databricks

How Databricks Improves Analyst Productivity

Secure Cloud Service

COST-EFFICIENT SCALE

Reliably process, store and analyze large populations of diverse patient data with an optimized version of Apache Spark and Delta Lake

REPRODUCIBILITY

Collaborative workspaces integrated with ML libraries and MLflow provide model tracking, management and revision histories

UNIFIED, SECURE DATA

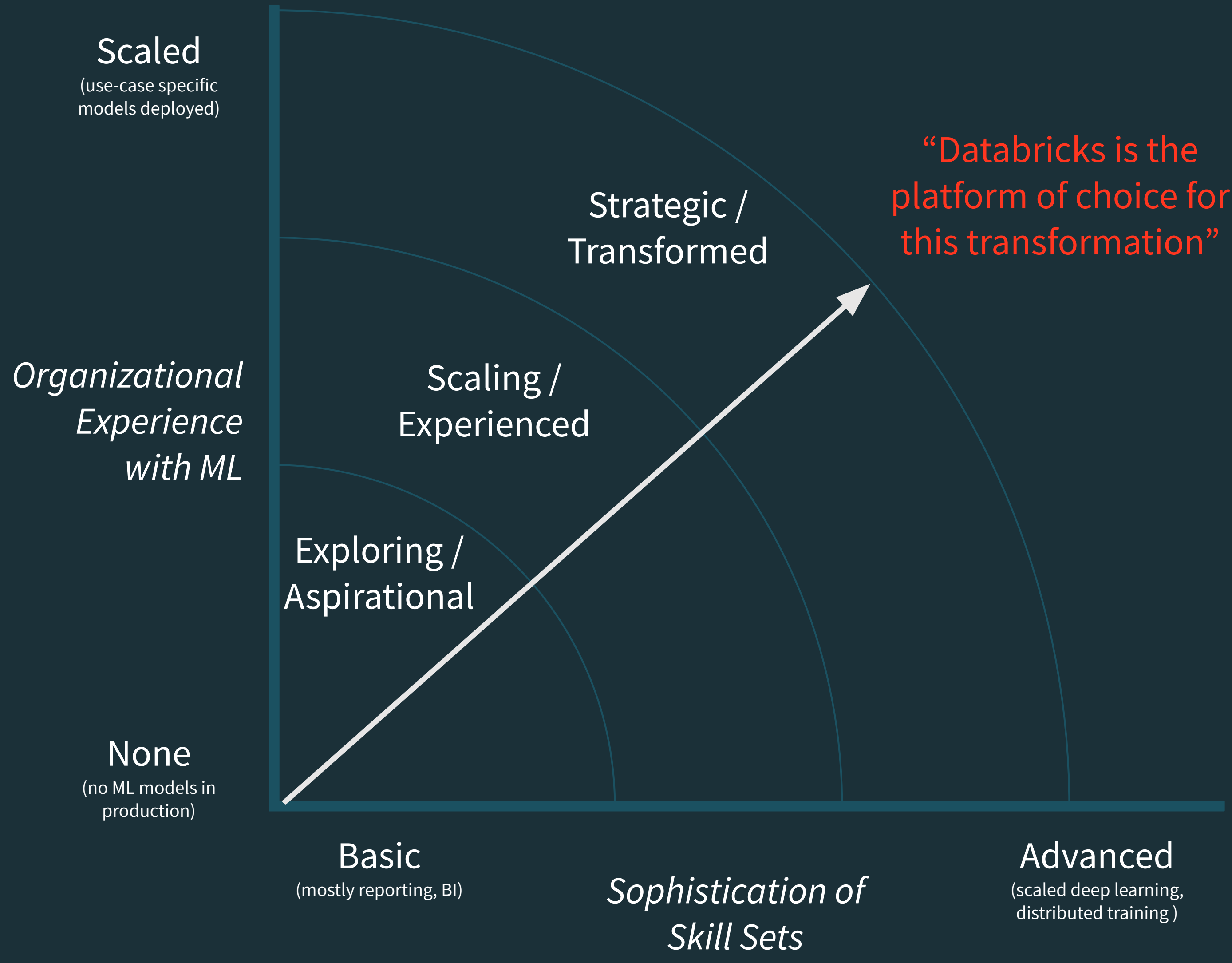
Managed platform with enterprise-grade security including data centric security, role-based control enables rapid and compliant data access

Build a Single View of All Your Data

Improve Collaboration

Power Analytics at Scale with Confidence

Stages of the Analytics Journey*



Scaled
(use-case specific models deployed)

Organizational Experience with ML

None
(no ML models in production)

Strategic / Transformed

Scaling / Experienced

Exploring / Aspirational

Basic
(mostly reporting, BI)

Sophistication of Skill Sets

Advanced
(scaled deep learning, distributed training)

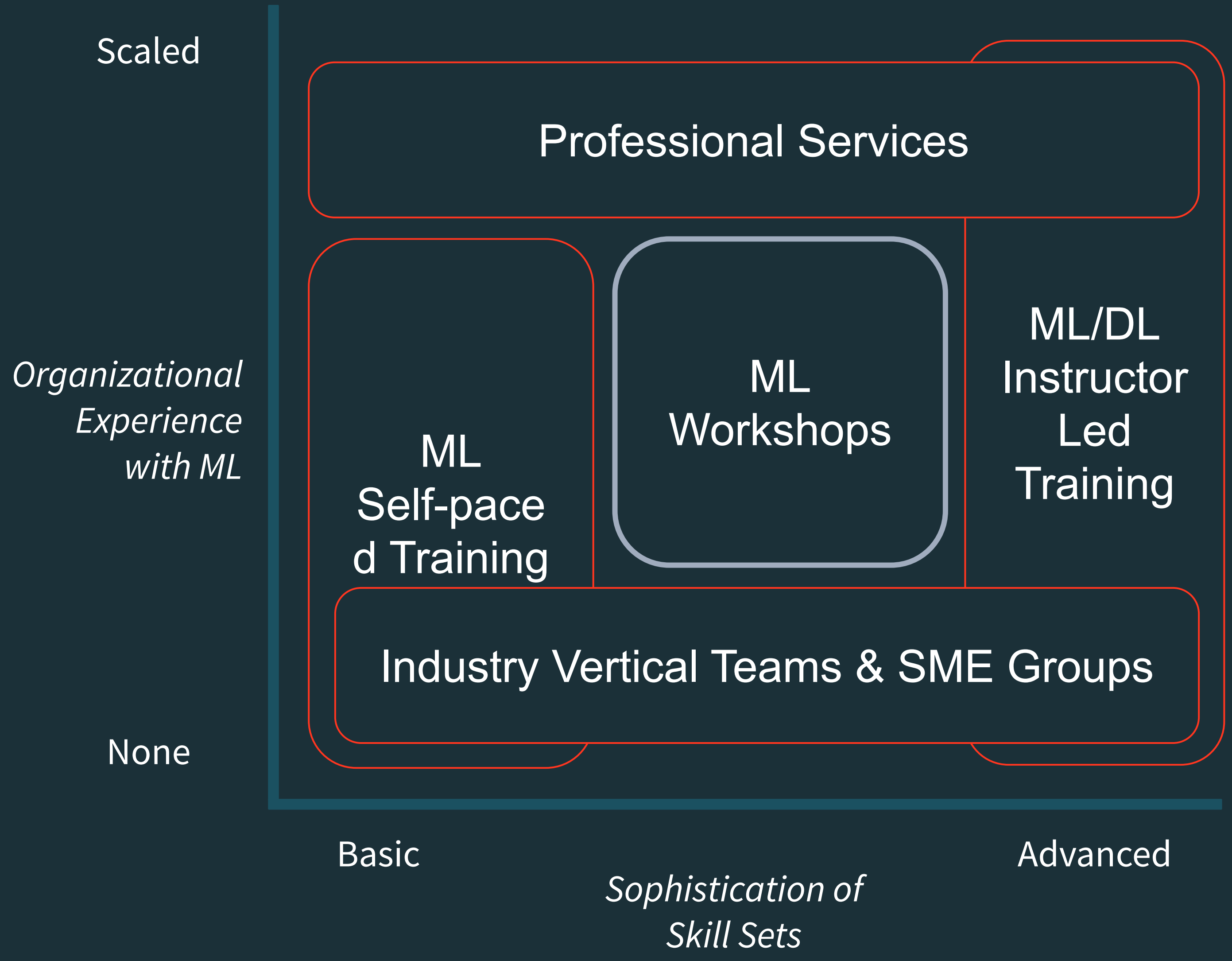
"Databricks is the platform of choice for this transformation"

