

MLFLow

## Machine Engineering Principles with Python and

Intro

---

Natu Lauchande

## **About Me :**

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

What we will cover and not cover in this talk ?

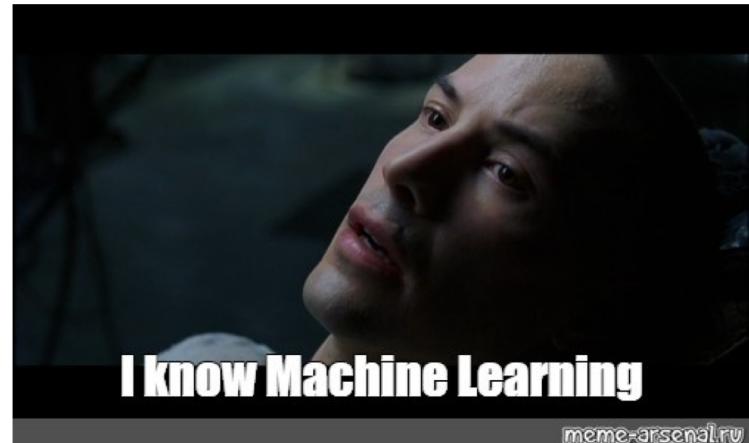
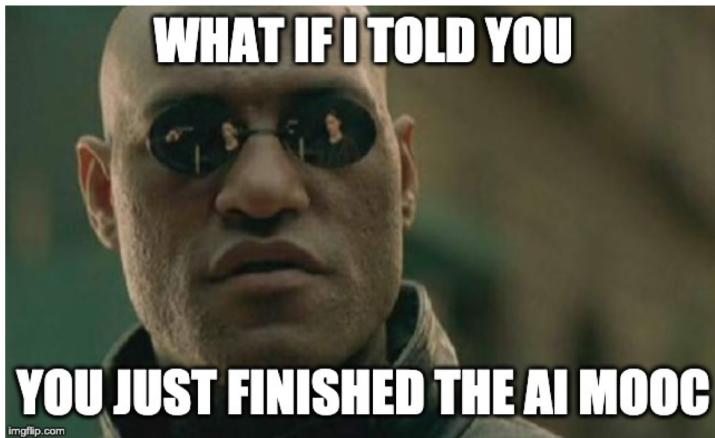
### In Scope:

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.**)

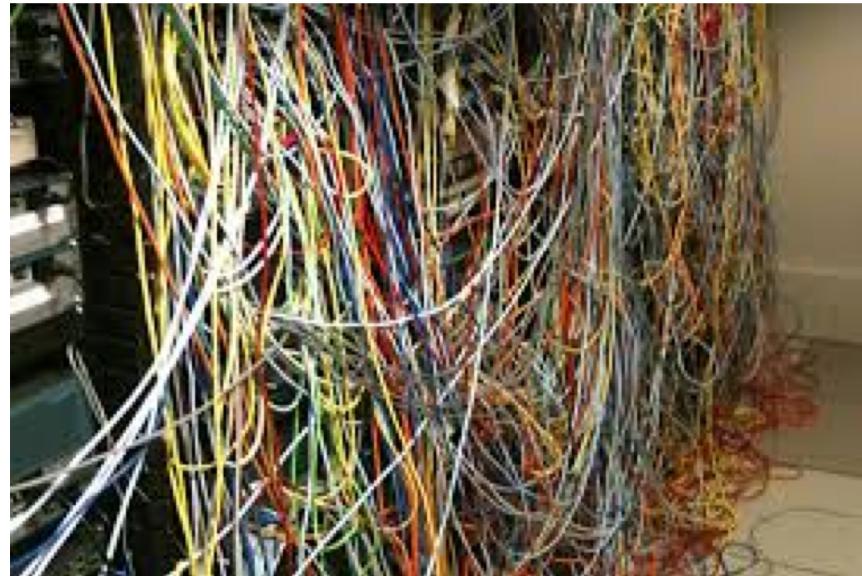
## Machine Learning easily accessible



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



## Pain points - Data Pipeline Entanglement



## Pain Points - Experimentation Management



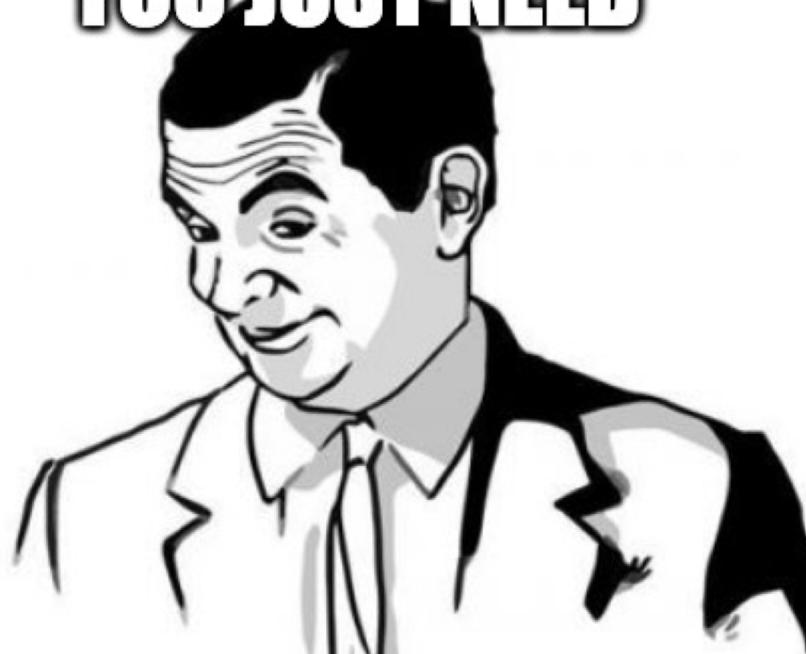
# Machine Learning Engineering Pain - Points Metrics



