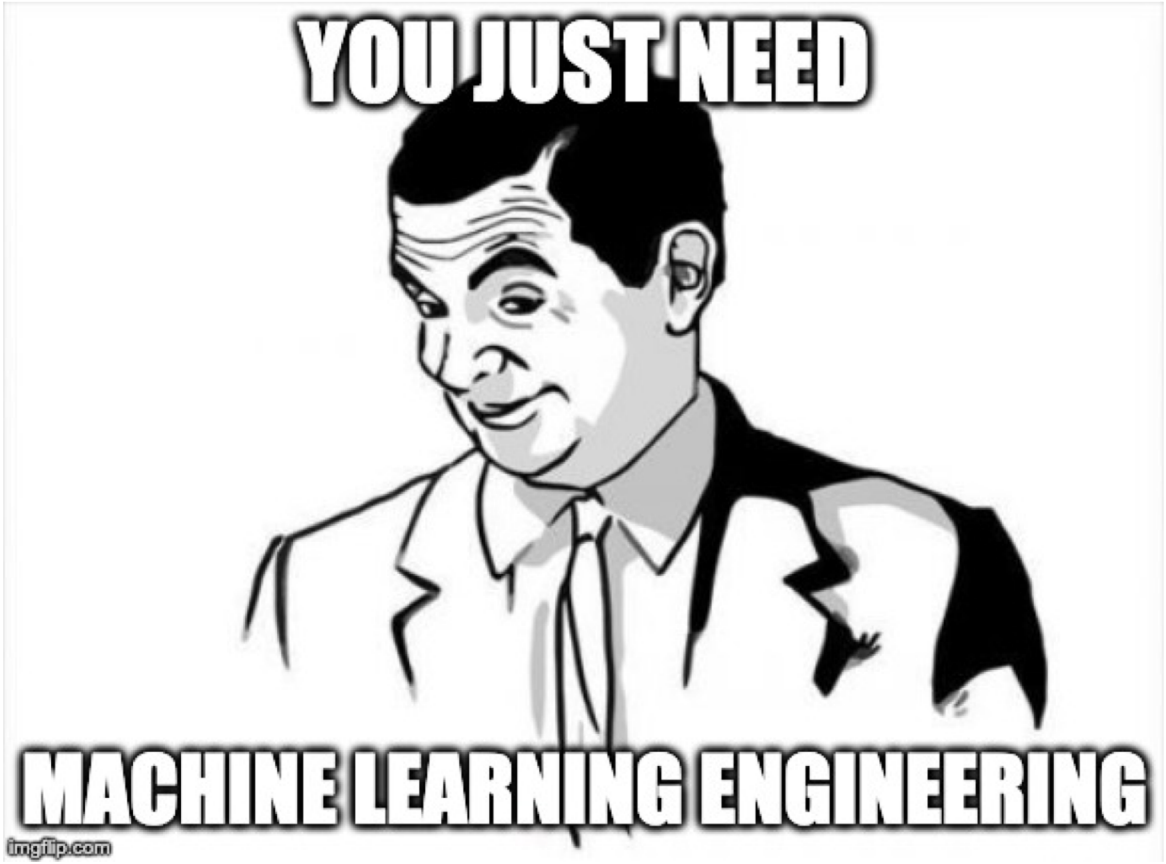


Pain Points - Experimentation Management

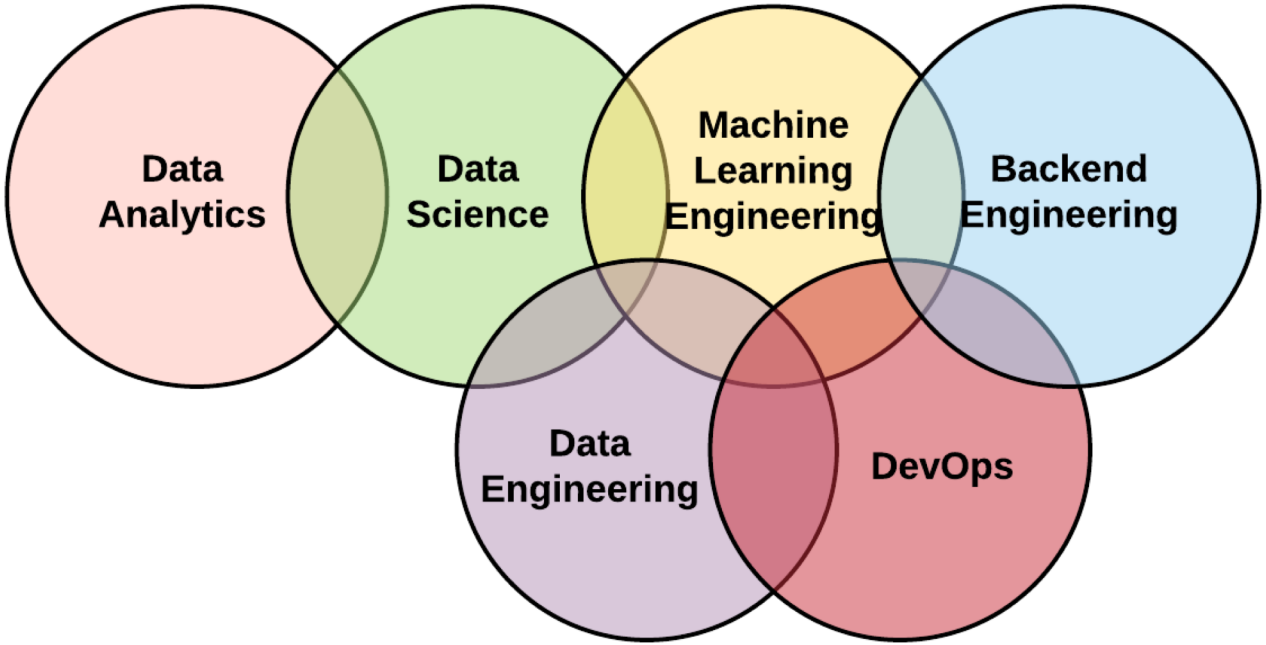
Machine Learning Engineering Pain - Points Metrics