*MIT Sloan Management Review

*Accenture Insights

*SAS

ML Workshops as a Targeted Offering


Scaled
Organizational Experience with ML
None
Professional Services
ML Self-paced Training
ML Workshops
ML/DL Instructor Led Training
Industry Vertical Teams & SME Groups
Basic
Advanced
Sophistication of Skill Sets

databricks

ML Workshop Topics

Preferred Topic Ideas:

- **Single-node** Data Science on Databricks + Visualization Libraries
- **MLflow** Machine Learning Lifecycle Management & Model Deployment
- **Koalas** Scaling Pandas with Spark
- **Environment Management** ML Runtime, Container Services, Conda, Git Projects
- **Parallelizing Machine Learning**
  - Parallelize Feature Engineering with **Spark**
  - Parallelize Hyperparameter Tuning with **HyperOpt**
  - Parallelize Single Model Training with **SparkML**
  - Train Many Models in Parallel with **Pandas UDFs**

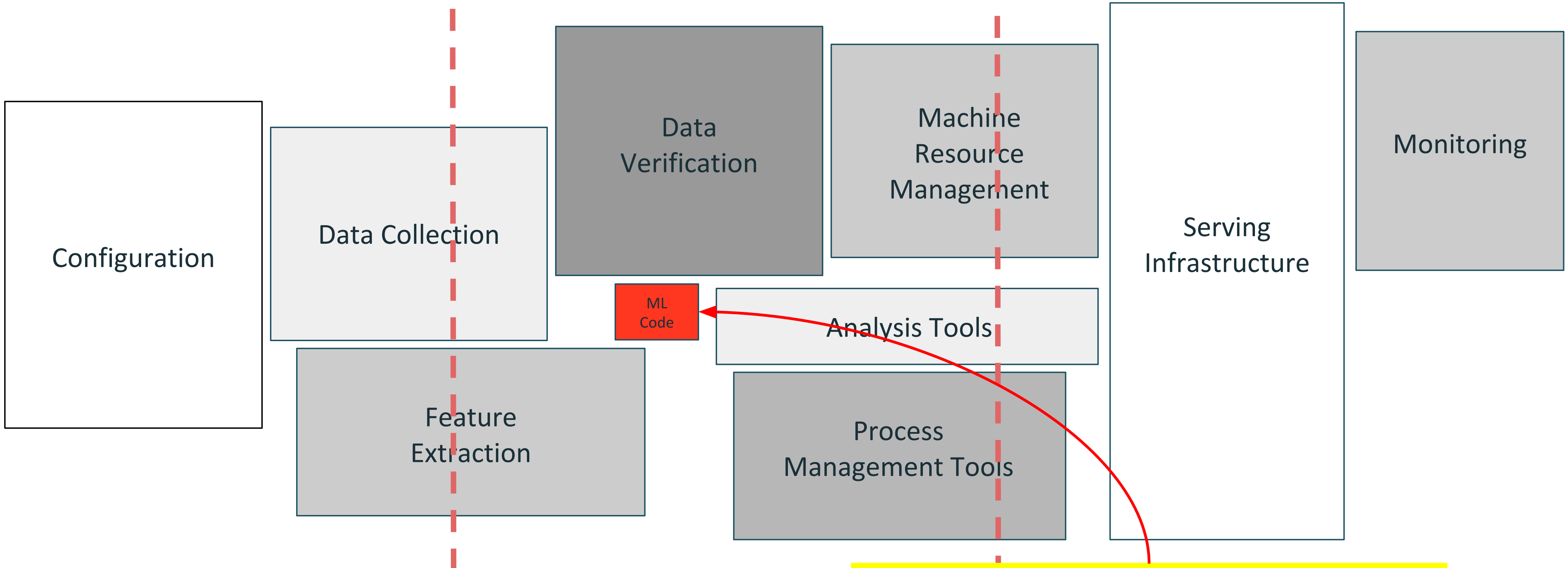
Topics served better with other offerings:

- Intro to ML/AI/DL → *Self-paced Training*
- Spark ML/MLlib → *Self-paced Training*
- Sklearn, Tensorflow, Keras, Pytorch → *Instructor-led Training*
- Deep Learning: NN’s/CNN’s, SGD, optimizers, activation functions → *Instructor-led Training*
- Horovod → *Instructor-led Training*
- LIME, SHAP → *Instructor-led Training*
- Reinforcement and Transfer Learning → *Instructor-led Training*
- Industry-specific ML use-cases: IoT, DBR for Genomics, Geospatial, etc. → *Industry Vertical Teams, SME teams*
- Performance Tuning, Cluster Optimization → *Instructor-led Training, Professional Services*
- Productionizing Models → *Professional Services*
- Governance, GDPR, CCPA → *Professional Services*

How do I manage my data for ML

How do I efficiently train models?

How do I productionize models?



Everyone uses the same libraries
(scikit-learn, xgboost, keras/tensorflow, etc.)

How do I manage my data for ML?

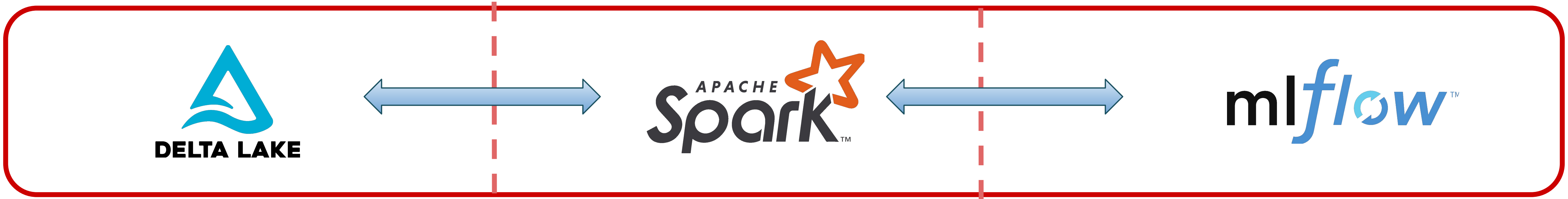
- Dataset isolation
- Binary files
- Batch + Streaming
- Schema Evolution/Enforcement
- Governance
- Regulatory Compliance
- ACID Transactions
- Efficient Upserts/Delete

How do I efficiently train models?

- Articulate tradeoff between…
  - Model Performance
  - Compute cost
  - Wall clock
- Take advantage of elastic compute
- Simple, consistent, scalable training environment
- Foster a culture of collaboration and experimentation
- Unlock advanced techniques

How do I productionize models?