## Machine Learning Engineering Pain Points

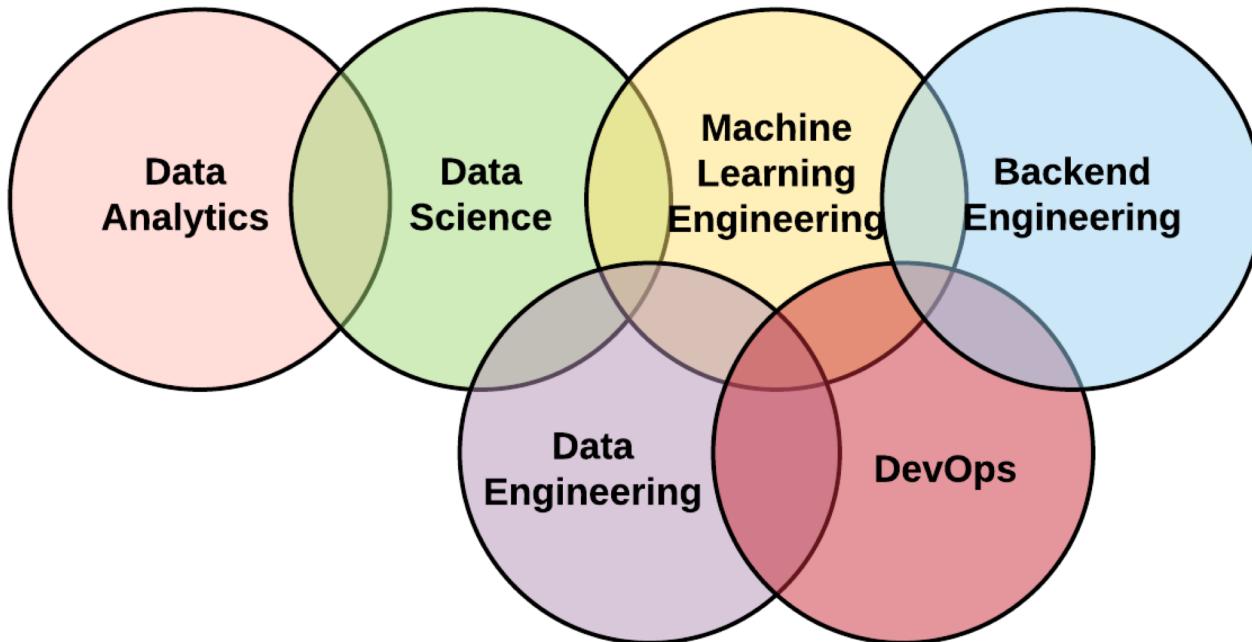
We have a solution MEME

**YOU JUST NEED**

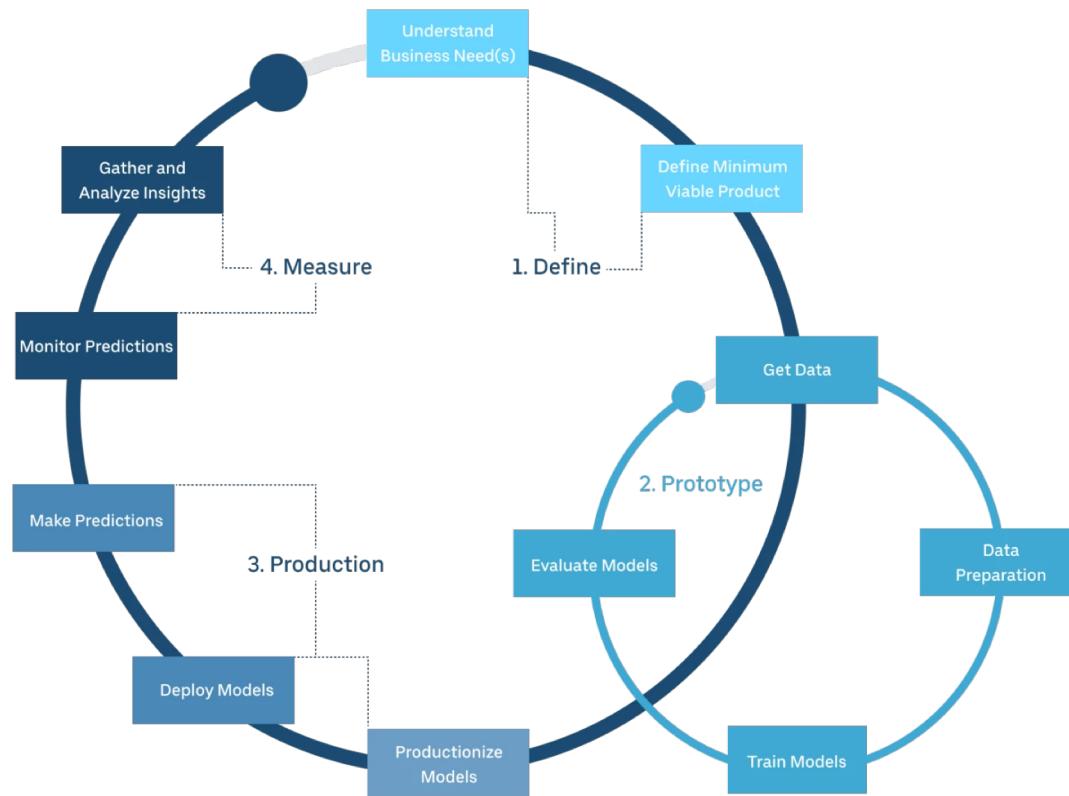


**MACHINE LEARNING ENGINEERING**

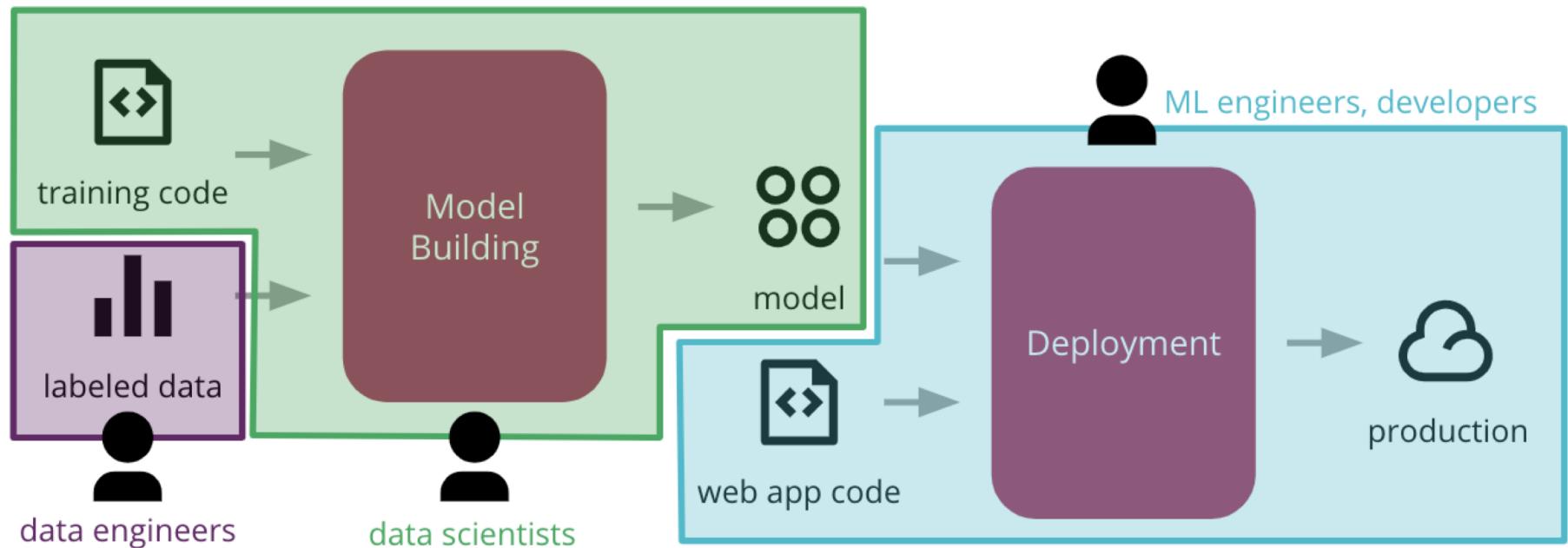
