Introduction to Data Science
Lecture 10
Machine Learning
Part 2a

SEAS 6401 Fall 2019

Benjamin Harvey

OVERVIEW

- Introduction
  - Background (mine and yours)
- Organizational Personas
  - DoD Siloed and Proprietary Systems
  - Intelligence & Genomic Signals Analysis Processes
- Problem: Bias and Fairness in Machine Learning
  - Bias
  - Fairness
- Solution: Governance Framework for ML Algorithms
  - Explainability
  - Transparency
  - Accountability
- ML/AI Governance Platform: Managing the ML Lifecycle
  - Open Source MLflow

Organizational ML/AI Personas

Persona	Offering	Details
 Analyst/ Warfighter	 UI Conversational Agents, Recommendation Systems	/ Integrated in Application / AI Transparent to the User / Augmentation vs. Decision Making
 Citizen Data Scientist	databricks, jupyter, AutoML Toolkit, DataRobot, Michelangelo	/ Feature Factory / Feature Importance / Evolutionary Model Search
 ML Engineer	Keras, Caffe, TensorFlow, PyTorch, mlflow ML Frameworks and MLflow	/ Hyperparameter Tuning / Model Architecture Search
 ML Expert / Researcher	  ML Framework Optimizations	/ Distributed Execution of Libraries / Latest AutoML Libraries (e.g. Uber's Ludwig) /MLLib, Horovod, Hyperopt

Sources of Bias: These (1) **personas interact** with the (2) **model** and the (3) **data**

THE "Tipping Point" PROBLEM

- Find a balance between Microarray Gene Quality (MGQ) & accuracy/performance in Large-Scale Cancer Genomic (LSCG) datasets for **efficient cancer patient and cell line classification**.
- Increase the robustness by utilizing a "Tipping Point" gauge in a cloud-scale distributed parallel (**CSDP**) environment.

VINCENT J. CAREY, PhD
Brigham And Women's Hospital

EDUCATIONAL TITLES
Professor, Medicine, Harvard Medical School
Associate Biostatistician, Channing Laboratory, Brigham And Women's Hospital

DF/HCC PROGRAM AFFILIATION
Cancer Data Sciences, Member

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Packages: BiocParallel, GGtools

StarCluster @ MIT

amazon web services™ | EC2

Package 'parallel'

ML/AI Bias and Fairness

- **What is Bias in Data**
  - Types of Bias
  - Bias in Practice
  - Bias Variance Tradeoff
- Algorithmic Fairness
  - Types of Discrimination
- Combatting Bias

What is Bias

Data, especially big data, is often heterogeneous, generated by subgroups with their own characteristics and behaviors. The **heterogeneities**, some of which are described below, **can bias the data**. A model learned on biased data may lead to unfair and inaccurate predictions.

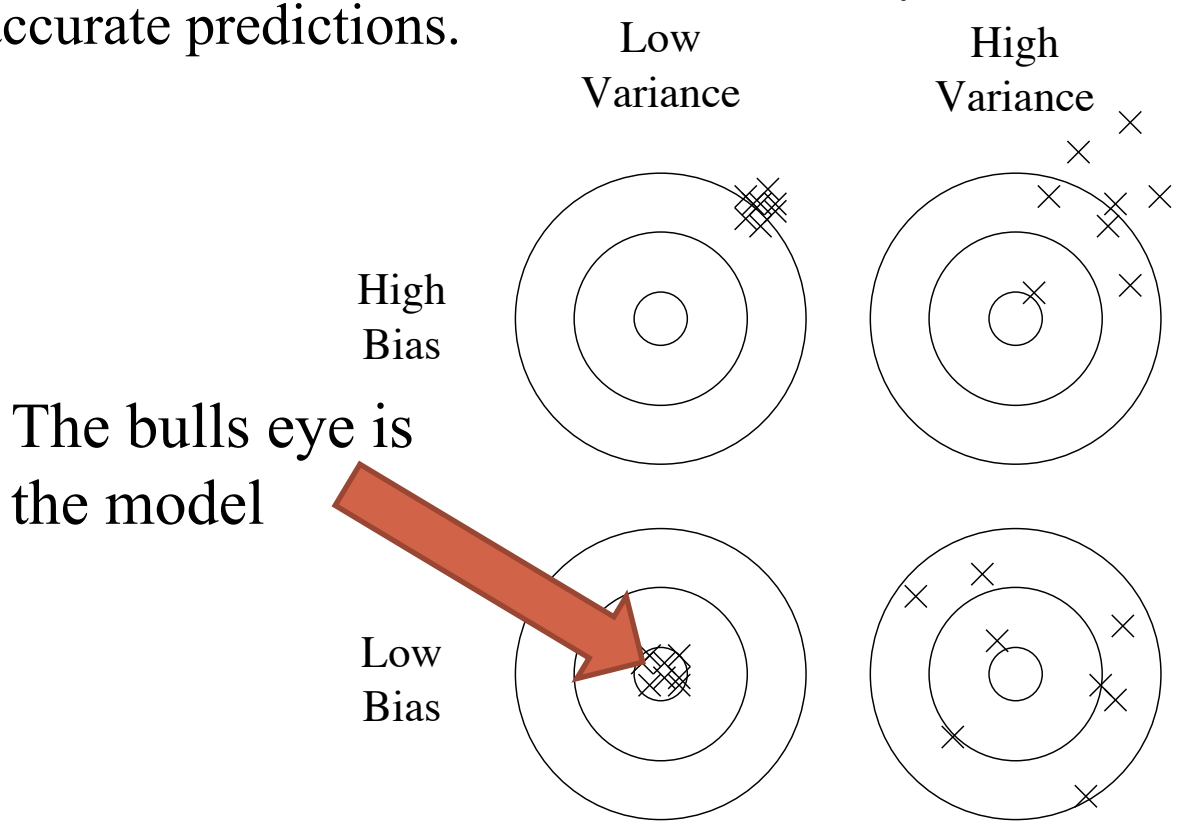


Figure 1: Bias and variance in dart-throwing.

ML/AI Bias and Fairness

- Bias in Data
  - **Types of Bias**
  - Bias in Practice
  - Bias Variance Tradeoff
- Algorithmic Fairness
  - Types of Discrimination
- Combatting Bias

Types of Bias

- (1) **Historical Bias**. Historical bias is the already existing bias and socio-technical issues in the world and can seep into from the data generation process even given a perfect sampling and feature selection.
- (2) Representation Bias. Representation bias happens from the way we define and sample from a population.
- (3) Measurement Bias. Measurement bias happens from the way we choose, utilize, and measure a particular feature.
- (4) Evaluation Bias. Evaluation bias happens during model evaluation.
- (5) **Aggregation Bias**. Aggregation bias happens when false conclusions are drawn for a subgroup based on observing other different subgroups or generally when false assumptions about a population affect the model's outcome and definition.

Types of Bias

- (6) **Population Bias**. Population bias arises when statistics, demographics, representatives, and user characteristics are different in the user population represented in the dataset or platform from the original target population.
- (7) Simpson's Paradox. Simpson's paradox can bias the analysis of heterogeneous data that is composed of subgroups or individuals with different behaviors.
  - A trend, association, or characteristic observed in underlying subgroups may be quite different from association or characteristic observed when these subgroups are aggregated
- (8) Longitudinal Data Fallacy. Observational studies often treat cross-sectional data as if it were longitudinal, which may create biases due to Simpson's paradox
- (9) **Sampling Bias**. Sampling bias arises due to non-random sampling of subgroups. As a consequence of sampling bias, the trends estimated for one population may not generalize to data collected from a new population.
- (10) **Behavioral Bias**. Behavioral bias arises from different user behavior across platforms, contexts, or different datasets.

Types of Bias

- (11) Content Production Bias. Content Production bias arises from structural, lexical, semantic, and syntactic differences in the contents generated by users.
- (12) **Linking Bias**. Linking bias arises when network attributes obtained from user connections, activities, or interactions differ and misrepresent the true behavior of the users.
- (13) Temporal Bias. Temporal bias arises from differences in populations and behaviors over time..
- (14) Popularity Bias. Items that are more popular tend to be exposed more. However, popularity metrics are subject to manipulation—for example, by fake reviews or social bots.
- (15) **Algorithmic Bias**. Algorithmic bias is when the bias is not present in the input data and is added purely by the algorithm.
- (16) User Interaction Bias. User Interaction bias is a type of bias that can not only be observant on the Web but also get triggered from two sources—the user interface and through the user itself by imposing his/her self-selected biased behavior and interaction. This type of bias can be influenced by other types and subtypes, such as Presentation and Ranking biases.
  - **Presentation Bias**. Presentation bias is a result of how information is presented. For example, on the Web users can only click on content that they see, so the seen content gets clicks, while everything else gets no click. And it could be the case that the user does not see all the information on the Web.
  - **Ranking Bias**. The idea that top-ranked results are the most relevant and important will result in attraction of more clicks than others. This bias affects search engines and crowdsourcing applications.