- Which is the best model to deploy?
- How was it created?
- How do I hand it off to DevOps?
- How should it be deployed?
  - Batch vs Realtime
  - A/B test
- Who approved deployment?
- Make deployment easy
- Monitor/Alert/Debug



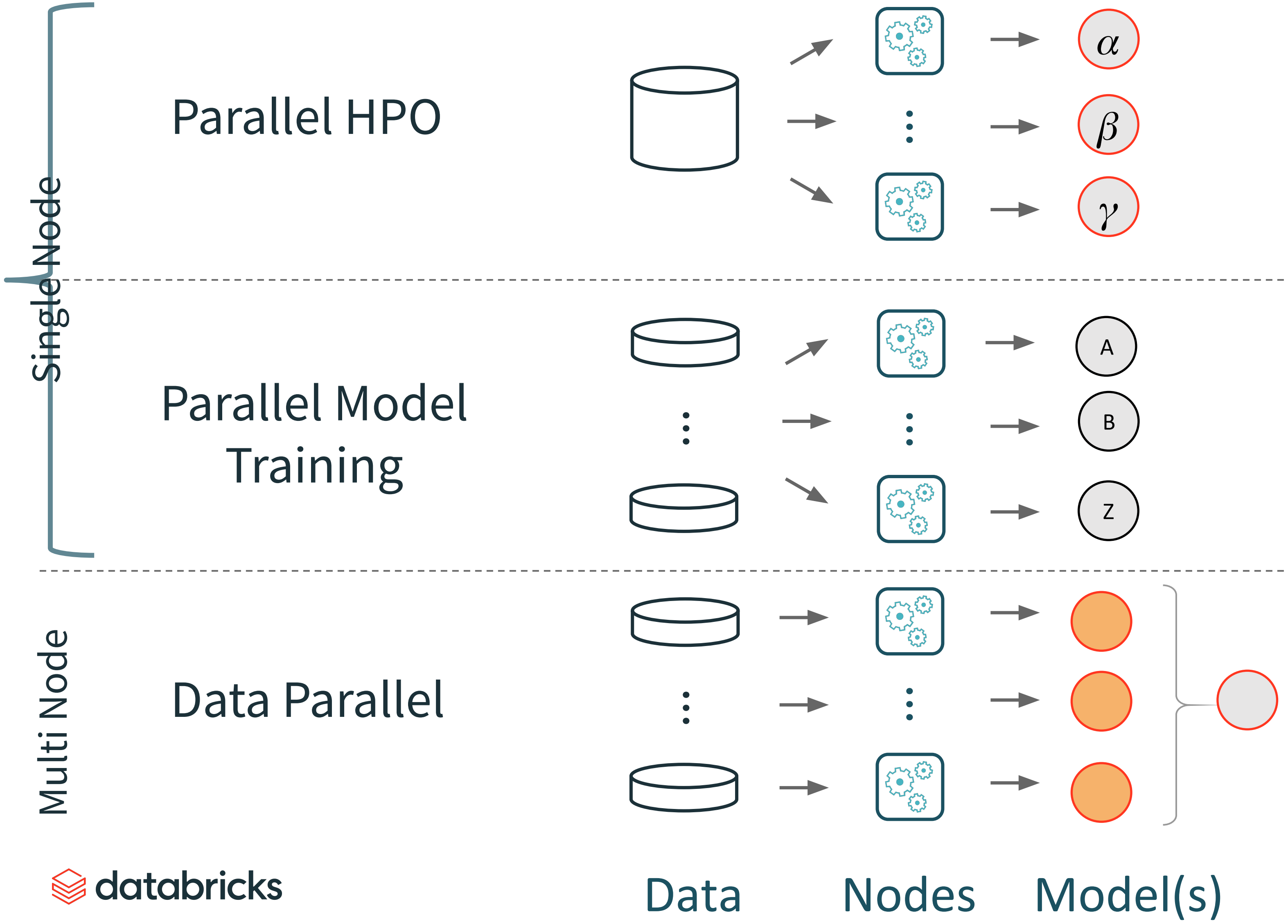
Delta: ML Ready Data Lake

Key Features

- ACID Transactions
- Schema Enforcement
- Binary File Support
- Unified Batch & Streaming
- Time Travel/Data Snapshots

- Guarantee data is “complete”, and won’t result in training failures
- Input types to model don’t change resulting in model failure
- Parquet to load many binary files at once and reduce training time (“small file problem”)
- One paradigm to score batch and streaming data
- Simplify Isolation and recovery of data set versions

Spark: Use Compute Efficiently



Single Node

Parallel HPO

α

β

γ

Parallel Model Training

A

B

Z

Multi Node

Data Parallel

Data

Nodes

Model(s)

Evaluate multiple HyperParameter configurations

- Hyperopt (with SparkTrials)
- Spark ML CV
- *Works with any ML Library*

Create multiple unique versions of a model in parallel

- pandas_udf
- *Works with any ML Library*

Train one model across multiple nodes

- Spark ML
- lightgbm, xgboost4j
- HorovodRunner

How do you design an experiment?

An **Experiment** is an evaluation of a model using a combination of controllable factors that affect the response

Experiments must be designed correctly using statistical methodology

An Experiment should be:

- Independent of other responses
- Controlled for variance and uncontrollable error
- Reproducible, especially between model candidates

Techniques include:

- Measuring Classifier Performance
- Hypothesis Testing

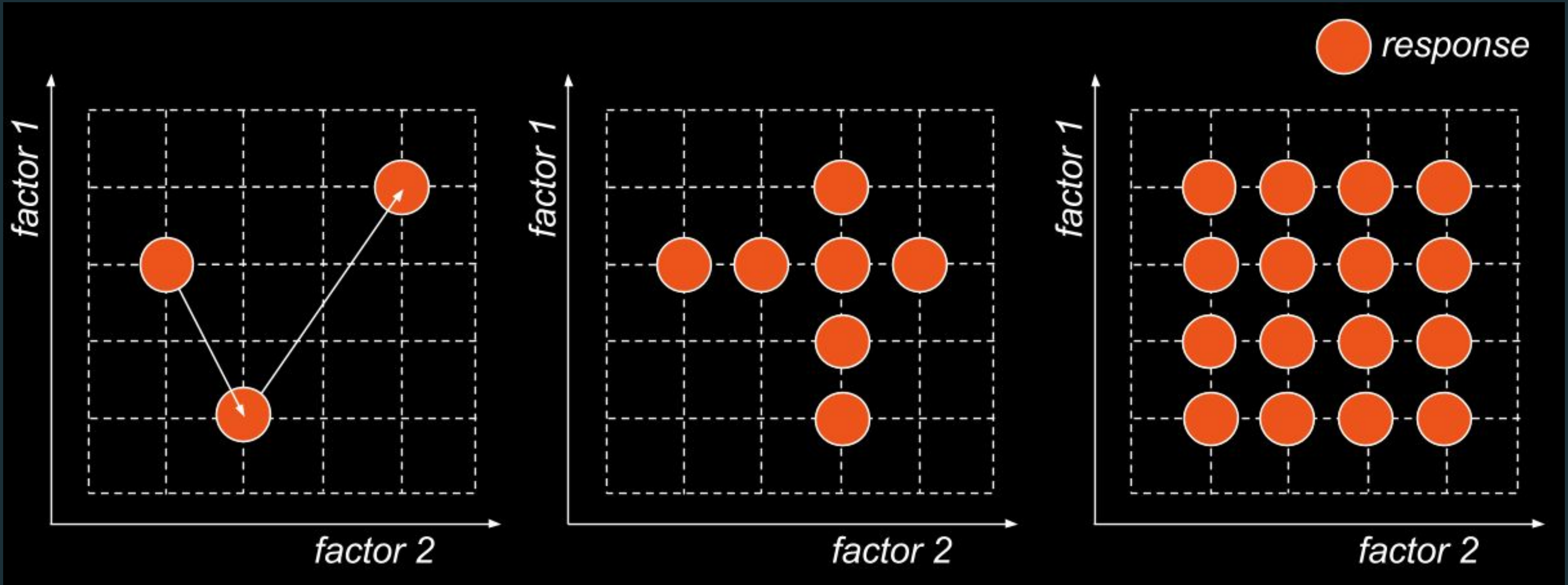
Factors that affect model outcomes

CONTROLLABLE

- Learning algorithm
- Input data
- Model parameters
- Model hyperparameters

UNCONTROLLABLE

- Noise in the data
- Optimization randomness
- Outcomes not observed during training but part of the system being modeled (I.e., a rare disease outcome)