# ML Engineering



# Idealized Workflow of a Machine Learning Project



# ML Engineering



## ML code in the Context of ML backed System

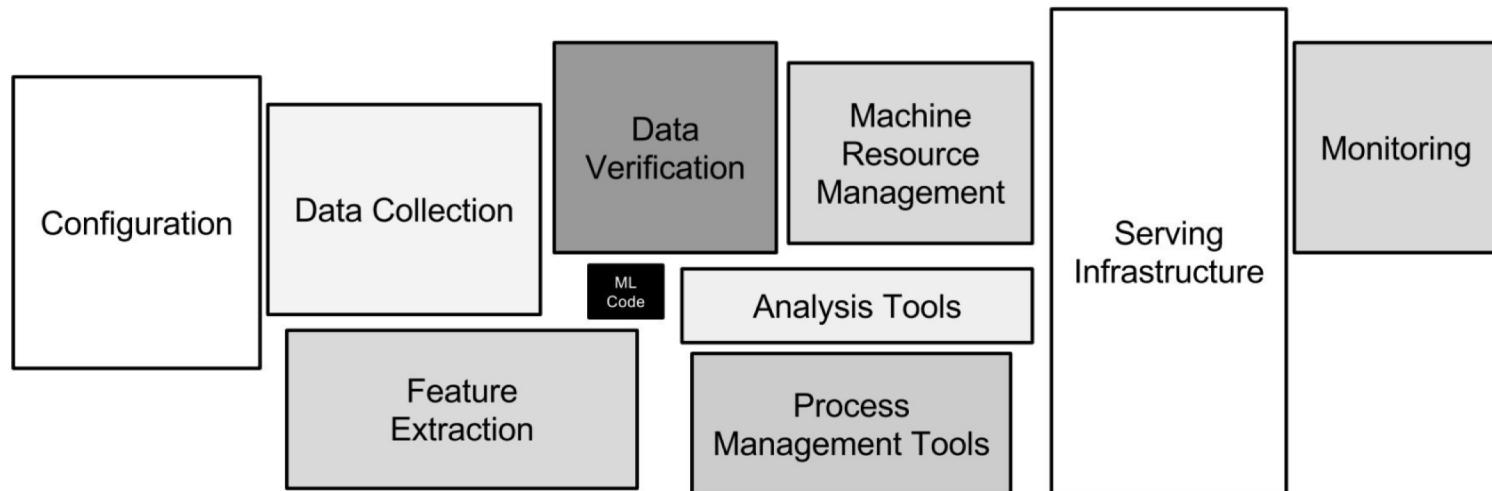
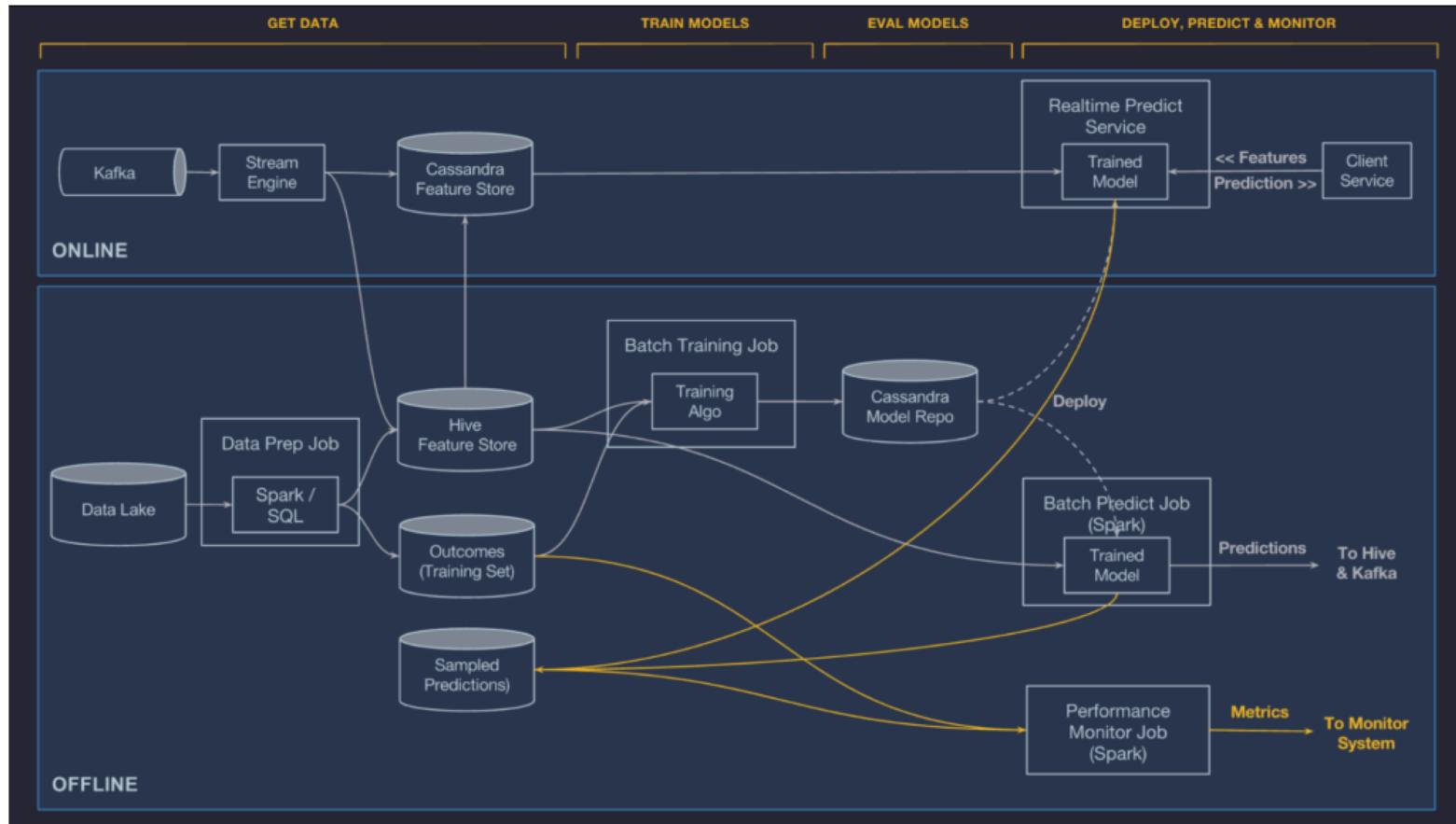