Types of Bias

- (17) Social Bias. Social bias happens when other people's actions or content coming from them affect our judgment.
- (18) Emergent Bias. Emergent bias happens as a result of use and interaction with real users.
- (19) Self-Selection Bias. Self-selection bias is a subtype of the selection or sampling bias in which subjects of the research select themselves.
- (20) **Omitted Variable Bias**. Omitted variable bias occurs when one or more important variables are left out of the model.
- (21) **Cause-Effect Bias**. Cause-effect bias can happen as a result of the fallacy that correlation implies causation.
- (22) Observer Bias. Observer bias happens when researchers subconsciously project their expectations onto the research.
- (23) Funding Bias. Funding bias arises when biased results are reported in order to support or satisfy the funding agency or financial supporter of the research study.

ML/AI Bias and Fairness

- Bias in Data
  - Types of Bias
  - **Bias in Practice**
  - Bias Variance Tradeoff
- Algorithmic Fairness
  - Types of Discrimination
- Combatting Bias

Bias in Practice

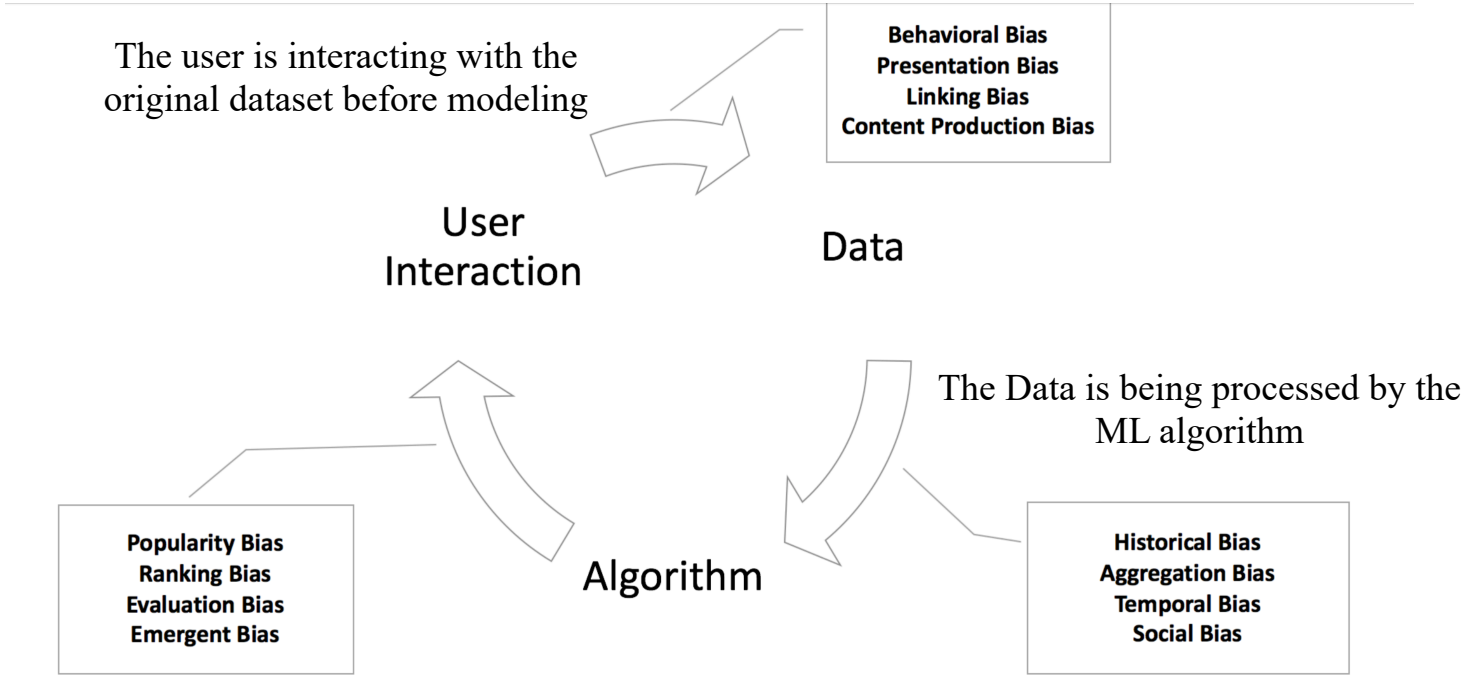
Persona


Analyst/
Warfighter


Citizen
Data
Scientist


ML
Engineer


ML Expert /
Researcher



ML/AI Bias and Fairness

- Bias in Data
  - Types of Bias
  - Bias in Practice
  - **Bias Variance Tradeoff**
- Algorithmic Fairness
  - Types of Discrimination
- Combatting Bias

Predicting from Samples

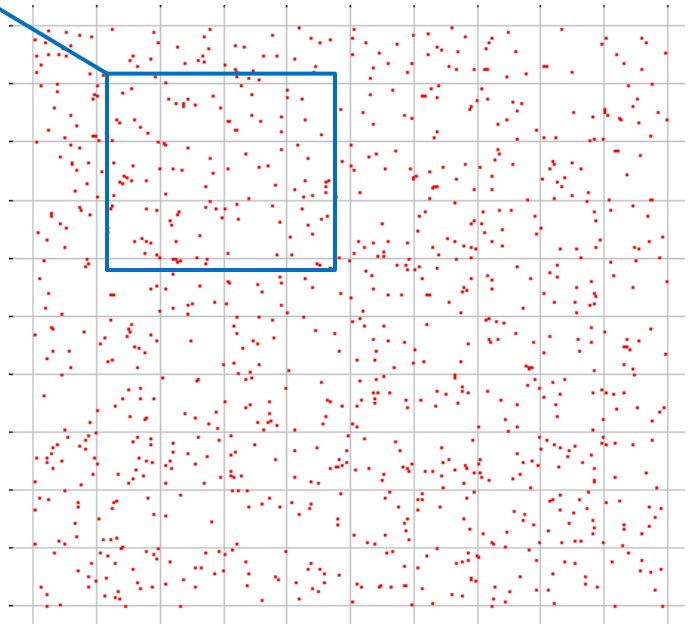
- Most datasets are **samples** from an **infinite population**.
- We are most interested in **models of the population**, but we have access only to a **sample** of it.

For datasets consisting of (X,y)

- features X + label y

a model is a prediction y = f(X)

We train on a training sample D
and we denote the model as $f_D(X)$



Bias and Variance

Our data-generated model $f_D(X)$ is a **statistical estimate** of the true function $f(X)$.

Because of this, its subject to bias and variance:

Bias: if we train models $f_D(X)$ on many training sets D, bias is the expected difference between their predictions and the true y's.

i.e. $$Bias = \mathrm{E}[f_D(X) - y]$$

E[] is taken over points X and datasets D

Variance: if we train models $f_D(X)$ on many training sets D, variance is the variance of the estimates:

$$Variance = \mathrm{E}\left[\left(f_D(X) - \bar{f}(X)\right)^2\right]$$

Where $\bar{f}(X) = \mathrm{E}[f_D(X)]$ is the average prediction on X.