response
factor 1
factor 2
factor 1
factor 2
factor 1
factor 2

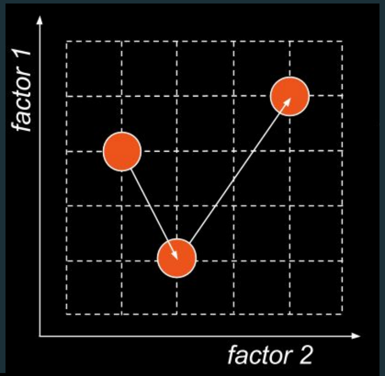
$O(LF)$ $O(LF)$ $O(L^F)$

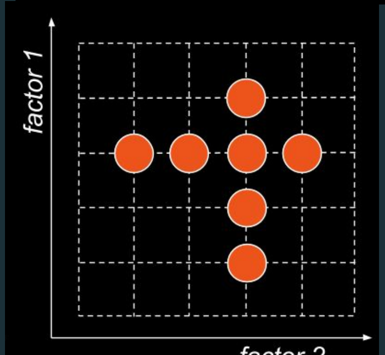
Generating Responses

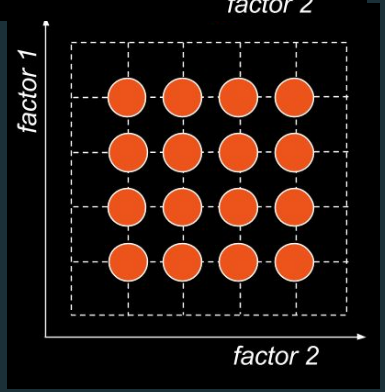
```
val p1 = new ParamMap().put(factor1.w(3), factor2.w(1))
val p2 = new ParamMap().put(factor1.w(1), factor2.w(2))
val p3 = new ParamMap().put(factor1.w(4), factor2.w(4))

val factorGrid = new ParamGridBuilder()
  .addGrid(factor1, Array(1, 2, 3, 4))
  .addGrid(factor2, Array(3))
  .build()

val factorGrid = new ParamGridBuilder()
  .addGrid(factor1, Array(1, 2, 3, 4))
  .addGrid(factor2, Array(1, 2, 3, 4))
  .build()
```







Generating Responses

Models support different methods / metrics

Train-Validation Split

```
val tvs =
 new TrainValidationSplit()
  .setEstimatorParamMaps(factorGrid)
  .setEvaluator(new RegressionEvaluator)
  .setTrainRatio(r)
```

```
val model = tvs.fit(data)
model.bestModel
     .extractParamMap
```

- Creates an estimator based on the parameter map or grid
- Randomly splits the input dataset into train and validation sets based on the training ratio r
- Uses evaluation metric on the validation set to select the best model

Generating Responses

Models support different methods / metrics

Cross Validator

```
val cv = new CrossValidator()
  .setEstimatorParamMaps(factorGrid)
  .setEvaluator(new
      BinaryClassificationEvaluator)
  .setNumFolds(k)
```

```
val model = cv.fit(data)
model.bestModel
     .extractParamMap
```

- Creates k non-overlapping randomly partitioned folds which are used as separate training and test datasets
- Controls for uncontrollable factors and variance
- The ‘bestModel’ contains the model with the highest average cross-validation
- Tracks the metrics for each param map evaluated

Reducing the Number of Responses

Computational Complexity Limits Practicality of Factorial Search

- Use a well-formulated conceptual model. This informs factor choice and reduces unnecessary model iterations
- Normalize factors where possible (i.e., factor is determined by input as-opposed to arbitrarily chosen by the modeler)
- If your data is large enough, you can split your dataset into multiple parts for use during cross-validation

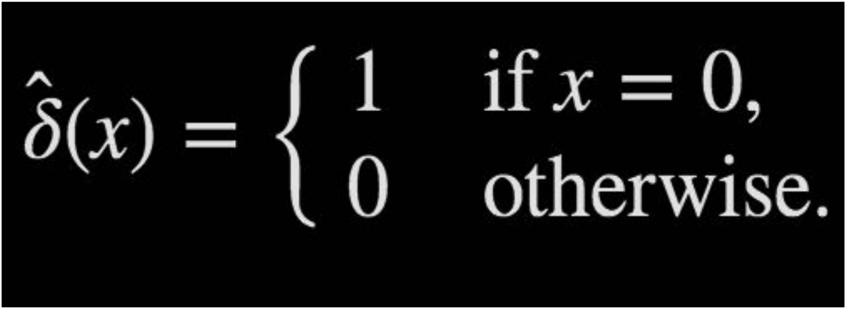
How do you analyze model output?

CLASSIFICATION

- Precision / recall relationships for binary classification problems
  - Receiver Operating Characteristic Curve
- For multi-classification problems:
  - Most packages only support 0/1 error functions
  - Confusion matrix
- For multilabel classification:
  - Again, 0/1 indicator function is only supported
  - Measures by label are most appropriate

REGRESSION

- Linear: RSME = Easy
- Non-linear: … *runs*
  - SoftMax
  - Cross Entropy



$$\hat{\delta}(x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Analyzing Model Output

Dataframe of (prediction, label)

```
val metrics = new
BinaryClassificationMetrics(predictionAndLabels.rdd.map( r =>
(r.getAs[Double]("prediction"), r.getAs[Double]("label"))))
```

Binary Classification

Metric	Spark Implementation
Receiver Operating Characteristic	roc
Area Under Receiver Operating Characteristic Curve	areaUnderROC
Area Under Precision-Recall Curve	areaUnderPR
Measures by Threshold	{measure}ByThreshold