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

# **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

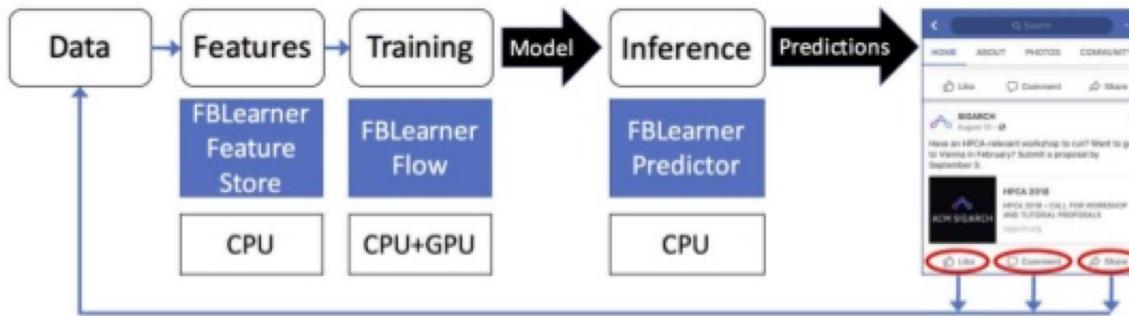


Fig. 1. Example of Facebook's Machine Learning Flow and Infrastructure.