Bias and Variance Tradeoff

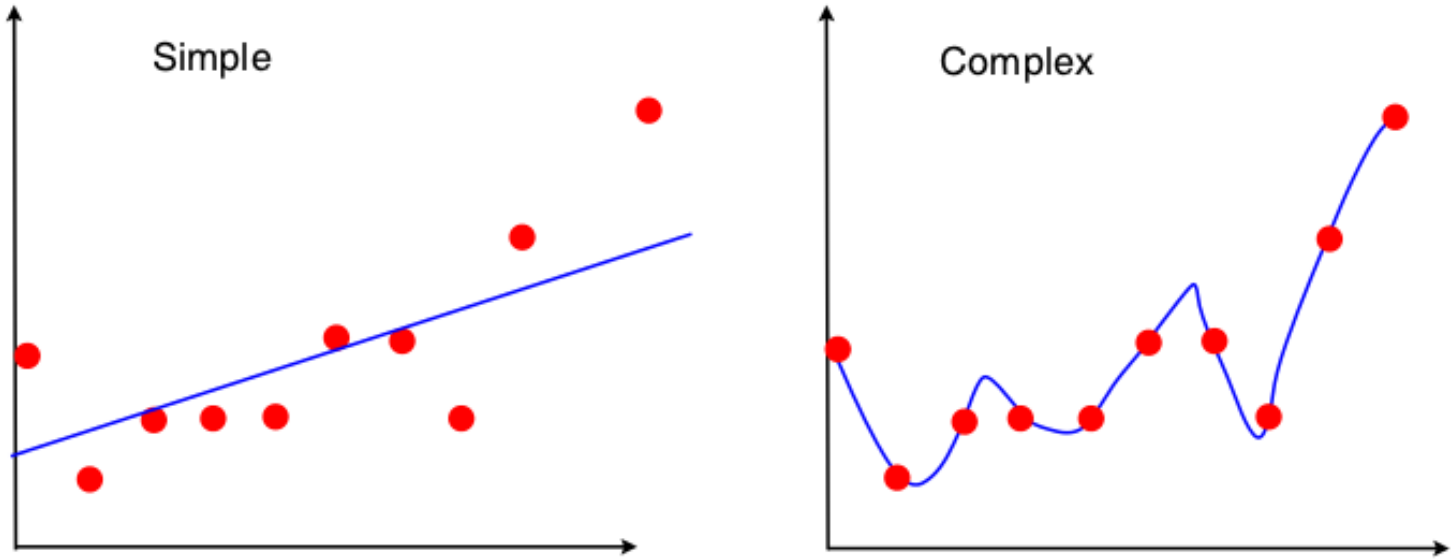
There is usually a bias-variance tradeoff caused by model complexity.

Complex models (many parameters) usually have lower bias, but higher variance.

Simple models (few parameters) have higher bias, but lower variance.

Bias and Variance Tradeoff

e.g. a linear model can only fit a straight line. A high-degree polynomial can fit a complex curve. But the polynomial fits the individual sample, and not the underlying population. Its shape can vary from sample to sample, so it has high variance.



Bias and Variance Tradeoff

The total expected error is

$$Bias^2 + Variance$$

Because of the bias-variance trade-off, we want to **balance** these two contributions.

If $Variance$ strongly dominates, it means there is too much variation between models. This is called **over-fitting**.

If $Bias$ strongly dominates, then the models are not fitting the data well enough. This is called **under-fitting**.

ML/AI Bias and Fairness

- Bias in Data
  - Types of Bias
  - Bias in Practice
  - Bias Variance Tradeoff
- **Algorithmic Fairness**
  - Types of Discrimination
- Combatting Bias

Algorithmic Fairness

- Broadly, algorithmic fairness is the **absence of any prejudice or favoritism towards the an feature/observation, or a group** within an underlying dataset based on their intrinsic or acquired traits in the context of ML decision-making.
- Even though fairness is an incredibly desirable quality in society, it can be surprisingly difficult to achieve in practice.

Sources of Bias: These (1) **personas** **interacts** with the (2) **model** and the (3) **data**

ML/AI Bias and Fairness

- Bias in Data
  - Types of Bias
  - Bias in Practice
  - Bias Variance Tradeoff
- Algorithmic Fairness
  - **Types of Discrimination**
- Combatting Bias

Types of Discrimination

- (1) Direct Discrimination. Direct discrimination happens when protected attributes of individuals explicitly result in non-favorable outcomes toward them.
- (2) Indirect Discrimination. In Indirect discrimination, individuals appear to be treated based on seemingly neutral and non-protected attributes; however, protected groups or individuals still get to be treated unjustly as a result of implicit effects from their protected attributes.
- (3) Systemic Discrimination. Systemic discrimination refers to policies, customs, or behaviors that are a part of the culture or structure of an organization that may perpetuate discrimination against certain subgroups of the population.
- (4) **Statistical Discrimination**. Statistical discrimination is a phenomenon where decision-makers use average group statistics to judge an individual belonging to that group.
- (5) Explainable Discrimination. Differences in treatment and outcomes amongst different groups can be justified and explained via some attributes in some cases.
- (6) Unexplainable Discrimination. In contrast to explainable discrimination, there is unexplainable discrimination in which the discrimination toward a group is unjustified and therefore considered illegal.

Combatting Bias

- (1) **Pre-processing**. Pre-processing techniques try **to measure then transform the data** so that the underlying discrimination is removed. If the algorithm is allowed to modify the training data, then pre-processing can be used.
- (2) **In-processing**. In-processing techniques try to **modify and change state-of-the-art learning algorithms** in order to remove discrimination during the model training process. If it is allowed to change the learning procedure for a machine learning model, then in-processing can be used during the training of a model— either by incorporating changes into the objective function or imposing a constraint
- (3) **Post-processing**. Post-processing is performed after training by **accessing a holdout set** which was not involved during the training of the model. If the algorithm can only treat the learned model as a black box without any ability to modify the training data or learning algorithm, then only post-processing can be used in which the labels assigned by the black-box model initially get reassigned based on a function during the post-processing phase

Combatting Bias

Domain	Sub-domain	Reference(s)
Data	Simpson's Paradox	[68]
Machine learning	Classification	[65] [86] [49] [72] [121] [55] [131] [127] [59] [24] [128]
Machine learning	Regression	[13] [1]
Machine learning	PCA	[111]
Machine learning	Community detection	[85]
Machine learning	Clustering	[30] [7]
Machine learning	Graph embedding	[21]
Machine learning	Causal inference	[81] [136] [137] [132] [95] [94] [134] [69] [105] [133]
Natural language processing	Word embedding	[19] [141] [50] [22] [138]
Natural language processing	Coreference resolution	[140] [109]
Natural language processing	Language model	[20]
Natural language processing	Sentence embedding	[84]
Natural language processing	Machine translation	[45]
Natural language processing	Semantic role labeling	[139]
Deep learning / representation learning	Variational auto encoders	[82] [4] [92] [38]
Deep learning / representation learning	Adversarial learning	[76] [129]

Table 2. List of papers targeting and talking about bias and fairness in different areas and sub-areas of machine learning.

ML/AI Governance

- ML/AI Governance Framework
- Transparency
  - Why Transparency?
  - Transparency of What?
  - Transparency Risks
- Accountability
  - Why do we Need Accountability?
  - Explainability
- Managing the ML Lifecycle: MLflow

ML/AI Governance Framework

- At the higher level two approaches are considered, **principles (e.g. people, processes, and technology) vs rules-based approaches** and regulation related to algorithms as a single regulatory category or rather as a kind of helper technology that should be regulated as a component of other technologies.
- Consideration of principles vs rules-based approaches in the context of technology-related governance reveals that much of the existing literature/practice focuses on **risks-oriented principles based approaches**.
- Methods emphasize maximizing the benefits and minimizing the risks that arise from the use of the technology by allocating resources in proportion to risks to society, considering both the impacts themselves and the likelihood that they happen, in order to establish appropriate levels of control.
- One common tool used to support risk-based approaches is an impact assessment.

Transparency

- Depending on the type and use of an algorithmic decision system, the desire for algorithmic transparency may refer to one, or more of the following aspects:
  - code, logic, model, goals (e.g. optimization targets), decision variables, or some other aspect that is considered to provide insight into the way the algorithm performs.
- Algorithmic system transparency can be global, seeking insight into the system behavior for any kind of input, or local, seeking to explain a specific input - output relationship.

Why Transparency?

We want systems to be transparent not to satisfy idle curiosity, but to help achieve important social goals related to accountability:

We want to inspect an algorithmic system's data and algorithms to:

- Check for bias in the data and algorithms that affects the fairness of the system. (The mechanics, costs, and secondary effects are different when checking data vs. algorithms).
- Check that the system is drawing inferences from relevant and representative data.
- See if we can learn anything from the machine's way of connecting and weighting the data - perhaps there's a meaningful correlation we had not been aware of.
- Look for, and fix, bugs.
- Guard against malicious/adversarial data injection.

Why Transparency?

We want the hierarchy of goals and outcomes to be transparent so:

- It can be debated and possibly regulated.
- Regulators and the public can assess how well an algorithmic system has performed relative to its goals, and compared to the pre-algorithmic systems it may be replacing or supplementing.

We want an organization's compliance status to be public so:

- Regulators can hold the organization accountable in case of failure.
- The public can evaluate the trustworthiness of the organization, so people can make informed decisions as users about the services offered, and so citizens can become better informed about the benefits, risks, and trade-offs of algorithmic-based services overall.

Transparency of What?