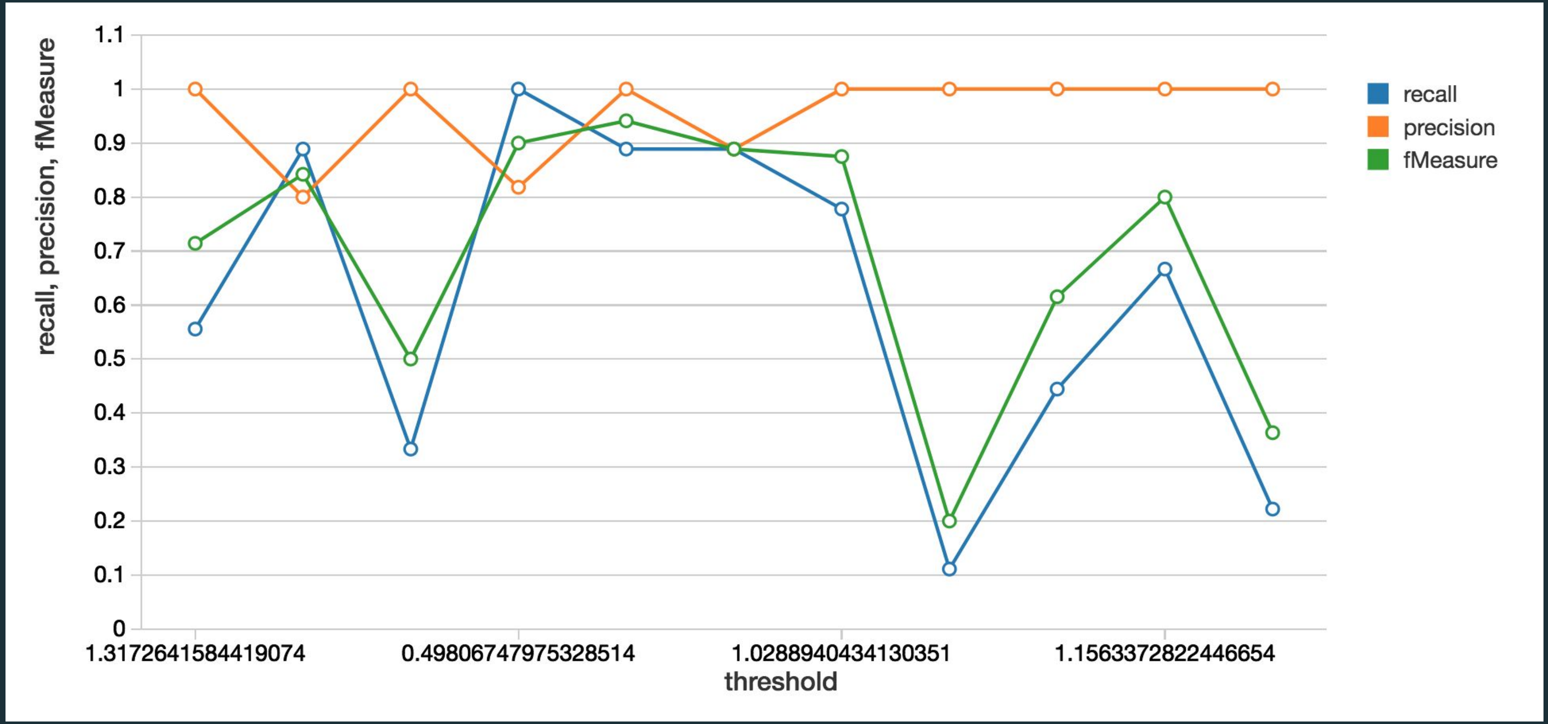
databricks

Threshold Curves

```
val threshold = metrics.{measure}ByThreshold
          .toDF('threshold', 'measure')

display(precision.join({measure}, 'threshold')
  .join(recall, 'threshold'))
```

Optional beta parameter for F1 measure (default = 1)



Analyzing Model Output

Dataframe of (prediction, label)

```
    val metrics = new
    MulticlassMetrics(predictionAndLabels.rdd.map( r =>
     (r.getAs[Double]("prediction"), r.getAs[Double]("label"))))
```

Multiclass Classification

Metric	Spark Implementation
Confusion Matrix	confusionMatrix
Accuracy	accuracy
Measures by Label	{measure}ByLabel
Weighted Measures	weighted{Measure}

Usually not a robust metric by itself

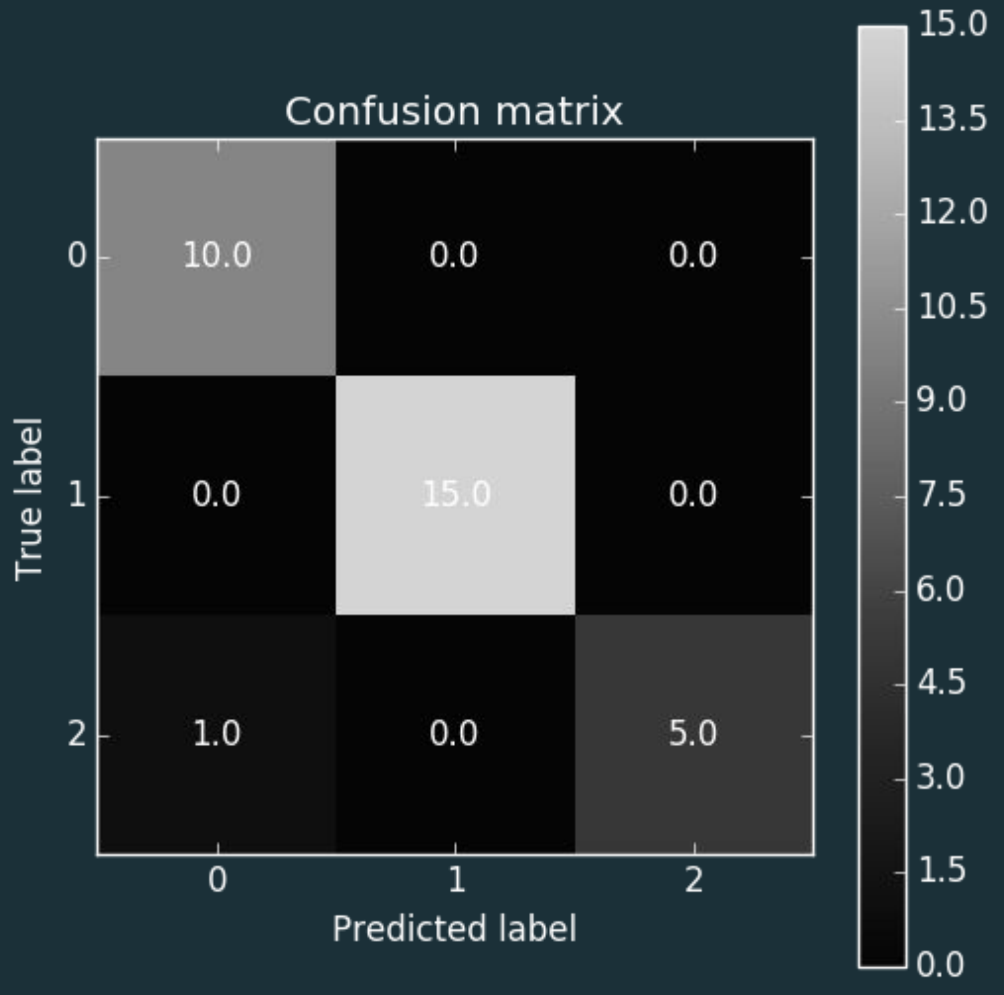
Confusion Matrix

```
metrics.confusionMatrix
       .toArray

//Display using non-Scala tool
%python
confusion = np.array([[...],[...]])
```

$$C_{ij} = \sum_{k=0}^{N-1} \hat{\delta}(\mathbf{y}_k - \ell_i) \cdot \hat{\delta}(\hat{\mathbf{y}}_k - \ell_j)$$

$$\begin{pmatrix} \sum_{k=0}^{N-1} \hat{\delta}(\mathbf{y}_k - \ell_1) \cdot \hat{\delta}(\hat{\mathbf{y}}_k - \ell_1) & \ldots & \sum_{k=0}^{N-1} \hat{\delta}(\mathbf{y}_k - \ell_1) \cdot \hat{\delta}(\hat{\mathbf{y}}_k - \ell_N) \\ \vdots & \ddots & \vdots \\ \sum_{k=0}^{N-1} \hat{\delta}(\mathbf{y}_k - \ell_N) \cdot \hat{\delta}(\hat{\mathbf{y}}_k - \ell_1) & \ldots & \sum_{k=0}^{N-1} \hat{\delta}(\mathbf{y}_k - \ell_N) \cdot \hat{\delta}(\hat{\mathbf{y}}_k - \ell_N) \end{pmatrix}$$



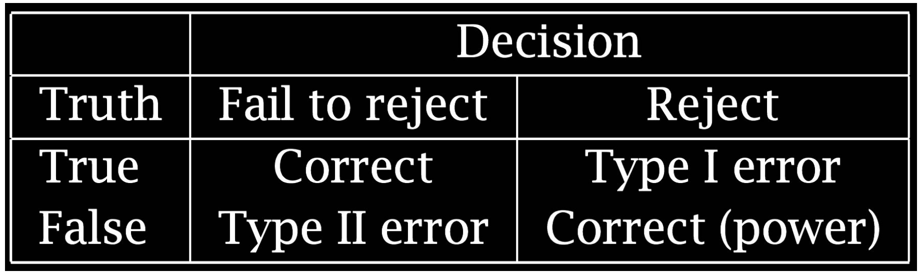
Why conduct and test a hypothesis?