# **Michelangelo - Main Principles**

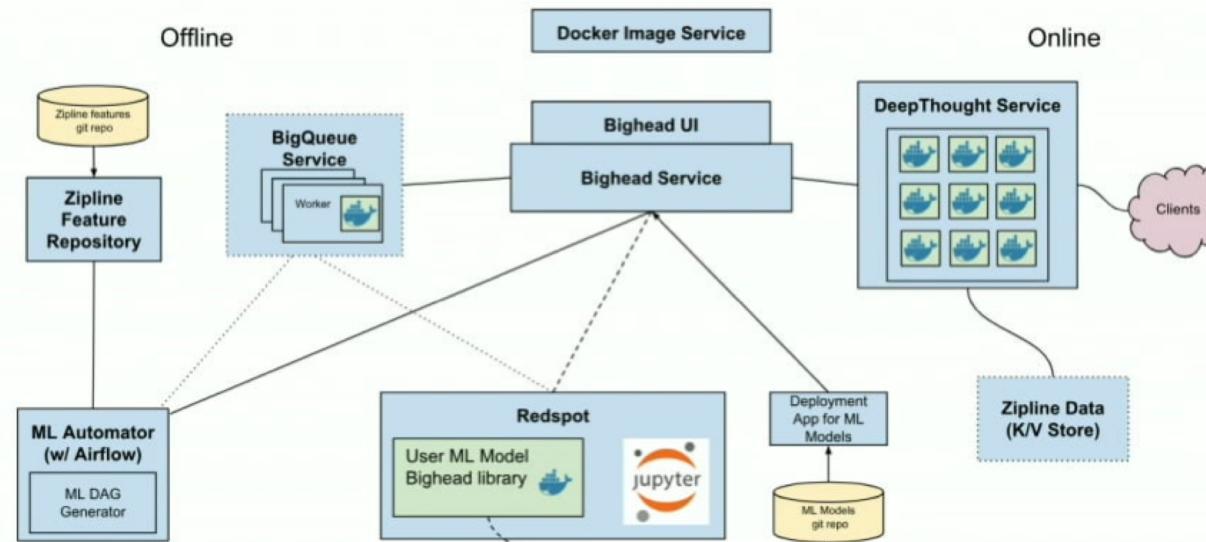
**Data Management:**

**Operations:**

**Tech Stack:**

# Big Head Overview

## Bighead Architecture



# **Michelangelo - Main Principles**

**Data Management:**

**Operations:**

**Tech Stack:**

# Machine Learning Comparative View

Bay Area MLflow Meetup - 20 June 2019

## Systems comparison



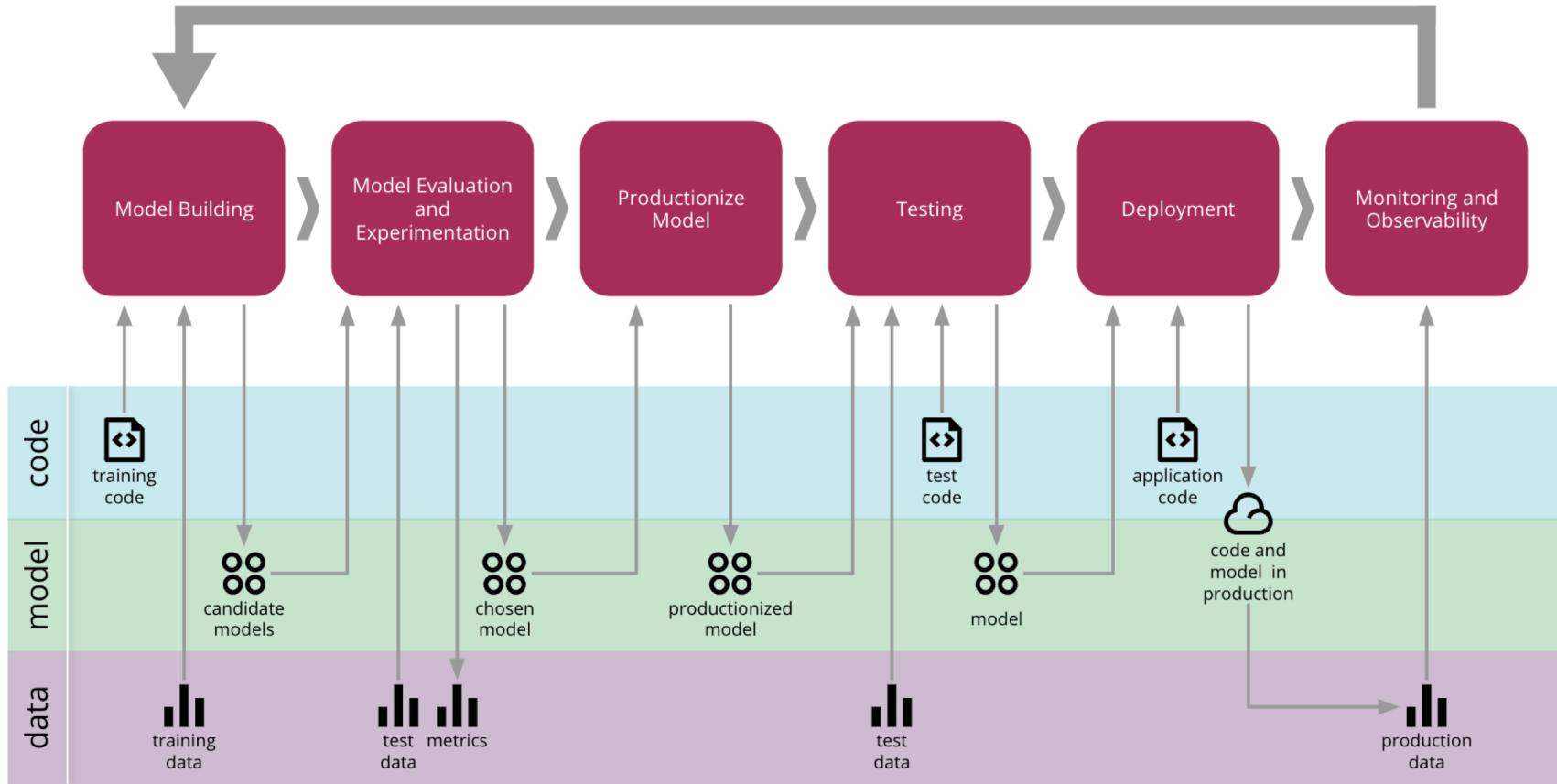
# BUILD YOU OWN COOL

**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](https://www.eecs.tufts.edu/~dsculley/papers/ml_test_score.pdf))

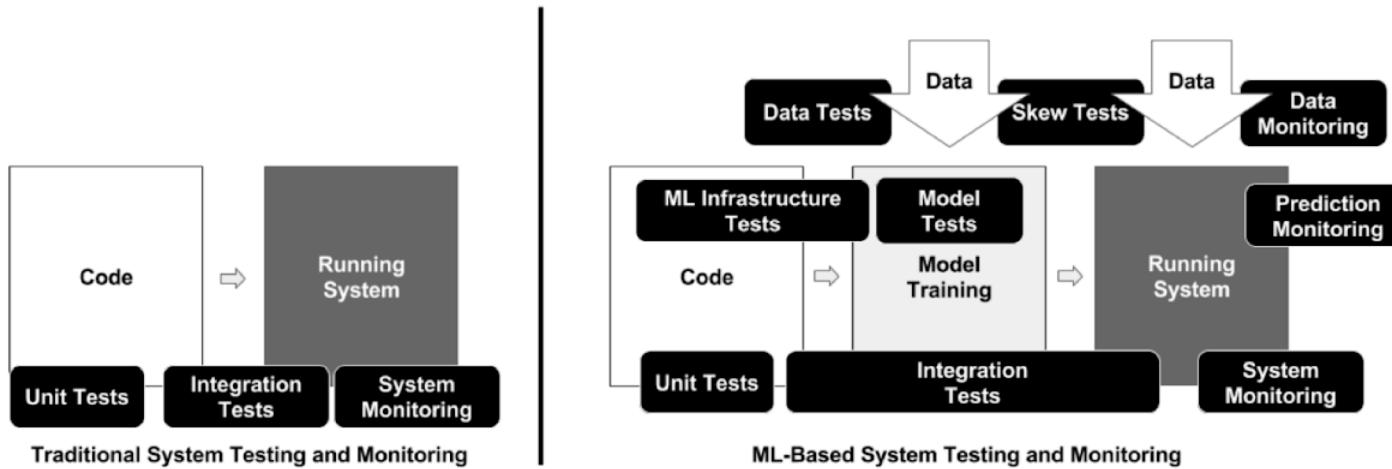