There are seven broad areas of machine learning systems about which transparency might be demanded:

- 1. **Data (Datasheets for Datasets)**. The transparency of the data used by the algorithmic system -- in particular by machine learning and deep learning algorithms -- can refer to the raw data, to the data's sources, to how the data were preprocessed, to the methods by which it was verified as unbiased and representative (including looking for features that are proxies for information about protected classes), or to the processes by which the data are updated and the system is retrained on them.
- 2. **Algorithms (Model Cards)**. The transparency of the systems' algorithms can refer to testing its output against inputs for which we know the proper output, reducing the variables to the most significant so we can validate them, testing the system with counterfactuals to see if prejudicial data is infecting the output, a third party code review, analysis of how the algorithms work, inspection of internal and external bug reports, or assurance the software development processes are sound.
- 3. **Goals**. Algorithmic systems can also be transparent about their goals. When a system has multiple goals, this would mean being transparent about their relative priorities.
- 4. **Outcomes**. Manufacturers or operators could be required to be transparent about the outcomes of the deployment of their algorithmic systems, including the internal states of the system (how worn are the brakes of an AV? how much electricity used?), the effects on external systems (how many accidents, or times it's caused another AV to swerve?), and computer-based interactions with other algorithmic systems (what communications with other AVs, what data fed into traffic monitoring systems?).
- 5. **Compliance**. Manufacturers or operators may be required to be transparent about their overall compliance with whatever transparency requirements have been imposed upon them. In many instances, we may insist that these compliance reports are backed by data that is inspectable by regulators or the general public.
- 6. **Influence**. Just as the public has an interest in knowing if an article in a newspaper was in fact paid for by an interested party, the public may have an interest in knowing if any element of the AI process was purposefully bent to favor a particular outcome (e.g. **boosting**).
- 7. **Usage**. Users may want to know what personal data a system is using, either to personalize outcomes or as data that can train the system to refine it or update it.

Transparency Risks

Potential costs include:

- Regulatory bodies have to have **staff sufficient to oversee compliance**.
- Businesses and other organizations have to create and **maintain the processes, code, and legal oversight required by the regulatory bodies. (e.g. Mlflow)**.
- Transparency might put justifiable **trade secrets at risk**.
- **Public access to data** can flame interest-driven controversies via the untutored or unscrupulous misuse of data.
- The requirement for transparency can lead to the **use of algorithms that are suboptimal for their purposes**, resulting in what can be serious harms when compared with achievable goods.
- Access to data that seems innocuous can lead to breaches of personal privacy by clever and determined hackers (e.g. **privacy vs. utility**).
- Increased transparency of algorithms can make them easier to hack for malicious purposes (e.g. **adversarial ML / data or model poisoning**).

Accountability

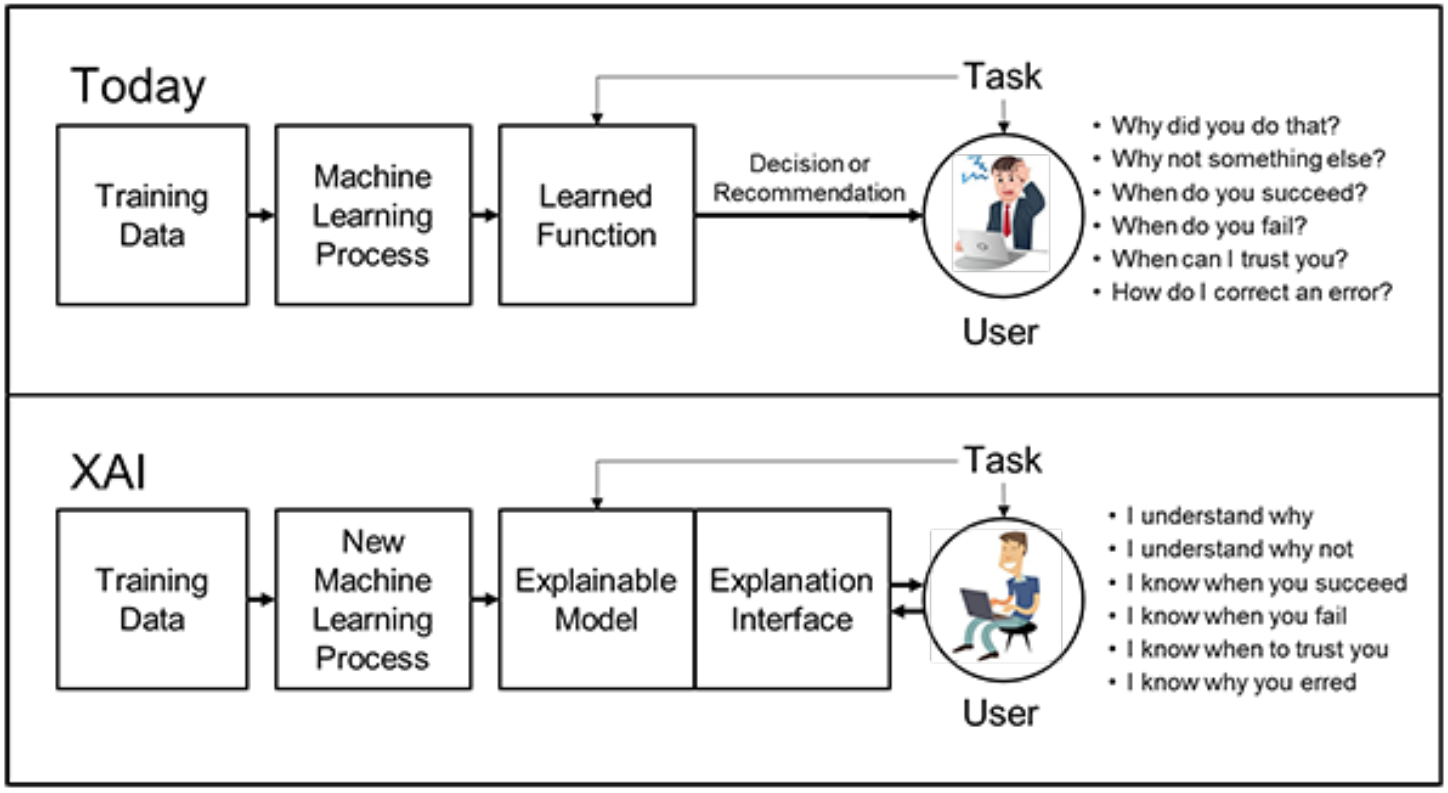
- Accountability serves to **ensure responsible development and use of algorithmic systems such that they improve human rights and benefit society**
- Accountability is primarily a **legal and ethical obligation** on an individual or organization to account for its activities, accept responsibility for them, and to disclose the results in a transparent manner.
- Transparency, logs of data provenance, code changes and other record keeping are important technical tools but ultimately accountability depends on establishing clear chains of responsibility.
- Accountability **ultimately lies with a (legal) person**

Why Accountability

In the context of algorithmic systems, the challenges arise from:

- **Complex interactions between sub-systems and data sources**, some of which might not be under the control of the same entity (e.g. systems relying on data acquired through data brokers who rely on data sources that use algorithmic inference to aggregate over 'similar' data subjects). A governance framework for algorithmic accountability and transparency
- **Unexpected outcomes associated** with the impossibility of testing against all possible input conditions when there are no methods for generating formal proofs for the system's performance.
- **Difficulties in translating algorithmically derived concepts** (e.g. clustering algorithm results that segment populations based on large numbers of input variables) into human understandable concepts (e.g. ethnic affiliation) resulting in incorrect interpretations of the meaning of algorithmic results.
- **Information asymmetries arising from algorithmic inferences** and black box processes that make it all but impossible for data subjects to gage which, potentially false, information might have resulted in a particular algorithmic outcome affecting them (including lack of knowledge that algorithmic processes were even involved).
- **Ubiquity of (small) algorithmic decisions** which, if systematically biased, may accumulate to have significant impacts on people even though no single decision would have achieved that legal threshold (e.g. impact on personal development due to reinforcement of racial/gender stereotypes by algorithmic recommendations).
- **Purposeful injections of adversarial data** to fool a system into making errors, often in ways that can be very difficult to detect.