Machine Learning Engineering Pain Points

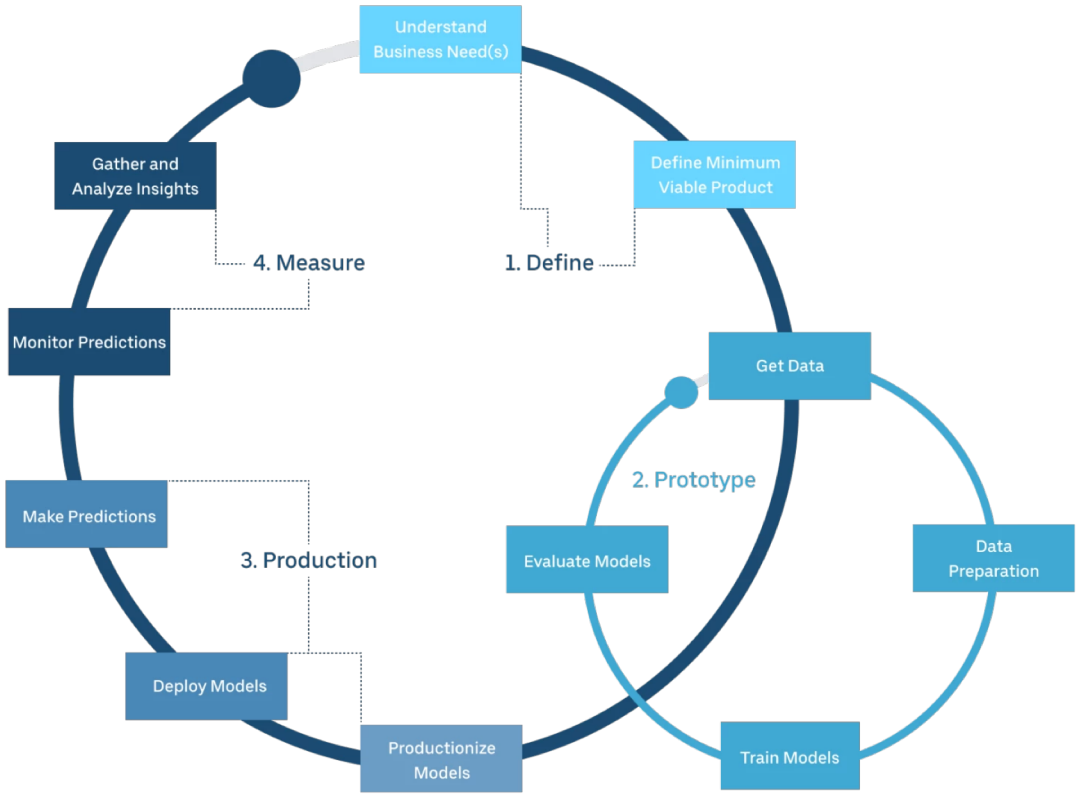
We have a solution MEME



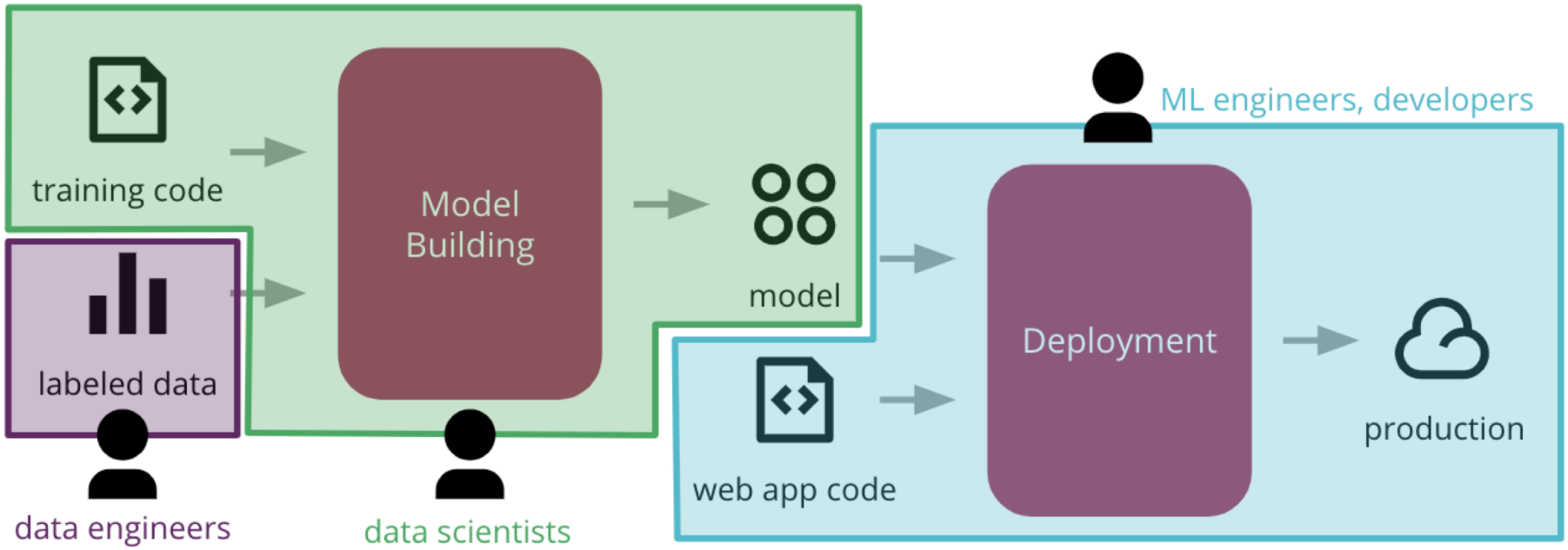
ML Engineering



Idealized Workflow of a Machine Learning Project



ML Engineering



ML code in the Context of ML backed System

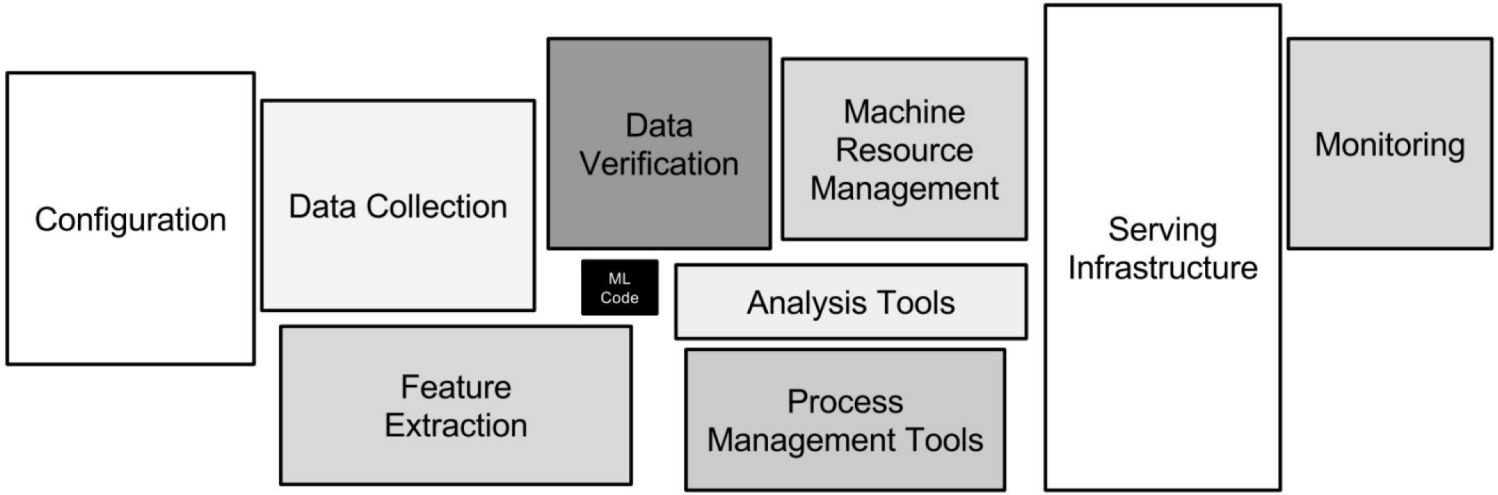
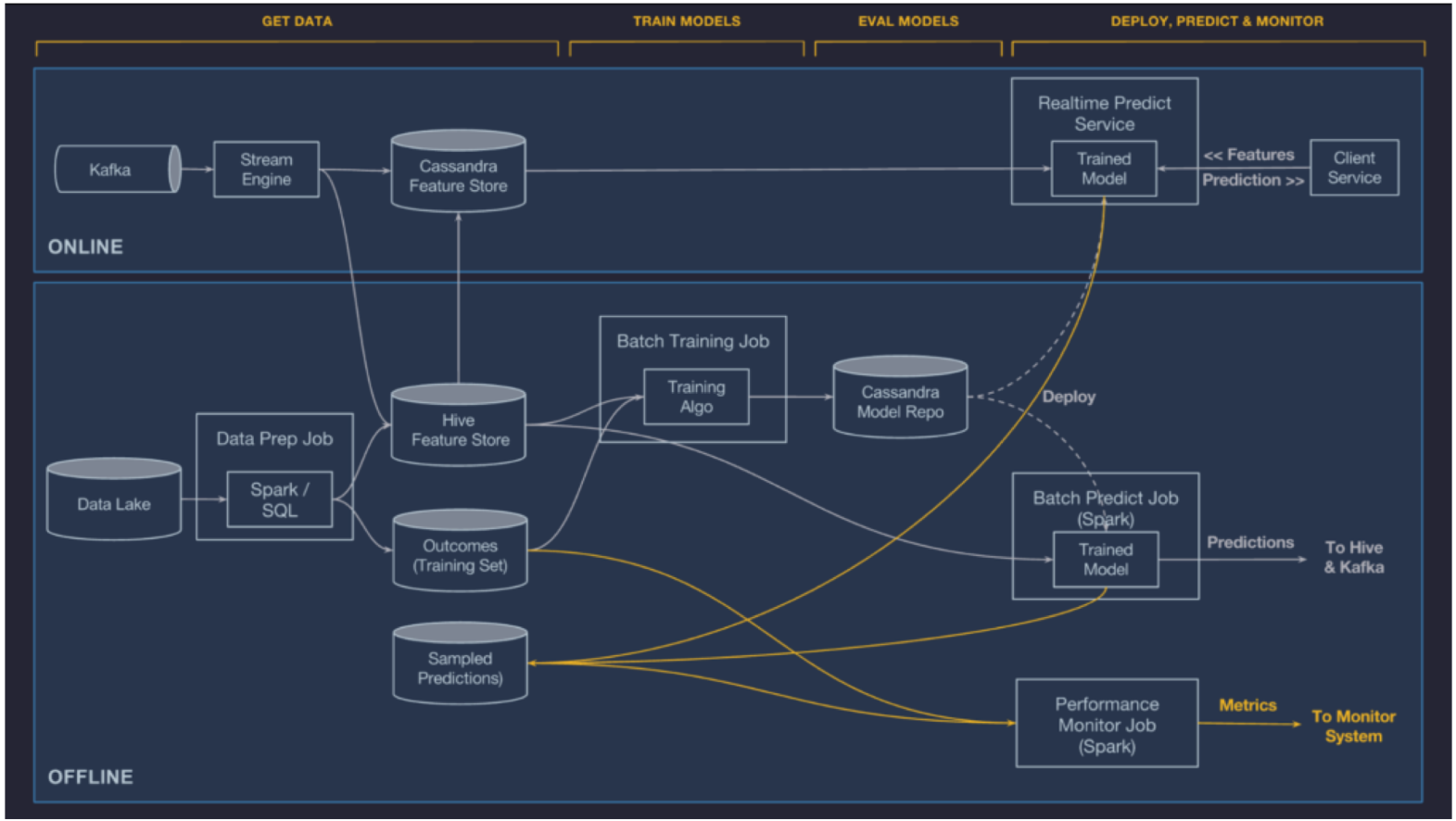