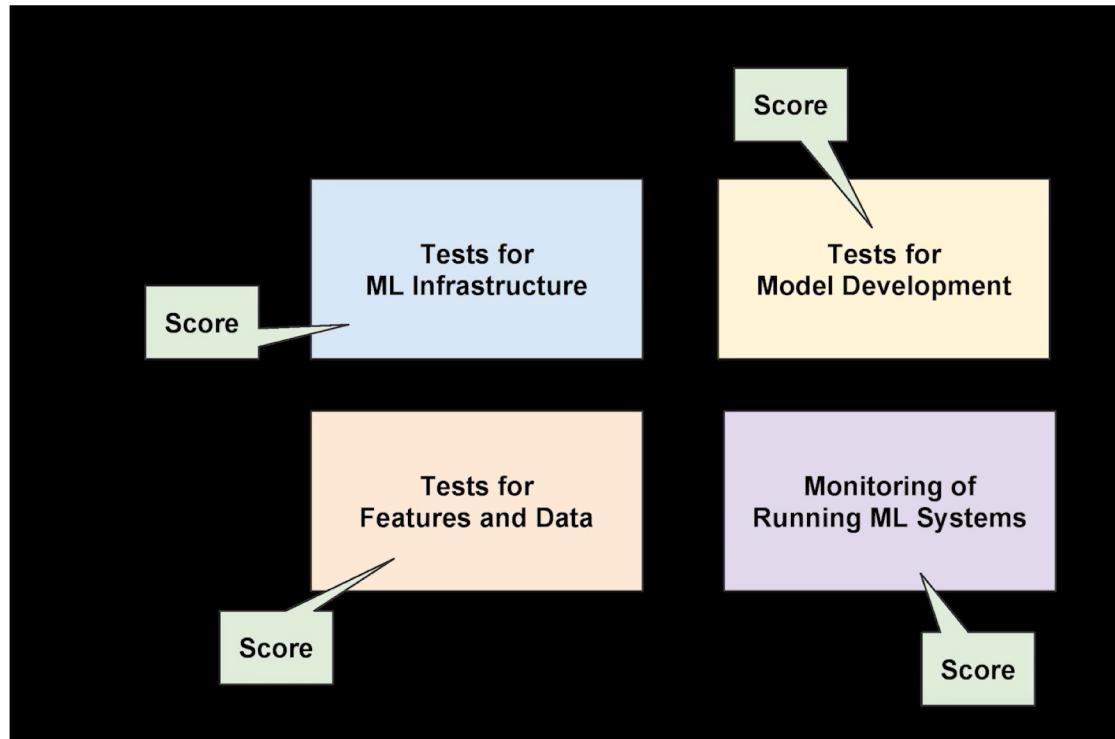


Figure 1. **ML Systems Require Extensive Testing and Monitoring.** The key consideration is that unlike a manually coded system (left), ML-based system behavior is not easily specified in advance. This behavior depends on dynamic qualities of the data, and on various model configuration choices.

Ideal Pipeline - Real Time Monitoring Perspective ( D. Sculley - <https://www.youtube.com/watch?v=V18AsBIHWs>)

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 the 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