Explainable Artificial Intelligence

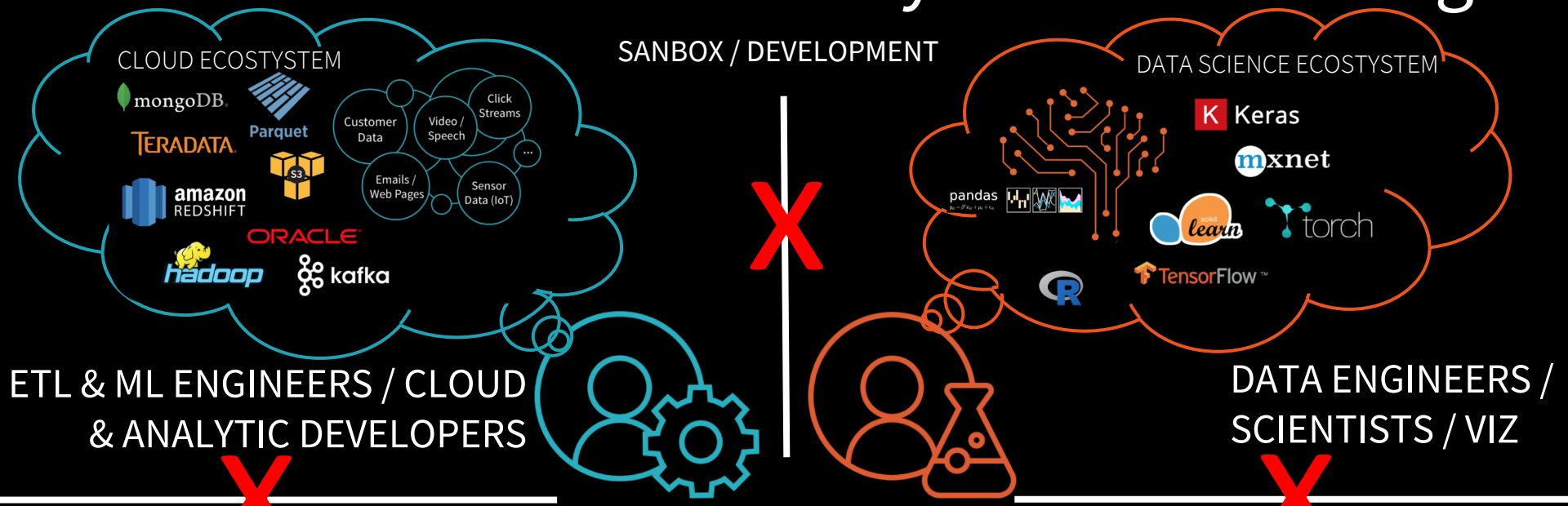


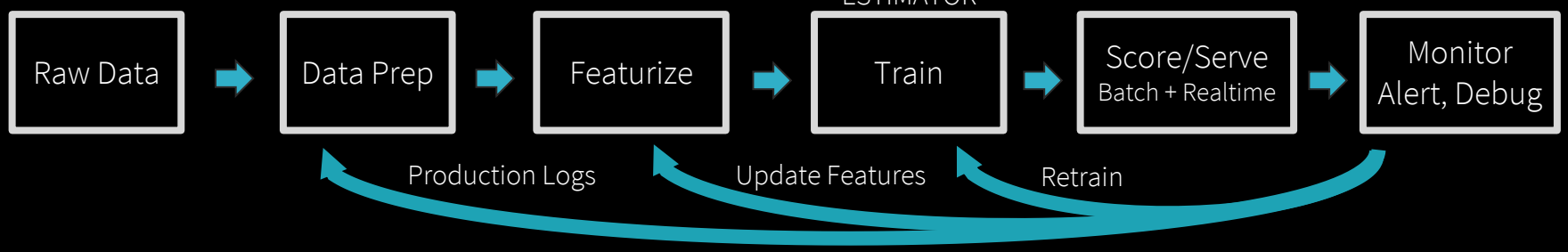
Source: https://www.darpa.mil/program/explainable-artificial-intelligence

Introduction to mlflow

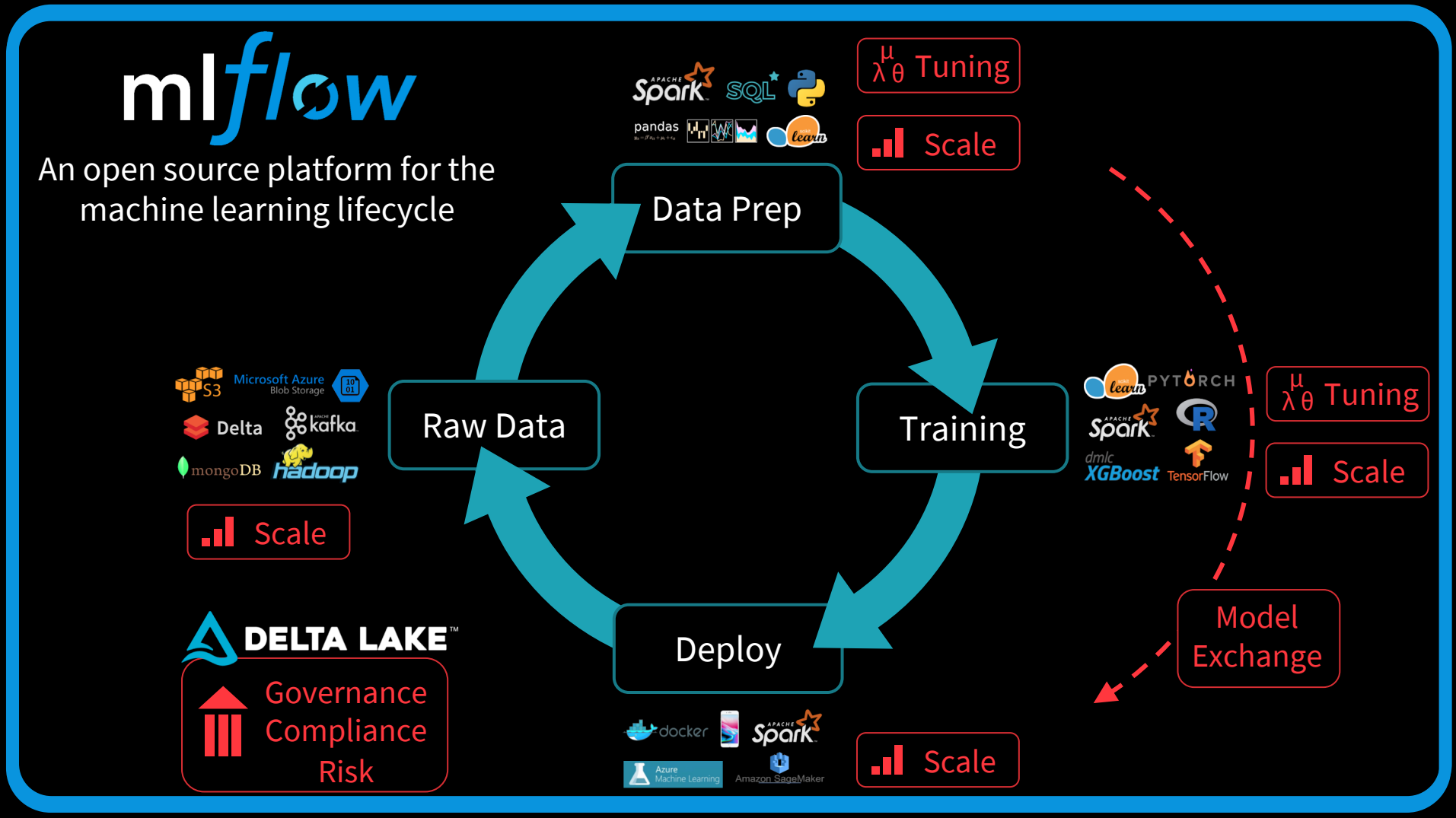
databricks

Data & ML Silos – Difficulty Productionizing





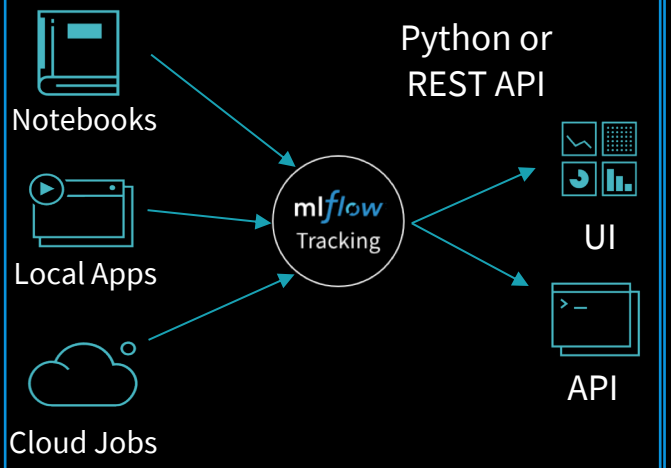
databricks


mlflow
An open source platform for the machine learning lifecycle
Data Prep
Training
Deploy
Raw Data
Tuning
Scale
Tuning
Scale
Scale
Scale
Model Exchange
DELTA LAKE
Governance
Compliance
Risk
Microsoft Azure Blob Storage
Delta
kafka
mongoDB
hadoop
PYTORCH
XGBoost
TensorFlow
docker
Azure Machine Learning
Amazon SageMaker