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Machine Learning Engineering Principles (Some…)

- **Software engineering approach, platform view of your ML functioning - CD4ML**
- **Cross-functional teams w/ a somewhat defined Lifecycle process**
- **Integrated data Lifecycle**
- **Producing software based on code, data, and models Small and safe increments**
- **Reproducible and reliable software release Software release at any time**
- **ML Systems Test Scoring**

Michelangelo - Overview



Michelangelo - Main Principles

Data Management:

Operations:

Tech Stack:

FB Learner Overview

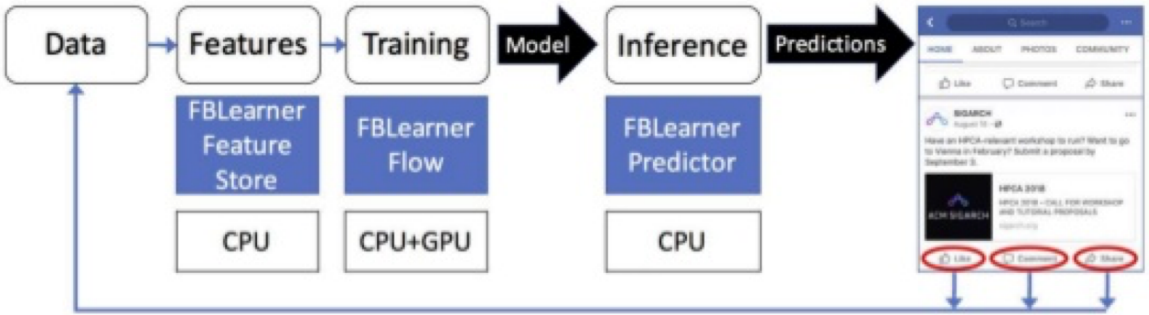


Fig. 1. Example of Facebook's Machine Learning Flow and Infrastructure.

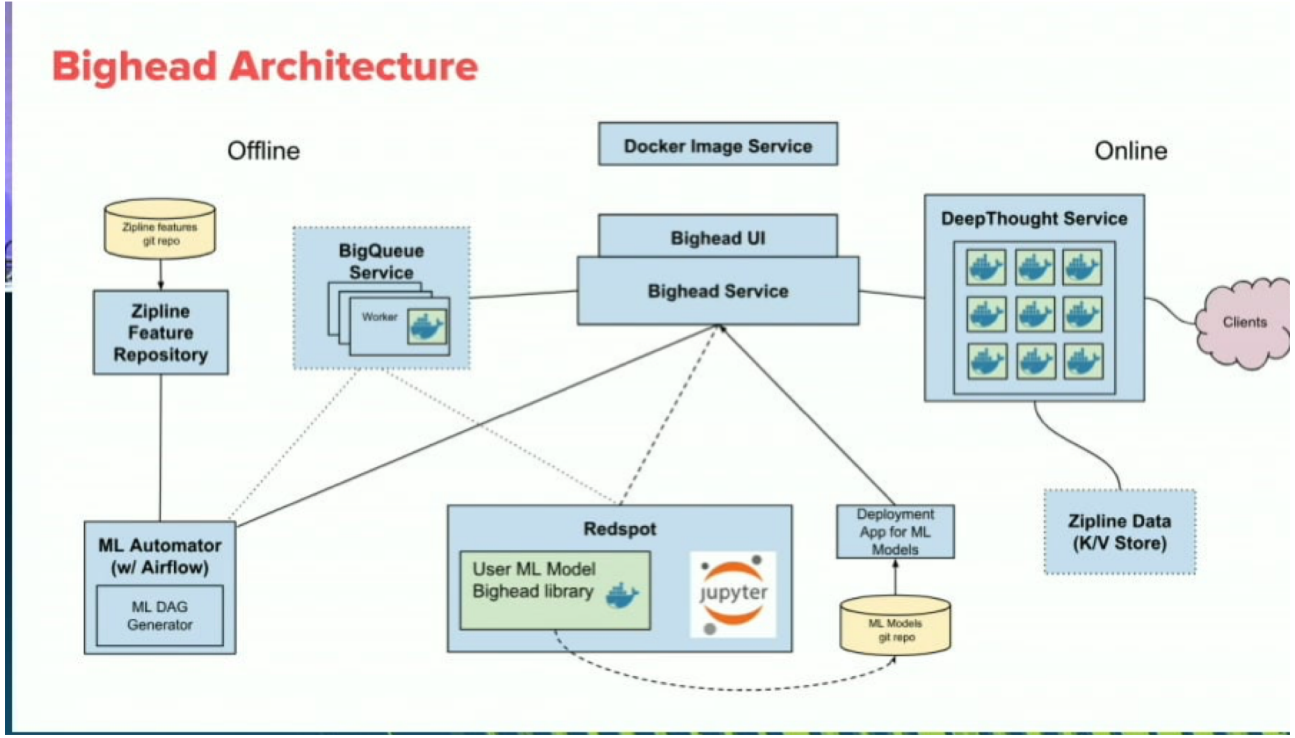
Michelangelo - Main Principles

Data Management:

Operations:

Tech Stack:

Big Head Overview



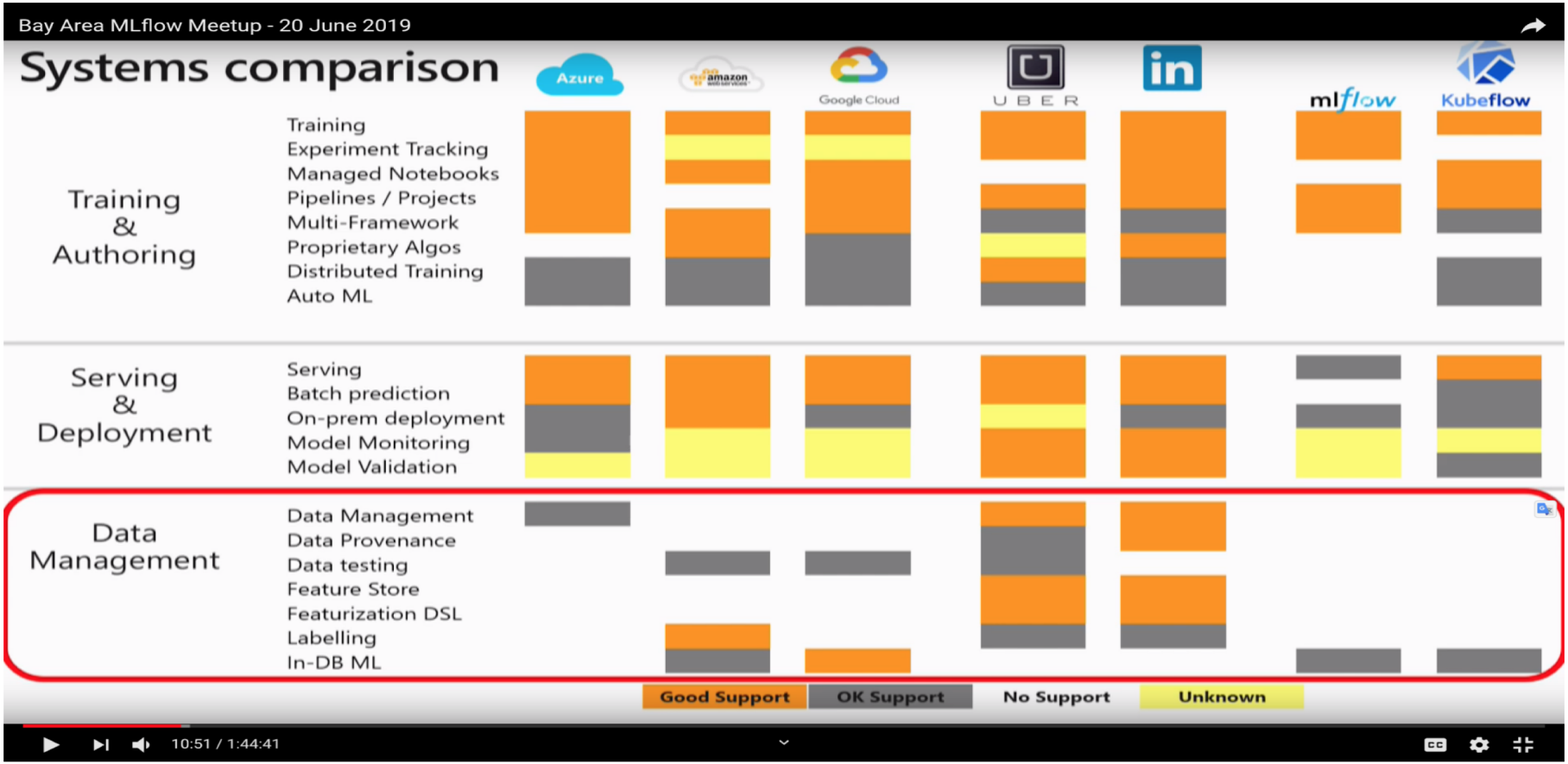
Michelangelo - Main Principles

Data Management:

Operations:

Tech Stack:

Machine Learning Comparative View



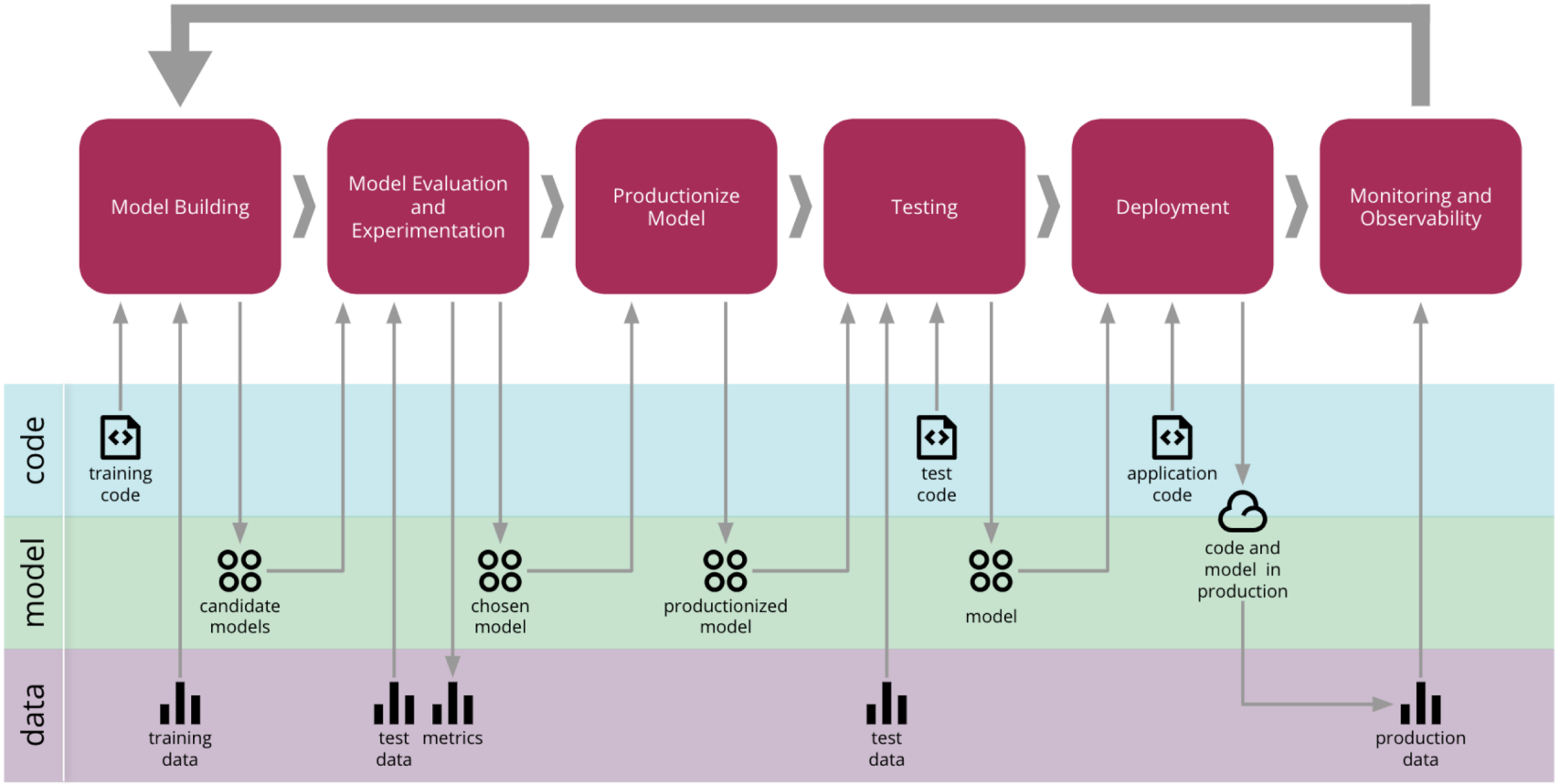
BUILD YOU OWN COOL

Data Management:

Operations:

Tech Stack:

Ideal Pipeline - Operations (CD4ML)



Ideal Pipeline - Real Time Monitoring Perspective (PICTURE NOT CLEAR COMPLICATED - VISUAL AID - Test Score, https://www.eecs.tufts.edu/~dsculley/papers/ml_test_score.pdf)