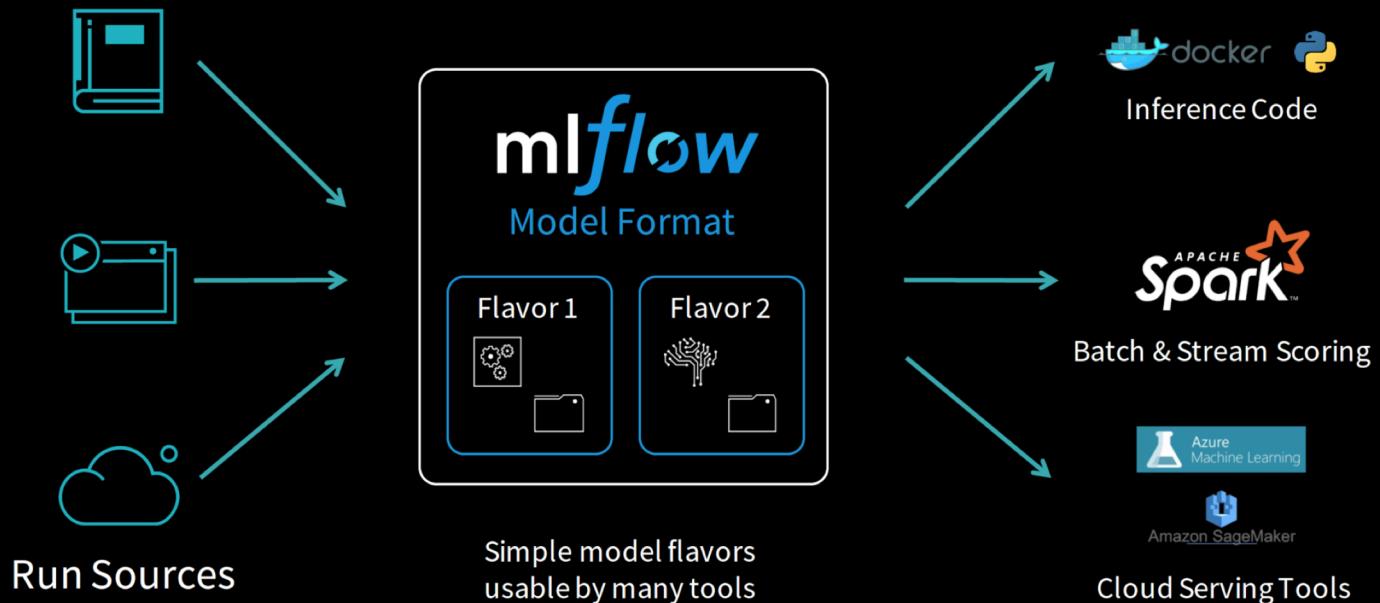
## mlflow Projects

Packaging format for reproducible runs on any platform

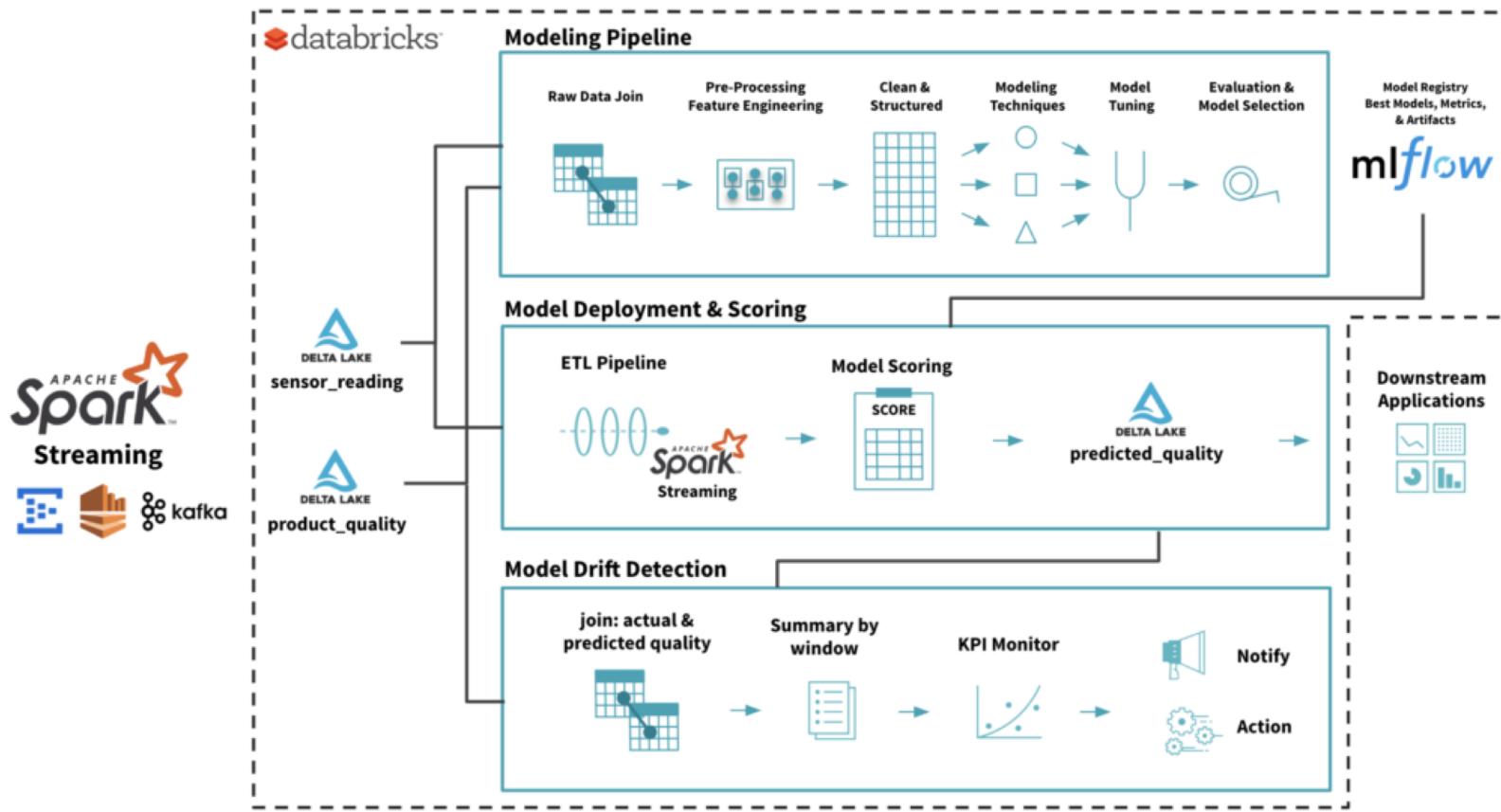
## mlflow Models

General model format that supports diverse deployment tools

# MLflow Models



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





## 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**

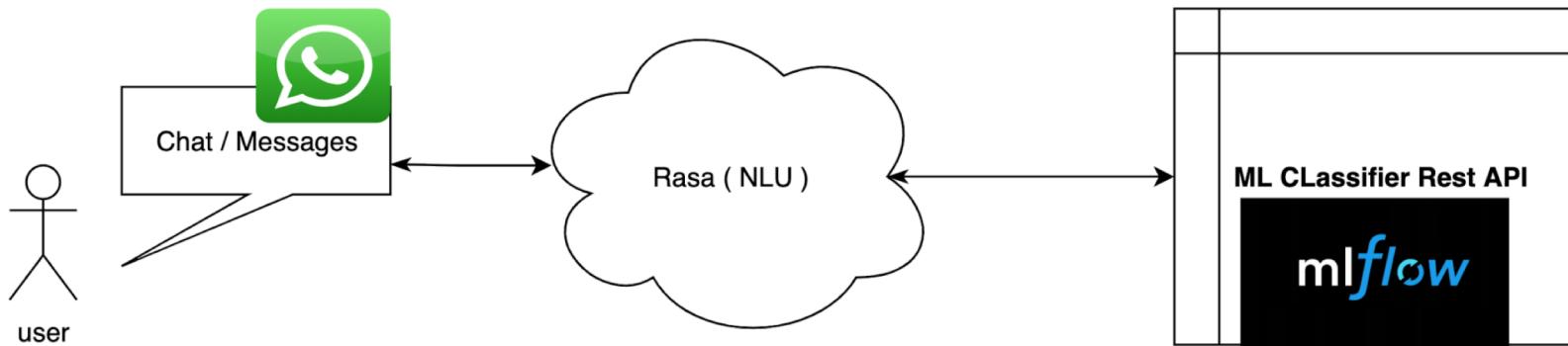
## Demo Example I - Bitpred - Data

<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

```
1
2 train, test = train_test_split(final_dataset, random_state=42, test_size=0.33, shuffle=True)
3 X_train = train.text
4 X_test = test.text
5
6 LogReg_pipeline = Pipeline([
7     ('tfidf', TfidfVectorizer(sublinear_tf=True, min_df=5, norm='l2', encoding='latin-1',
8         ('clf', LogisticRegression(solver='sag')),),
9     ])

```

view raw

## Demo Example - Log your model

```
1 with mlflow.start_run():
2     LogReg_pipeline.fit(X_train, train["label"])
3
4     # compute the testing accuracy
5     prediction = LogReg_pipeline.predict(X_test)
6     accuracy = accuracy_score(test["label"], prediction)
7
8     mlflow.log_metric("model_accuracy", accuracy)
9     mlflow.sklearn.log_model(LogReg_pipeline, "LogisticRegressionPipeline")
10
11
```

...

view raw

# Demo Example - Track Experiments

mlflow

[GitHub](#) [Docs](#)

Experiments



Default

Experiment ID: 0

Artifact Location: file:///Users/natu.lauchande/development/research/omdena/ptsd/task1\_machine\_learning\_backend/classification\_backend/mlruns/0

Search Runs: metrics.rmse < 1 and params.model = "tree"

State: Active

[Search](#)

Filter Params: alpha, lr

Filter Metrics: rmse, r2

[Clear](#)

Showing 25 matching runs

[Compare](#)

[Delete](#)

[Download CSV](#)

	Date	User	Run Name	Source	Version	Tags	Parameters	Metrics
<input type="checkbox"/>	2019-09-01 16:02:00	natu.lauchande	<a href="#">classifica...</a>	edfead				modelA1: 0.9375
<input type="checkbox"/>	2019-09-01 15:24:50	natu.lauchande	<a href="#">ipykerne...</a>					model_accuracy: 0.75
<input type="checkbox"/>	2019-09-01 10:09:45	natu.lauchande	<a href="#">ipykerne...</a>					model_accuracy: 0.75
<input type="checkbox"/>	2019-09-01 10:09:11	natu.lauchande	<a href="#">ipykerne...</a>					
<input type="checkbox"/>	2019-09-01 10:08:44	natu.lauchande	<a href="#">ipykerne...</a>					