How do I know my model is correct?

Hypothesis testing describes the significance of the result and provides a mechanism for assigning confidence to model selection

What is the likelihood that my model will make a misclassification error? This probability is not known!

Given two learning algorithms, which has the lower expected error rate?



	Decision	
Truth	Fail to reject	Reject
True	Correct	Type I error
False	Type II error	Correct (power)

1 - Binomial Test

1 - Approximate Normal Test

1 - t Test

2 - McNemar's Test

2 - K-Fold Cross-Validated Paired t Test

Chi-Squared Test

New in Spark 2.2!

Hypothesis: Outcomes are statistically independent

- Conducts Pearson's independence test for every feature against the label
- Chi-squared statistics is computed from (feature, label) pairs
- All label and feature values must be categorical

```
import org.apache.spark.mllib.stat.Statistics
import org.apache.spark.mllib.stat.test.ChiSqTestResult
val goodnessOfFitTestResult = Statistics.chiSqTest(labels)
val independenceTestResult = Statistics.chiSqTest(contingencyMatrix)
```

```
Chi squared test summary:
method: pearson
degrees of freedom = 4
statistic = 0.12499999999999999
pValue = 0.998126379239318
No presumption against null hypothesis: observed follows the same distribution as
expected..
```

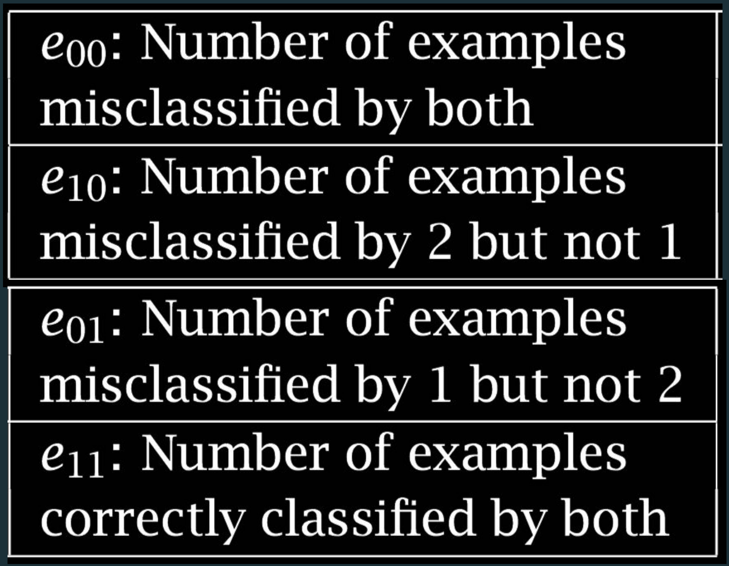
Nice, but you still need to do the hard work of constructing the hypothesis and validating!

McNemar’s Test

Hypothesis: Model 1 and model 2 have the same rate of generalization error

```
val totalObs = test.count
val conditions = "..."
val p1c =
predictions1.where(conditions).count()
val p1m = totalObs - p1c
val p2c =
predictions2.where(conditions).count()
val p2m = totalObs - p2c

val e00 = p1m + p2m
val e01 = p1m
val e10 = p2m
val e11 = p1c + p2c
```

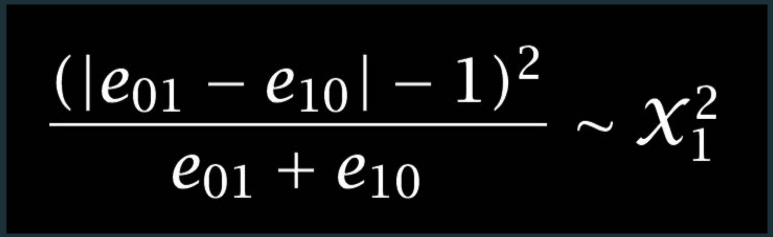


e_{00}: Number of examples misclassified by both

e_{10}: Number of examples misclassified by 2 but not 1

e_{01}: Number of examples misclassified by 1 but not 2

e_{11}: Number of examples correctly classified by both



$$\frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \sim \chi_1^2$$

databricks

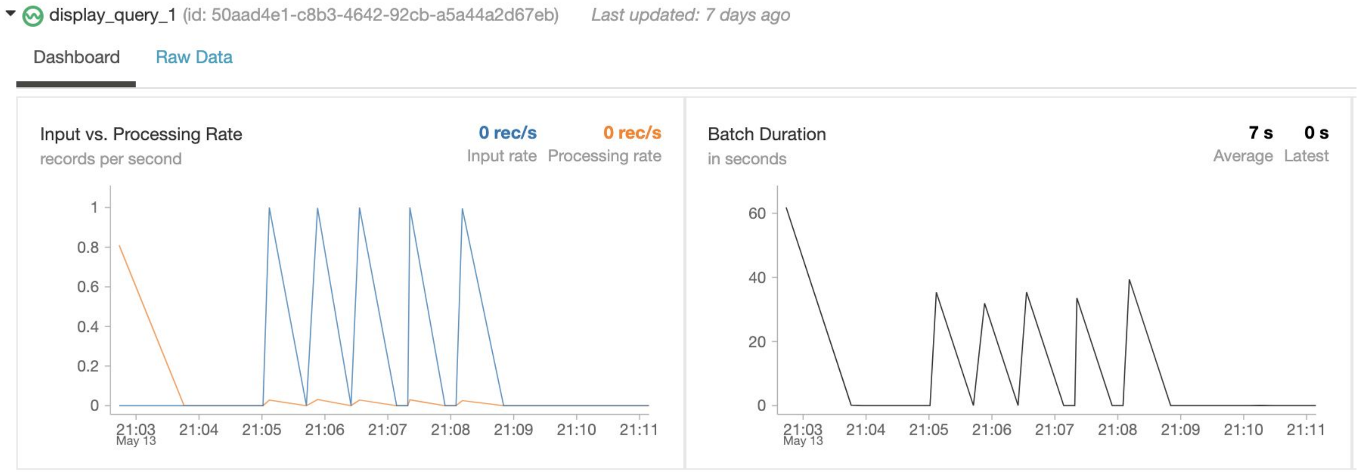
Analyzing of Variance

Analysis of Variance is used to compare multiple models. What is the statistical significance of running model 1 or model 2?

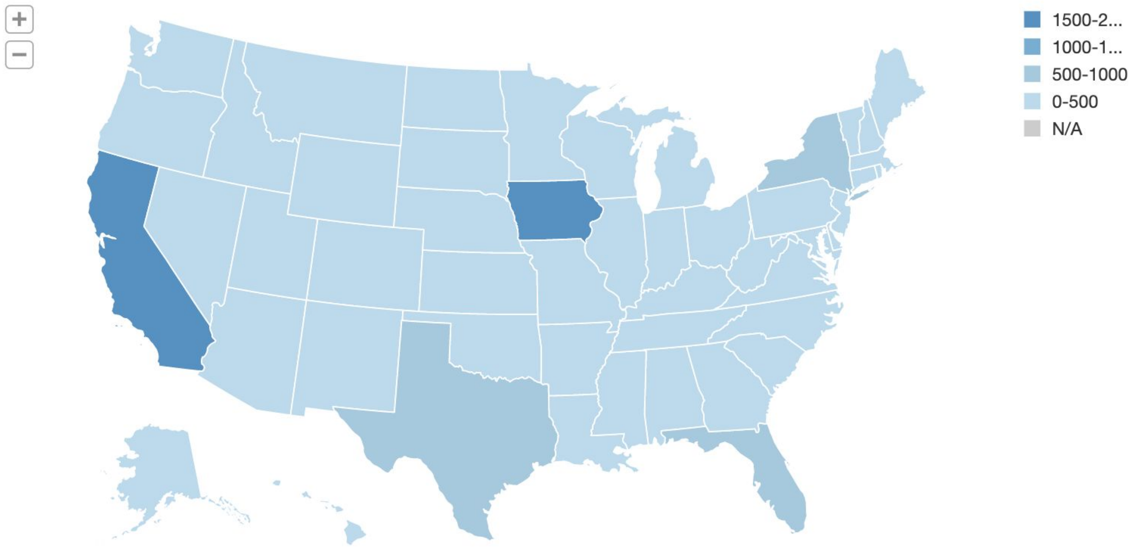
- Currently no techniques for ANOVA directly within MLlib
- Requires calculating statistics manually
- A very useful technique in-practice despite the manual work needed