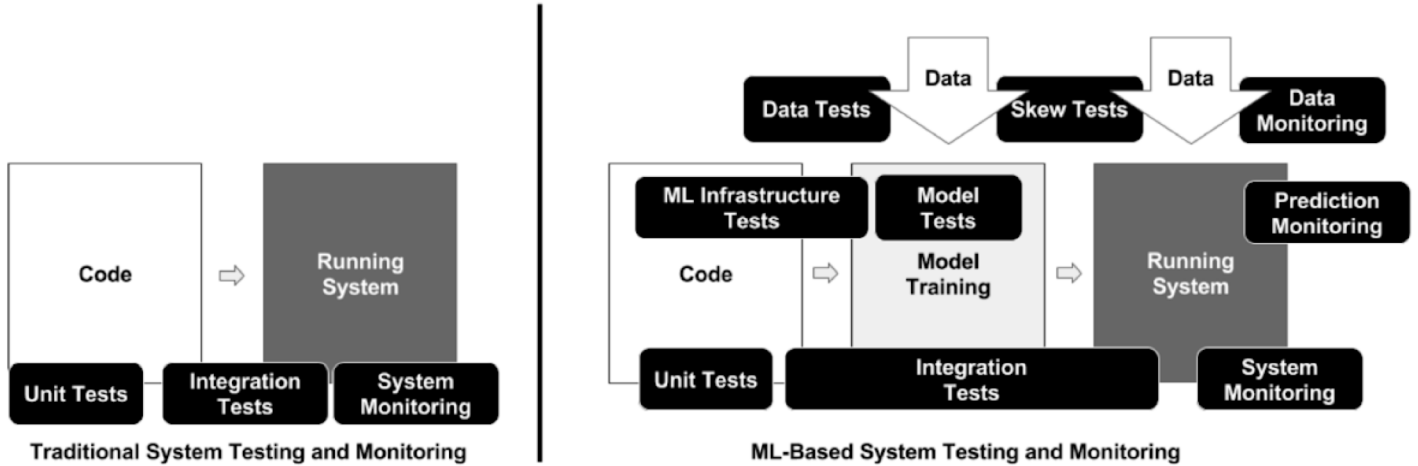
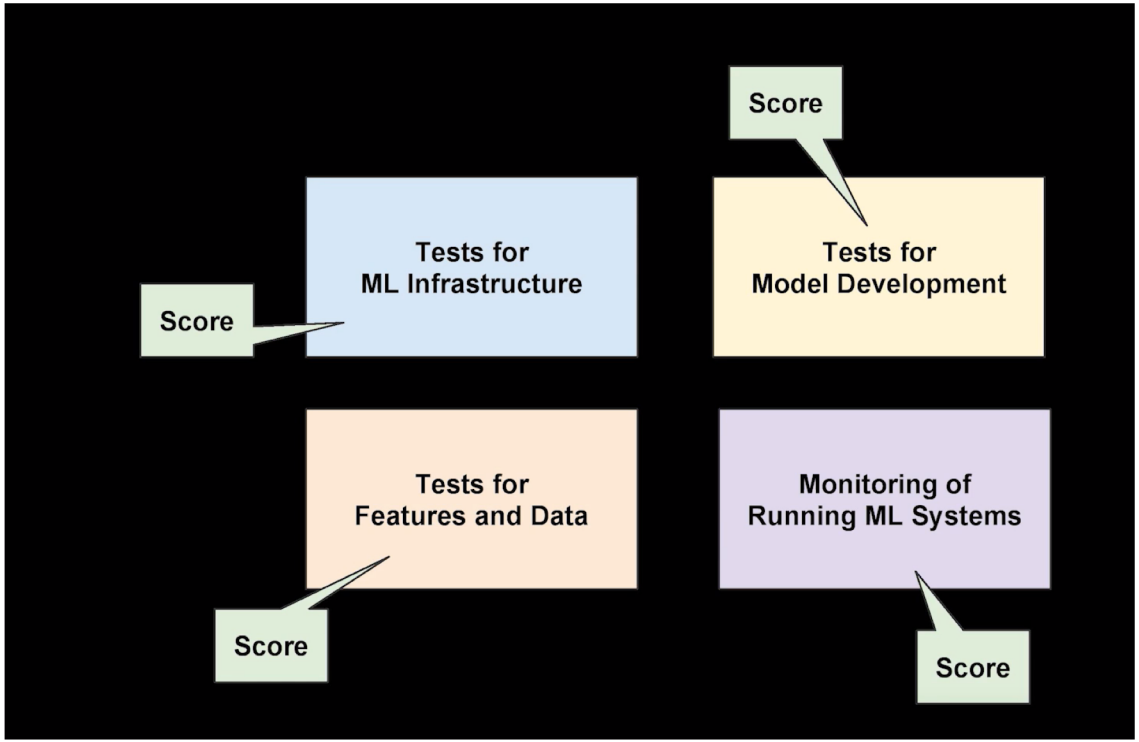


Figure 1. **ML Systems Require Extensive Testing and Monitoring.** The key consideration is that unlike a manually coded system (left), ML-based system behavior is not easily specified in advance. This behavior depends on dynamic qualities of the data, and on various model configuration choices.

Ideal Pipeline - Real Time Monitoring Perspective (D. Sculley - https://www.youtube.com/watch?v=V18AsBIHIWs**)**

ML Score - https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45742.pdf**)**



ML Flow Intro

Open source machine learning platform that addresses the following issues :

1. **Experimentation Management** : UI to manage and log different models

2. **Reproducibility** : Runs he same way anywhere

3. **Standardization of Operations/Serving Layer** : Designed to scale for 1 or 100 000 orgs

Data Layer

Data Versioning :

1. **DVC**

2. **Apache Delta Lake**

Data Quality/Tests:

1. **Deequ**

2. **Great Expectations**

MLflow Components

mlflow Tracking

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

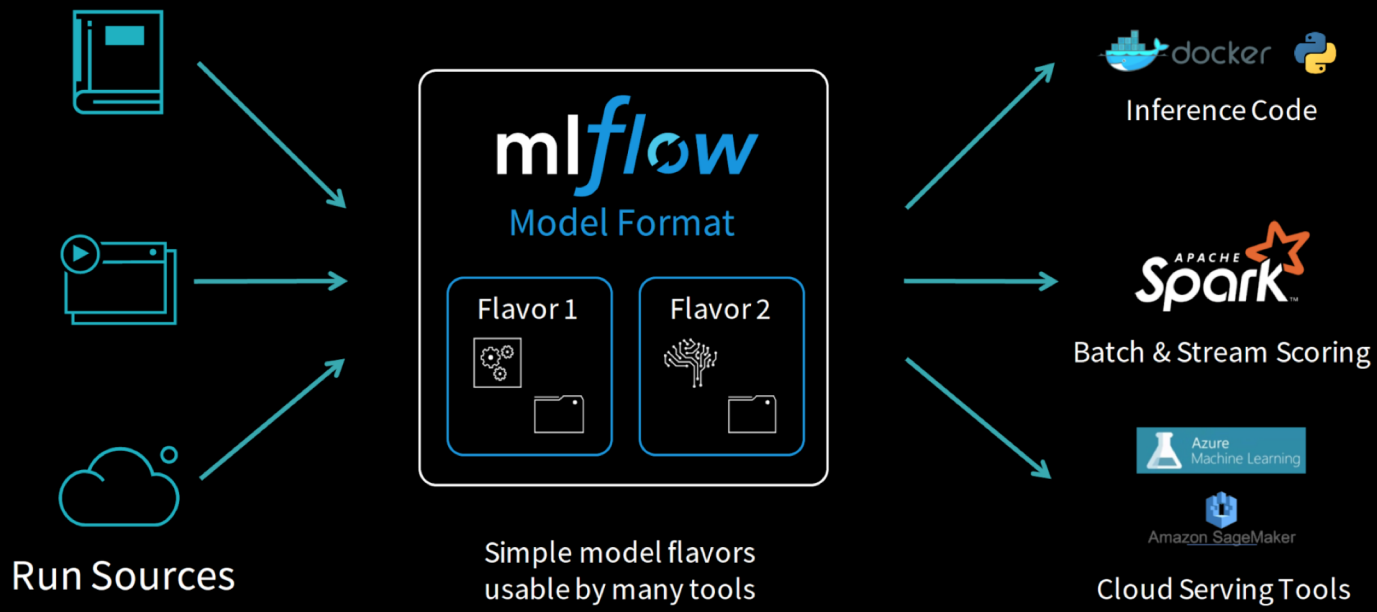
mlflow Projects

Packaging format for reproducible runs on any platform

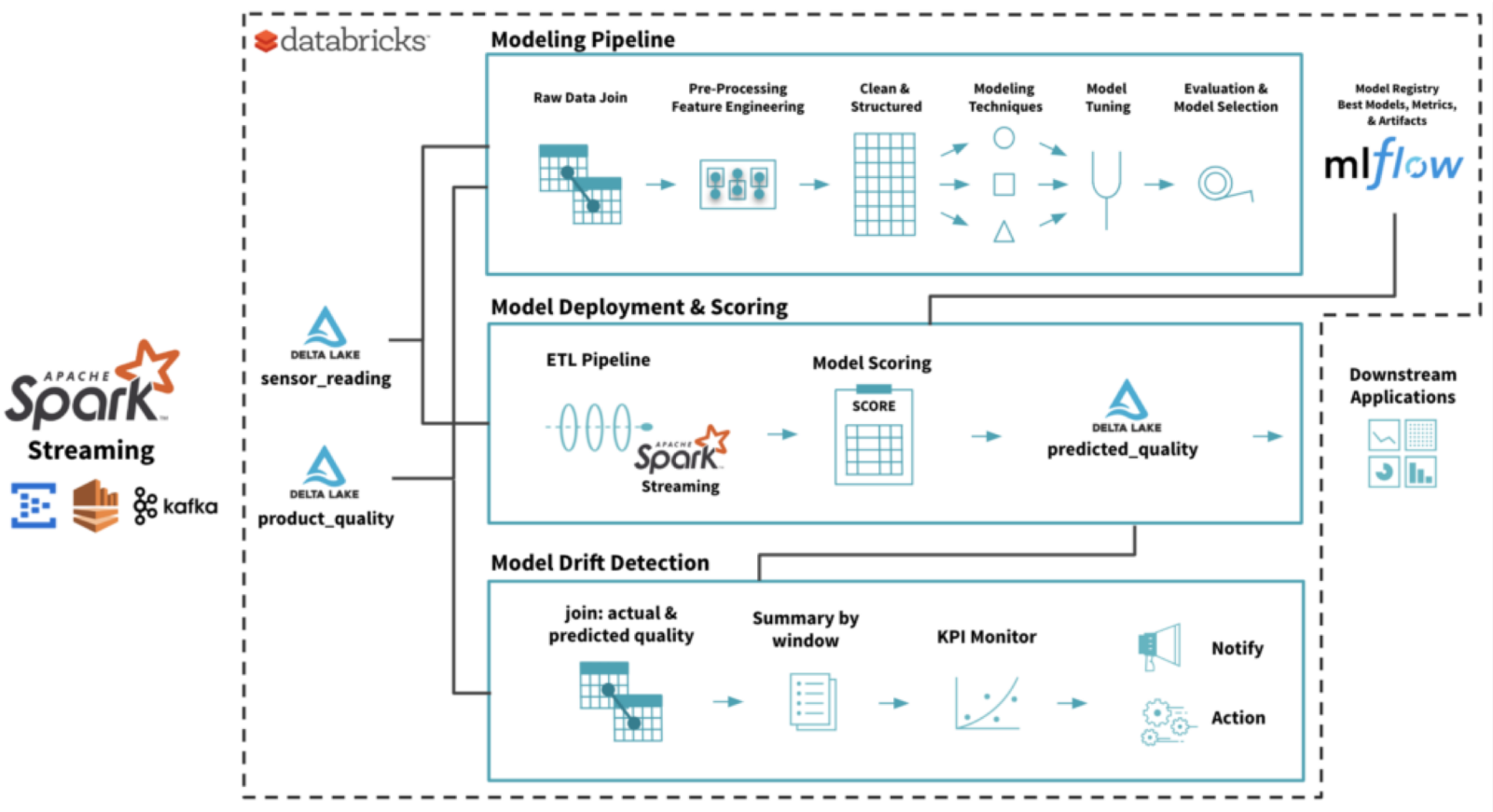
mlflow Models

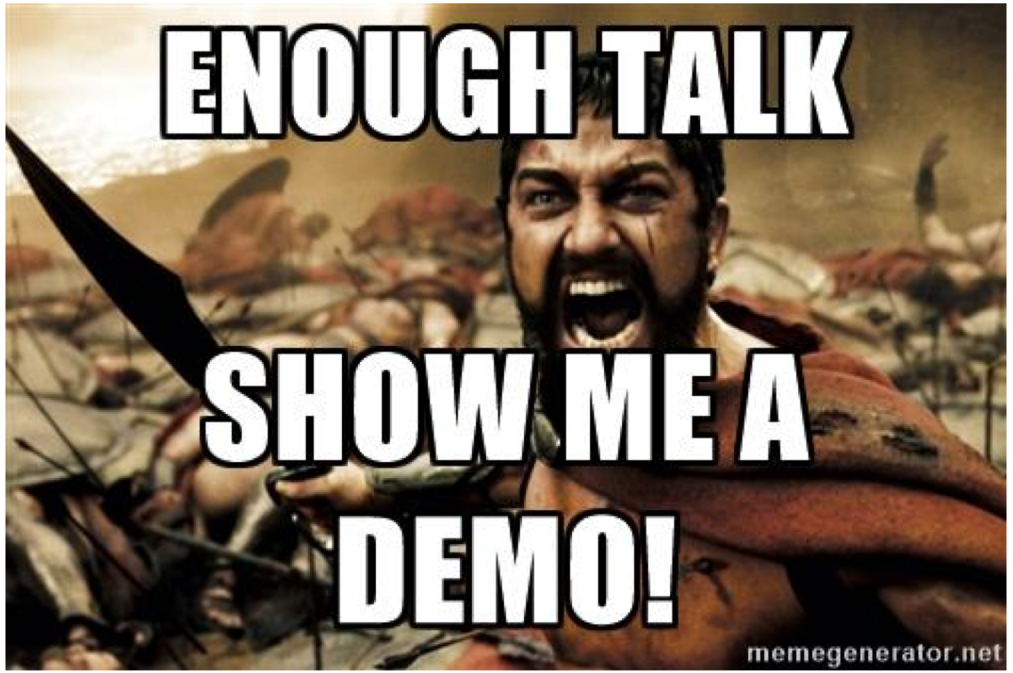
General model format that supports diverse deployment tools

MLflow Models



Machine Learning Principes : Pipeline - Structure Strategy (Databricks - Diagram pipeline)




ENOUGH TALK
SHOW ME A
DEMO!
memegenerator.net

ML Problem - Bitpred - Problem

Problem Statement : Will the bitcoin price go down tomorrow (weird problem)?