<input type="checkbox"/>	2019-09-01 10:08:00	natu.lauchande	<a href="#">ipykerne...</a>					
<input type="checkbox"/>	2019-08-05 19:22:35	natu.lauchande	<a href="#">classifica...</a>	3f53ef				modelA1: 0.9375 modelB1: 1 modelB2: 1 modelB3: 1 modelB4: 0.9375 modelB5: 0.9375 modelC1: 0.9375 modelC2: 0.9375 modelD1: 1 modelD2: 0.9375 modelD3: 1 modelD4: 0.9375

# Demo Example - Model Tracking and Versioning

mlflow

Default > Run 1a6cde6a239d40ffb427bdf34df9e2c8 ▾

Date: 2019-09-01 15:24:50      Run ID: 1a6cde6a239d40ffb427bdf34df9e2c8      Source: ipykernel\_launcher.py      User: natu.lauchande

Duration: 409ms

▼ Notes 

Initial experiment with initial data not the most important relevant of all the experiments.

▼ Parameters

Name	Value
------	-------

▼ Metrics

Name	Value
model_accuracy ↗	0.75

▶ Tags

▼ Artifacts

LogisticRegressionPipeline

- MLmodel
- conda.yaml
- model.pkl

Full Path: file:///Users/natu.lauchande/development/research/omdena/ptsd/task1\_machine\_learning\_backend/classification\_backend/mlruns/0/1a6cde6a239d40ffb427bdf34df9e2c8/artifacts/LogisticRegressionPipeline/conda.yaml  
Size: 130B

```
channels:
- defaults
dependencies:
- python=3.6.1
- scikit-learn=0.20.3
- pip:
  - mlflow
  -云pickle==0.8.0
name: mlflow-env
```

# Demo Example - Model Tracking and Versioning

mlflow

Default > Run 1a6cde6a239d40ffb427bdf34df9e2c8 ▾

Date: 2019-09-01 15:24:50      Run ID: 1a6cde6a239d40ffb427bdf34df9e2c8      Source: ipykernel\_launcher.py      User: natu.lauchande

Duration: 409ms

▼ Notes 

Initial experiment with initial data not the most important relevant of all the experiments.

▼ Parameters

Name	Value
------	-------

▼ Metrics

Name	Value
model_accuracy ↗	0.75

▶ Tags

▼ Artifacts

LogisticRegressionPipeline

- MLmodel
- conda.yaml
- model.pkl

Full Path: file:///Users/natu.lauchande/development/research/omdena/ptsd/task1\_machine\_learning\_backend/classification\_backend/mlruns/0/1a6cde6a239d40ffb427bdf34df9e2c8/artifacts/LogisticRegressionPipeline/conda.yaml  
Size: 130B

```
channels:
- defaults
dependencies:
- python=3.6.1
- scikit-learn=0.20.3
- pip:
  - mlflow
  -云pickle==0.8.0
name: mlflow-env
```

## 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-483ff163345a1c89dc10599b1396df919493fb2  
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**