MLflow Components

mlflow Tracking

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



Parameters: key-value inputs to your code

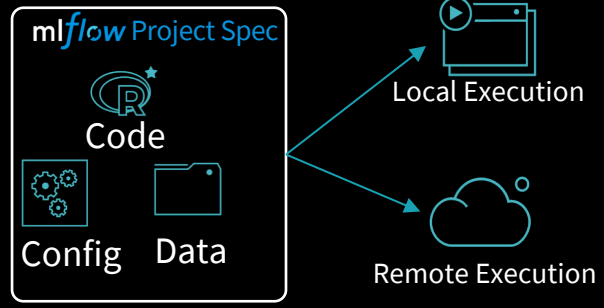
Metrics: numeric values (can update over time)

Artifacts: arbitrary files, including models

Source: what code ran?

mlflow Projects

Packaging format for reproducible runs on any platform

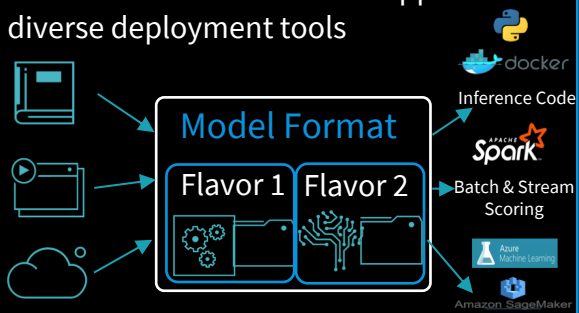



```
my_project/
├── Mlproject
│     conda_env: conda.yaml
│
│     entry_points:
│       main:
│         parameters:
│           training_data: path
│           lambda: {type: float, default: 0.1}
│         command: python main.py {training_data} {lambda}
├── conda.yaml
├── main.py
└── model.py
...
```

```
$ mlflow run git://<my_project>
mlflow.run("git://<my_project>", ...)
```

mlflow Models

General model format that supports diverse deployment tools




```
my_model/
├── MLmodel
│     run_id: 769915006efd4c4bbd662461
│     time_created: 2018-06-28T12:34
│     flavors:
│       tensorflow:
│         saved_model_dir: estimator
│         signature_def_key: predict
│       python_function:
│         loader_module: mlflow.tensorflow
└── estimator/
    ├── saved_model.pb
    └── variables/
```

databricks

mlflow Workflows and Model Registry

mlflow Workflows

Easy sharing + editing of multi-step pipelines

- Pipelines: Define in code and edit in UI
- Automatically reuse results
- Runs on existing job schedulers like Apache Airflow

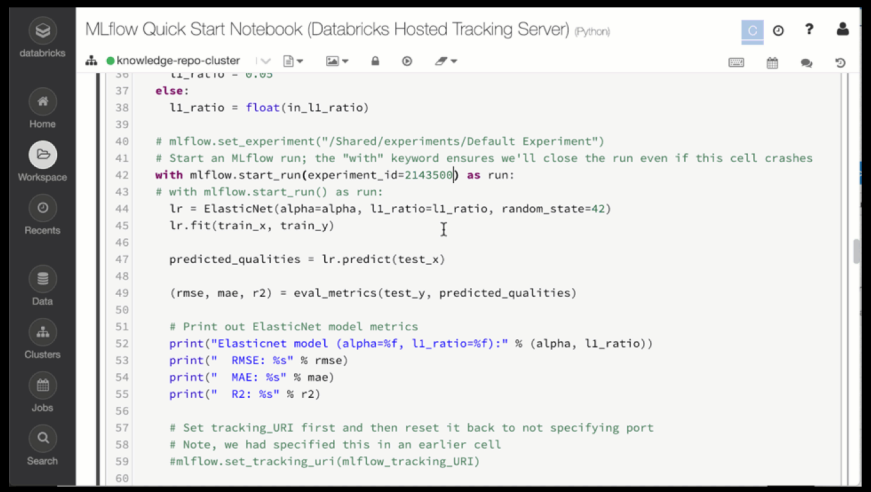
mlflow Model Registry

Manage, tag & version models in Mlflow server

- Promote any model in the Mlflow Models format
- Deploy to inference systems
- See model users and add metadata

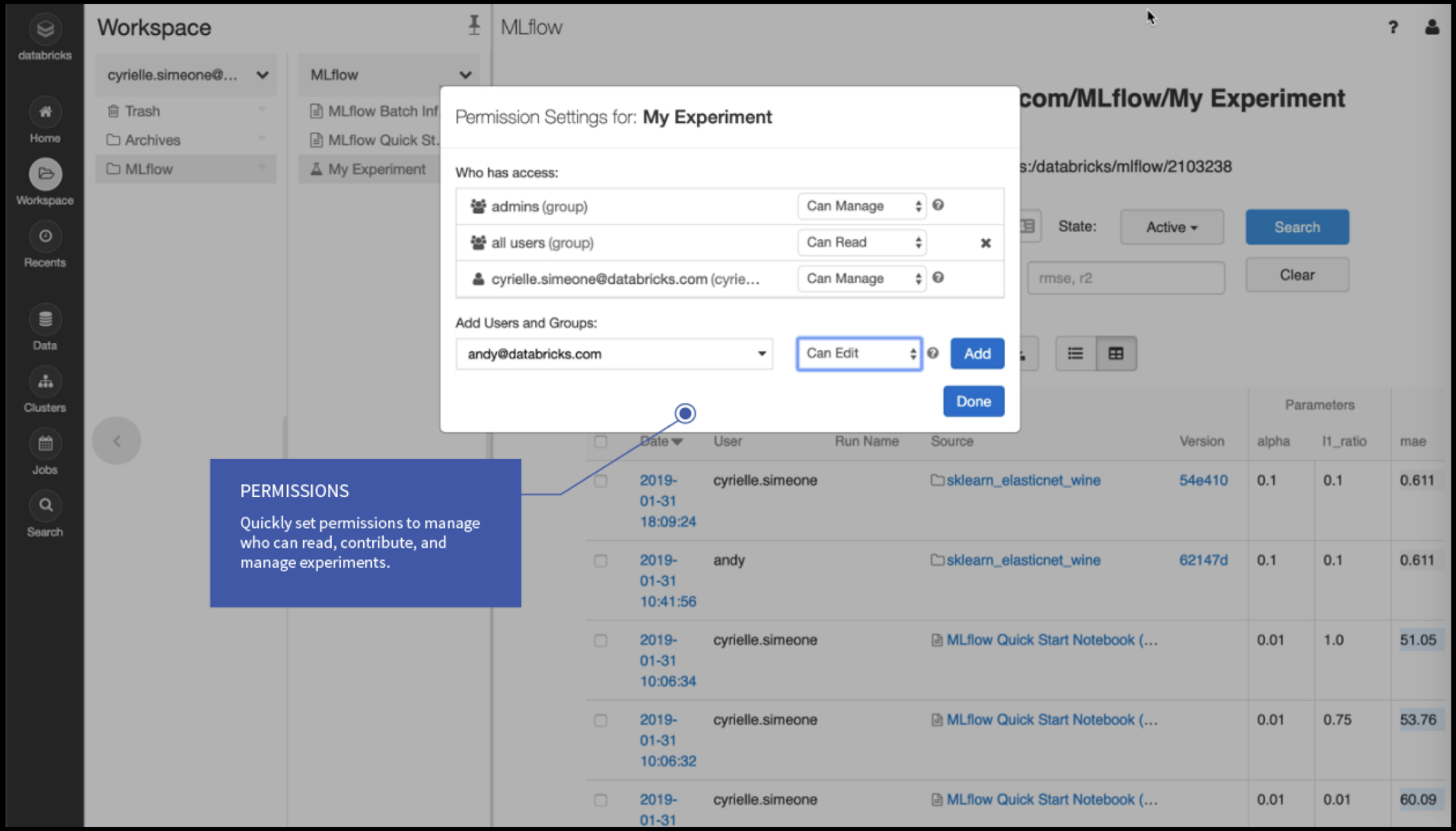
Experiment Tracking

Record runs, and keep track of models parameters, results, code, and data from each experiment and in one place.