DELTA LAKE

Demo - Focusing on Delta Lake Features



display_query_1 (id: 50aad4e1-c8b3-4642-92cb-a5a44a2d67eb) *Last updated: 7 days ago*



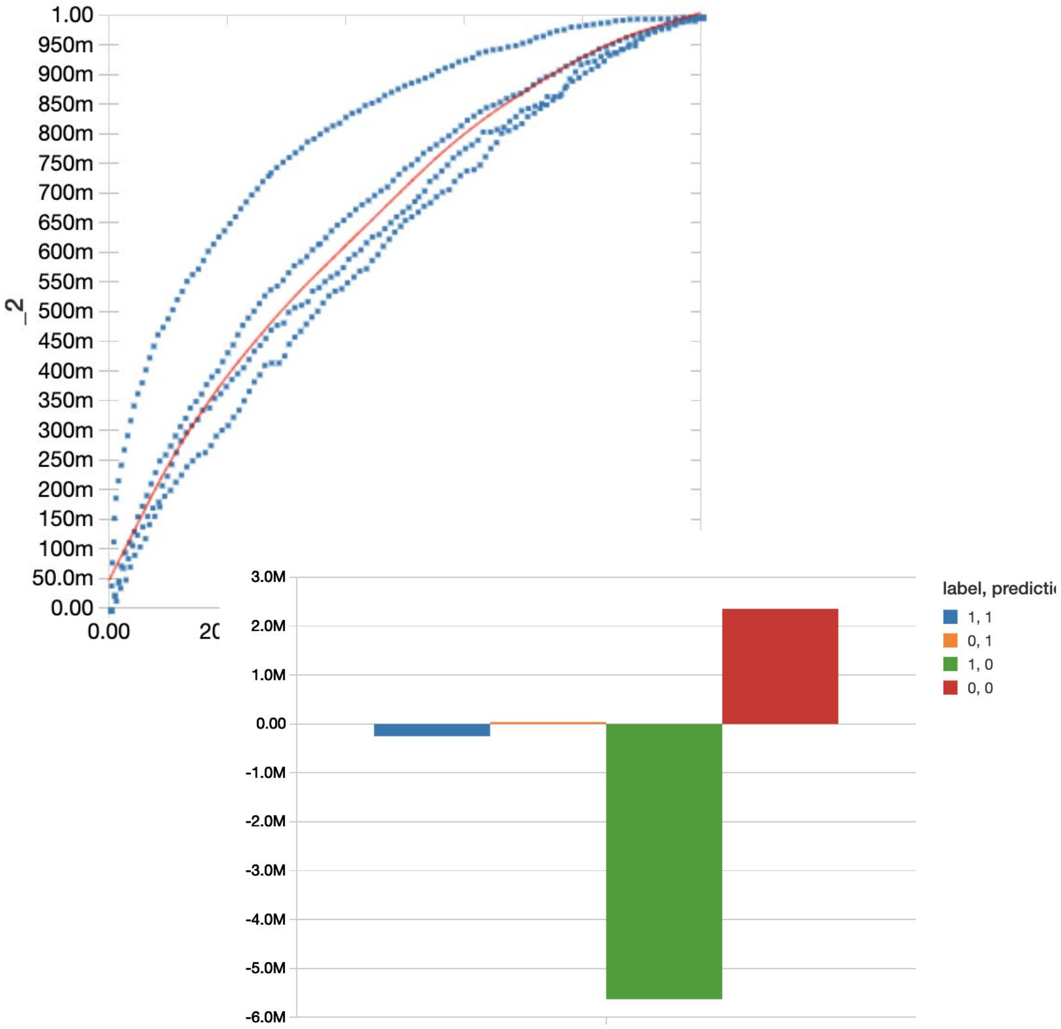
- Runs on DBCE
- Features:
  - Convert Parquet to Delta
  - Batch and Streaming Sync
  - Describe detail
  - Describe History
  - Time Travel
  - DDL
  - Schema Modification
- To do:
  - ACID Tx (leverage notebook)
  - Scalable Metadata