Metrics Review Session - MLOps

Have a couple of models
Run MLFlow

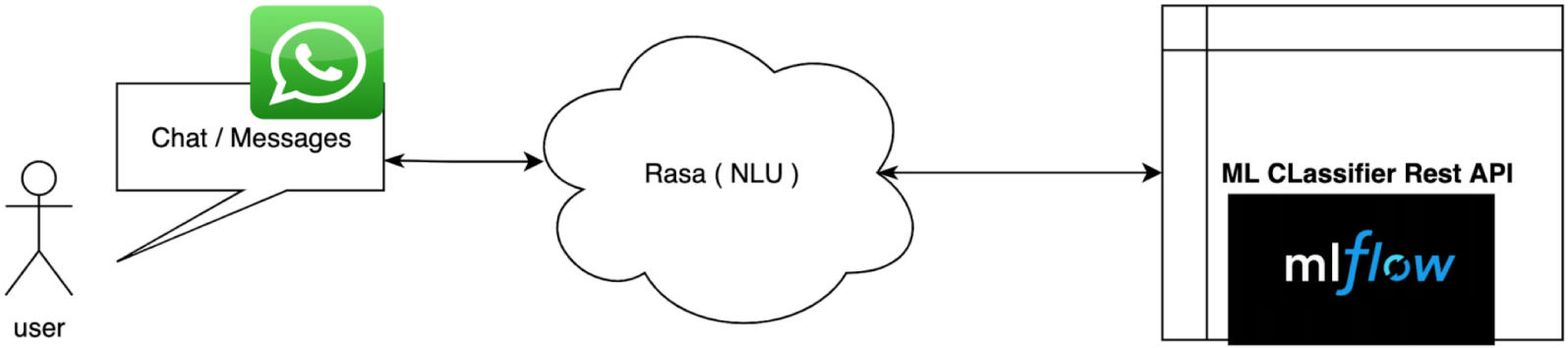
Demo Example I - Bitpred - Data

https://github.com/nlauchande/bitpred

Demo Example I - ML Pipeline for a PTSD Chatbot risk classifier

Therapist: Jill, do you mind if I ask you a few questions about this thought that you noticed, "I should have had them wait and not had them go on?"

Client: Sure.



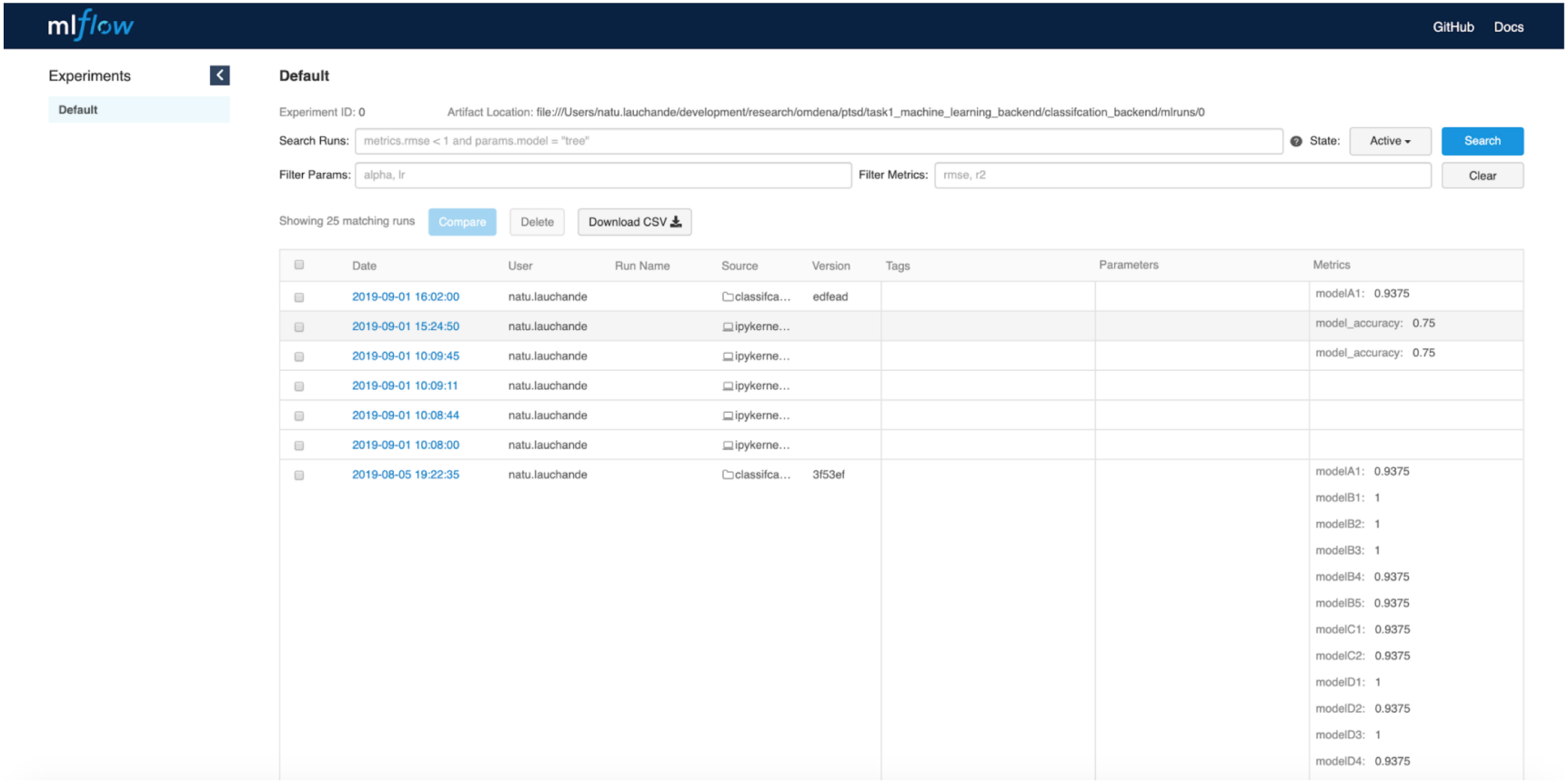
Demo Example - Sample sklearn pipeline

```
train, test = train_test_split(final_dataset, random_state=42, test_size=0.33, shuffle=Tr
X_train = train.text
X_test = test.text

LogReg_pipeline = Pipeline([
            ('tfidf', TfidfVectorizer(sublinear_tf=True, min_df=5, norm='l2', encoding='l
            ('clf',LogisticRegression(solver='sag'),),
        ])
```