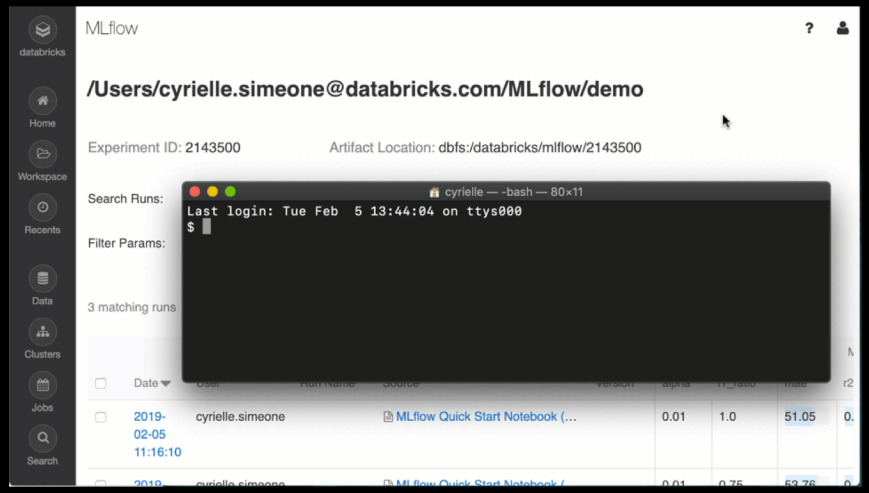
Managed MLflow Provides:

- Pre-configured MLflow tracking server
- Databricks Workspace & Notebooks UI integration
- S3, Azure Blob Storage, Google Cloud for artifacts storage
- Experiments management via role based Access Control Lists (ACLs)


Workspace
MLflow
Permission Settings for: My Experiment
Who has access:
admins (group)
Can Manage
all users (group)
Can Read
cyrielle.simeone@databricks.com (cyrie...
Can Manage
Add Users and Groups:
andy@databricks.com
Can Edit
Add
Done
PERMISSIONS
Quickly set permissions to manage who can read, contribute, and manage experiments.

Reproducible Projects

Build composable projects, capture dependencies and code history for reproducible results, and share projects with peers.



Managed MLflow Provides:

- Support for Git, Conda, and other file storage systems
- Remote execution via command line as a Databricks Job

databricks

Home

Workspace

Recents

Data

Clusters

Jobs

Search

MLflow

/Users/cyrielle.simeone@databricks.com/... > Run 1359afd3b6d341719f1ff4e2afcdedea

Date: 2019-01-30 11:16:16 Run ID: 1359afd3b6d341719f1ff4e2afcdedea

Source: MLflow Quick Start Notebook (Databricks Hosted Tracking Server) User: cyrielle.simeone@databricks.com

Duration: 1.7s

Notes

None

Parameters

Name	Value
alpha	0.01
l1_ratio	1.0

Metrics

Name	Value
mae	51.05
r2	0.395
rmse	63.25

Tags

Artifacts

ElasticNet-paths.png

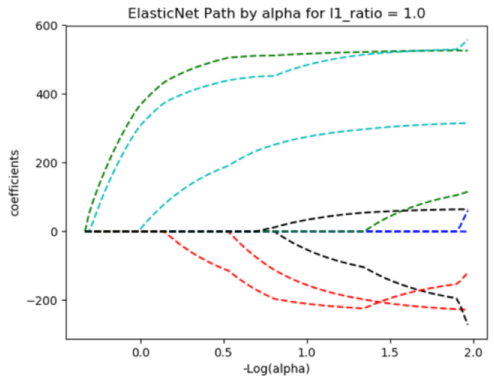
model

MLmodel

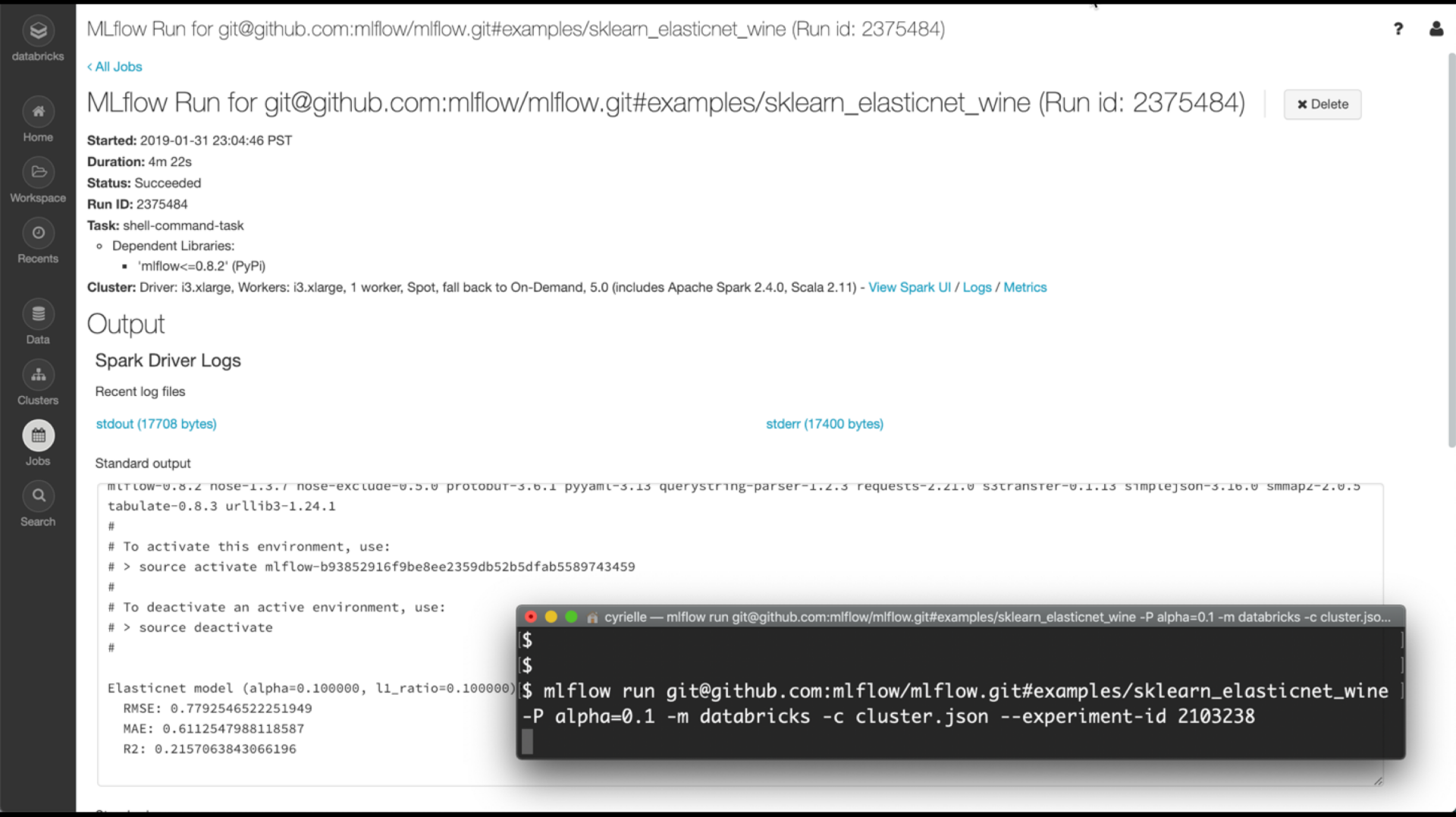
conda.yaml

model.pkl

Full Path: dbfs:/databricks/mlflow/2103238/1359afd3b6d341719f1ff4e2afcdedea/artifacts/ElasticNet-paths.png
Size: 45.04KB



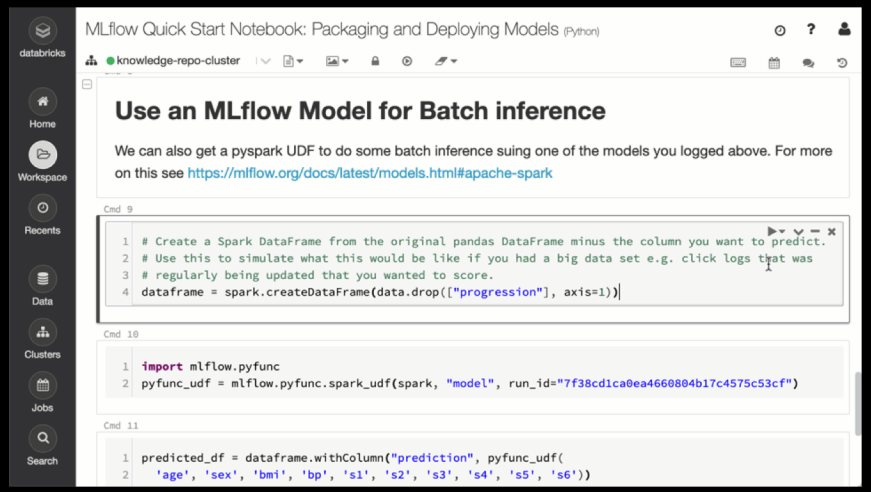
databricks


databricks
Home
Workspace
Recents
Data
Clusters
Jobs
Search
MLflow Run for git@github.com:mlflow/mlflow.git#examples/sklearn_elasticnet_wine (Run id: 2375484)
‹ All Jobs
MLflow Run for git@github.com:mlflow/mlflow.git#examples/sklearn_elasticnet_wine (Run id: 2375484)
Delete
Started: 2019-01-31 23:04:46 PST
Duration: 4m 22s
Status: Succeeded
Run ID: 2375484
Task: shell-command-task
Dependent Libraries:
'mlflow<=0.8.2' (PyPi)
Cluster: Driver: i3.xlarge, Workers: i3.xlarge, 1 worker, Spot, fall back to On-Demand, 5.0 (includes Apache Spark 2.4.0, Scala 2.11) - View Spark UI / Logs / Metrics
Output
Spark Driver Logs
Recent log files
stdout (17708 bytes)
stderr (17400 bytes)
Standard output
tabulate-0.8.3 urllib3-1.24.1
#
To activate this environment, use:
> source activate mlflow-b93852916f9be8ee2359db52b5dfab5589743459
#
To deactivate an active environment, use:
> source deactivate
#
Elasticnet model (alpha=0.100000, l1_ratio=0.100000):
RMSE: 0.7792546522251949
MAE: 0.6112547988118587
R2: 0.2157063843066196
cyrielle — mlflow run git@github.com:mlflow/mlflow.git#examples/sklearn_elasticnet_wine -P alpha=0.1 -m databricks -c cluster.jso...
$
$
$ mlflow run git@github.com:mlflow/mlflow.git#examples/sklearn_elasticnet_wine
-P alpha=0.1 -m databricks -c cluster.json --experiment-id 2103238

databricks

Model Deployment

Quickly deploy models to any platform based on your needs, locally or in the cloud, from experimentation to production.

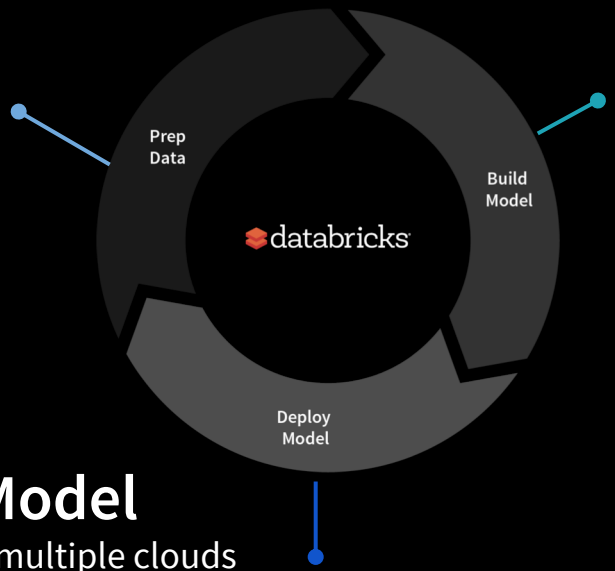


Managed MLflow Supports:

- Databricks Jobs and Clusters for Production Model Operations
- Batch inference on Databricks (Apache Spark)
- REST endpoints via Docker containers, Azure ML, or SageMaker

Standardizing the ML Lifecycle
on Databricks Unified Analytics Platform

Prep Data
Feed data to Models
Enrich data in experiments
Delta Lake

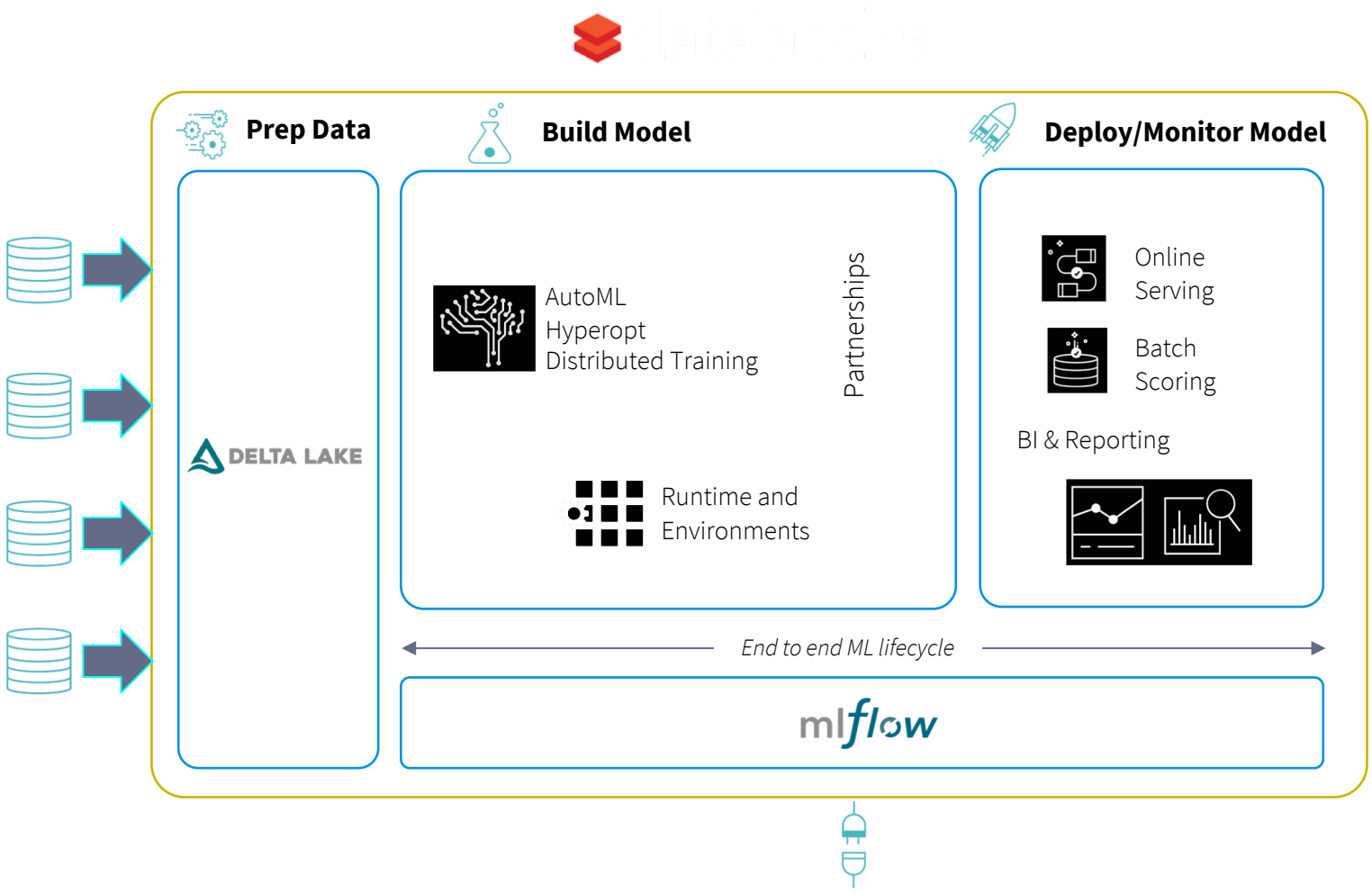


Build Model
Track Experiments
Reproduce experiments
Databricks Runtime for ML
MLflow Project & Tracking

Deploy Model
Integrate with multiple clouds
Manage and monitor models
MLflow Models

Unifying the End-to-end Data & ML Lifecycle

Overview



Part of ML/AI Governance:

- ML Lifecycle Management
- Model Management
- Experiment Management
- Model Serving
- Model Swap
- Explainability
- Bias
- Transparency
- Risk

Bias, Transparency, and Machine Learning and Artificial Intelligence Governance

THANK YOU!

bsharve@gmail.com; bsharve@gwu.edu;
benjamin.harvey@databricks.com

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. arXiv preprint arXiv:1908.09635.

B. S. Harvey and S. Ji, "Cloud-Scale Genomic Signals Processing for Robust Large-Scale Cancer Genomic Microarray Data Analysis," in IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 238-245, Jan. 2017.

Harvey, Benjamin.; Ji, S., "Cloud-Scale Genomic Signal Processing Classification Analysis for Gene Expression Microarray Data," *Engineering in Medicine & Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE* , vol., no., pp.7152,7155, 26-30 Au