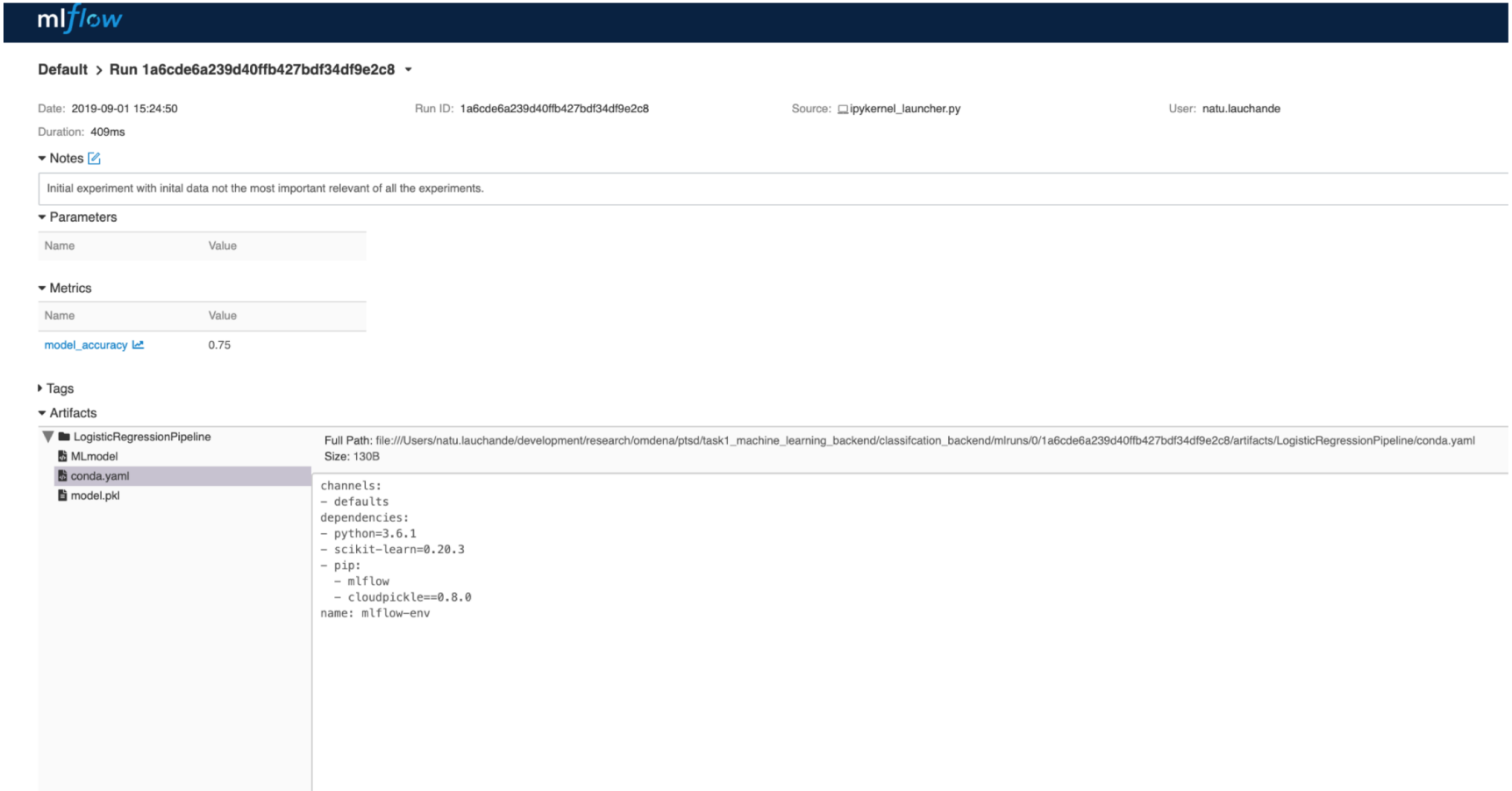
Demo Example - Log your model

```
with mlflow.start_run():
    LogReg_pipeline.fit(X_train, train["label"])

    # compute the testing accuracy
    prediction = LogReg_pipeline.predict(X_test)
    accuracy = accuracy_score(test["label"], prediction)

    mlflow.log_metric("model_accuracy", accuracy)
    mlflow.sklearn.log_model(LogReg_pipeline, "LogisticRegressionPipeline")


```

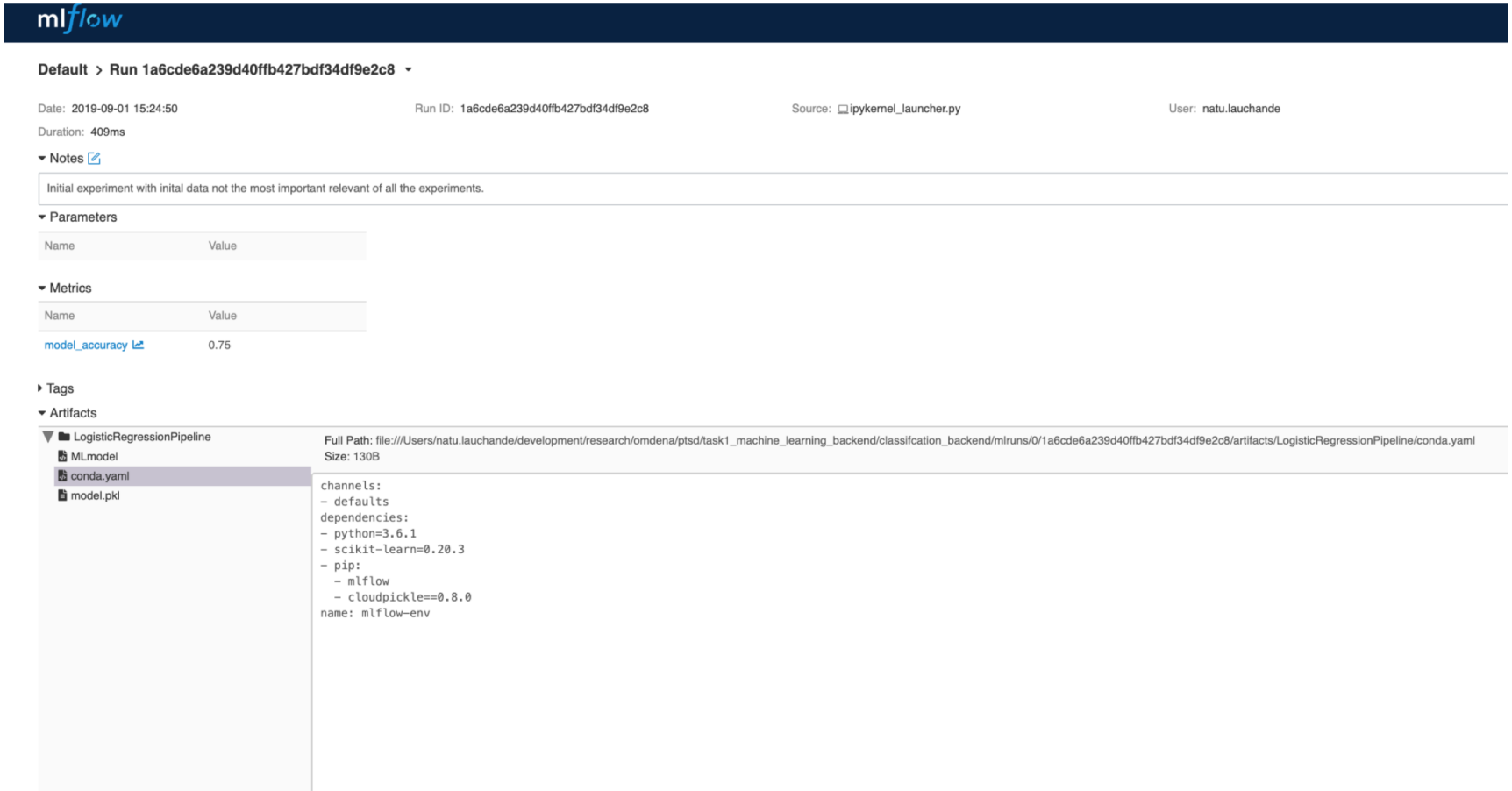
Demo Example - Track Experiments



Demo Example - Model Tracking and Versioning



Demo Example - Model Tracking and Versioning



Demo Example - Serving Layer

```
mlflow models serve -m
runs://0/104dea9ea3d14dd08c9f886f31dd07db/LogisticRegressionPipeline


2019/09/01 18:16:49 INFO mlflow.models.cli: Selected backend for
flavor 'python_function'


2019/09/01 18:16:52 INFO mlflow.pyfunc.backend: === Running command
'source activate mlflow-483ff163345a1c89dcd10599b1396df919493fb2
1>&2 && gunicorn --timeout 60 -b 127.0.0.1:5000 -w 1
mlflow.pyfunc.scoring_server.wsgi:app'


[2019-09-01 18:16:52 +0200] [7460] [INFO] Starting gunicorn 19.9.0


[2019-09-01 18:16:52 +0200] [7460] [INFO] Listening at:
http://127.0.0.1:5000 (7460)
```

Demo Example - Serving Layer

```
curl http://127.0.0.1:5000/invocations -H 'Content-Type:
application/json' -d '{"columns":["text"],"data":[[" concatenated
text of the transcript"]]}'


[0]%
```

Demo Example - ML Package manager

```
name: OmdenaPTSD

conda_env: conda.yaml

entry_points:
  main:
    command: "python train.py"
```

```
name: omdenaptsd-backend
channels:
  - defaults
  - anaconda
dependencies:
  - python==3.6
  - scikit-learn=0.19.1
  - pip:
    - mlflow>=1.1
```

Take aways