https://dbricks.co/dlw-01

databricks

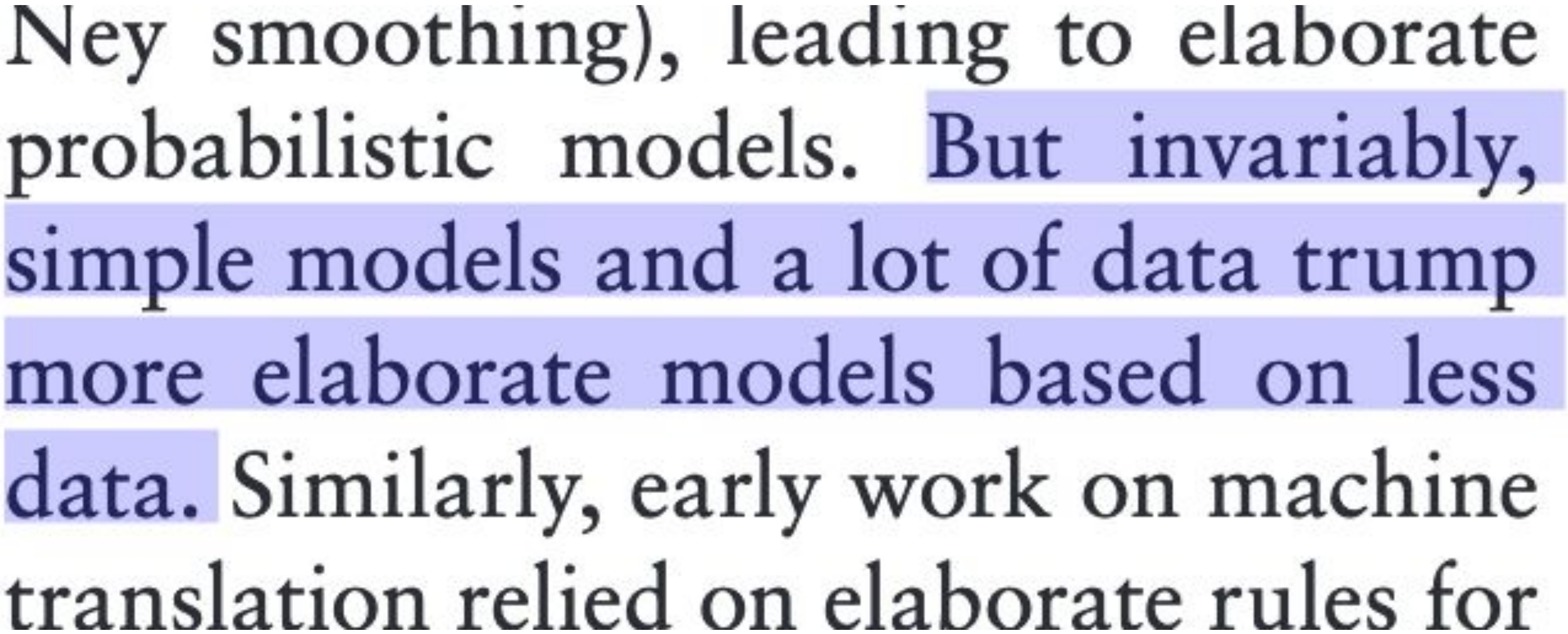
DELTA LAKE Demo - Expanding to ML

https://dbricks.co/dlw-02
https://dbricks.co/dlw-03



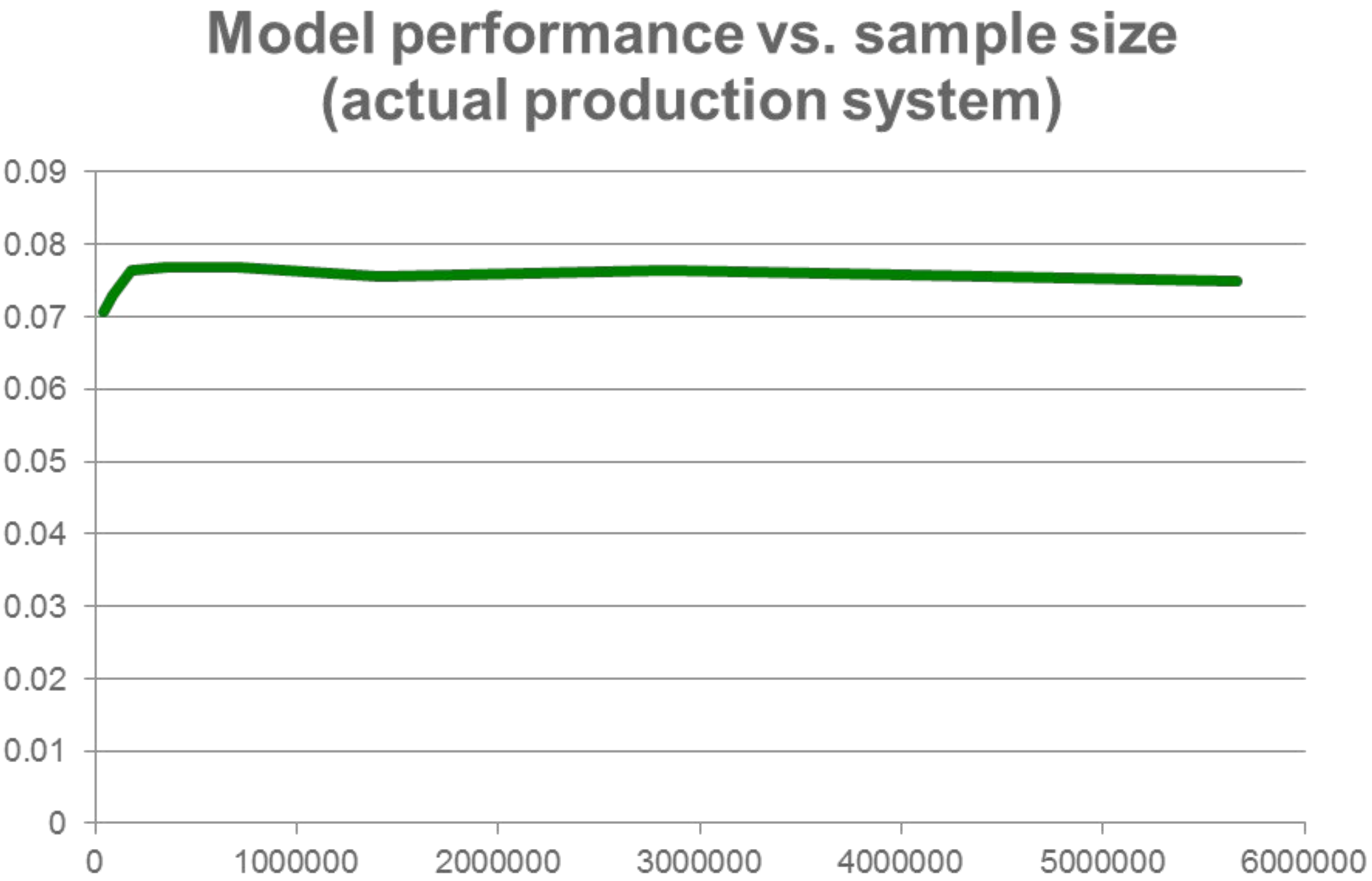
- Runs on DBCE
- Features:
  - Using Delta Table
  - Databricks Visualizations
  - Time Travel
  - Ad-hoc Analysis
  - Logistic Regression
  - Integrate MLflow
-

databricks



Ney smoothing), leading to elaborate probabilistic models. But invariably, simple models and a lot of data trump more elaborate models based on less data. Similarly, early work on machine translation relied on elaborate rules for

The Unreasonable Effectiveness of Data, *Alon Halevy, Peter Norvig, and Fernando Pereira, Google 2009*



Netflix recommendations: beyond the 5 stars, Xavier Amatriain *(Netflix)*

databricks

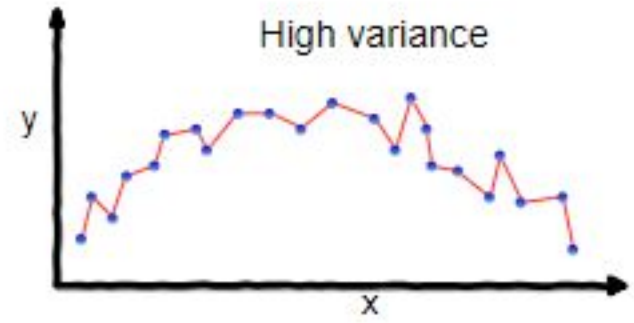
Oh, and what about the size of those data sets?

- 1 billion word corpus = ~2GB
- Netflix prize data = 700Mb compressed
  - 1.5 GB uncompressed (source)

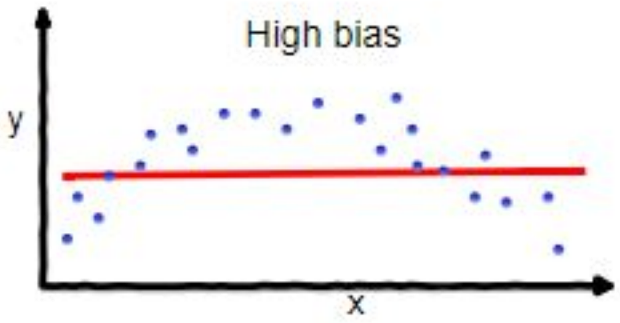
Conceptualizing a Distributed Model

- Model capabilities are different between serial and distributed applications
  - Some algorithms do not have a distributed implementation
  - Understanding computational complexity becomes an increasingly present limitation
  - Solver and optimizer implementations for existing algorithms may be different or not supported
  - Model assumptions may change when migrating a serial model to a distributed model
- Data characteristics are more challenging to reveal
  - Outliers are prevalent but may be incorrectly or poorly modeled
  - Missing value compensation can significantly skew results
  - Synthetic data can poorly represent actual system

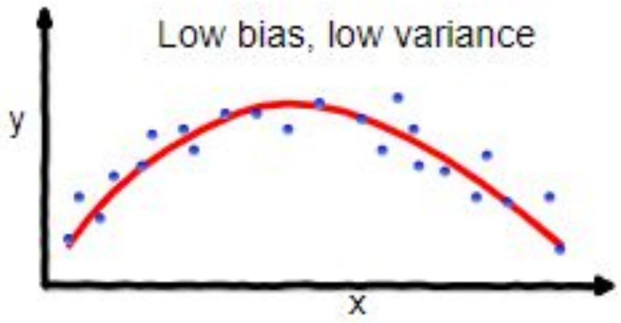
So where does that leave us?



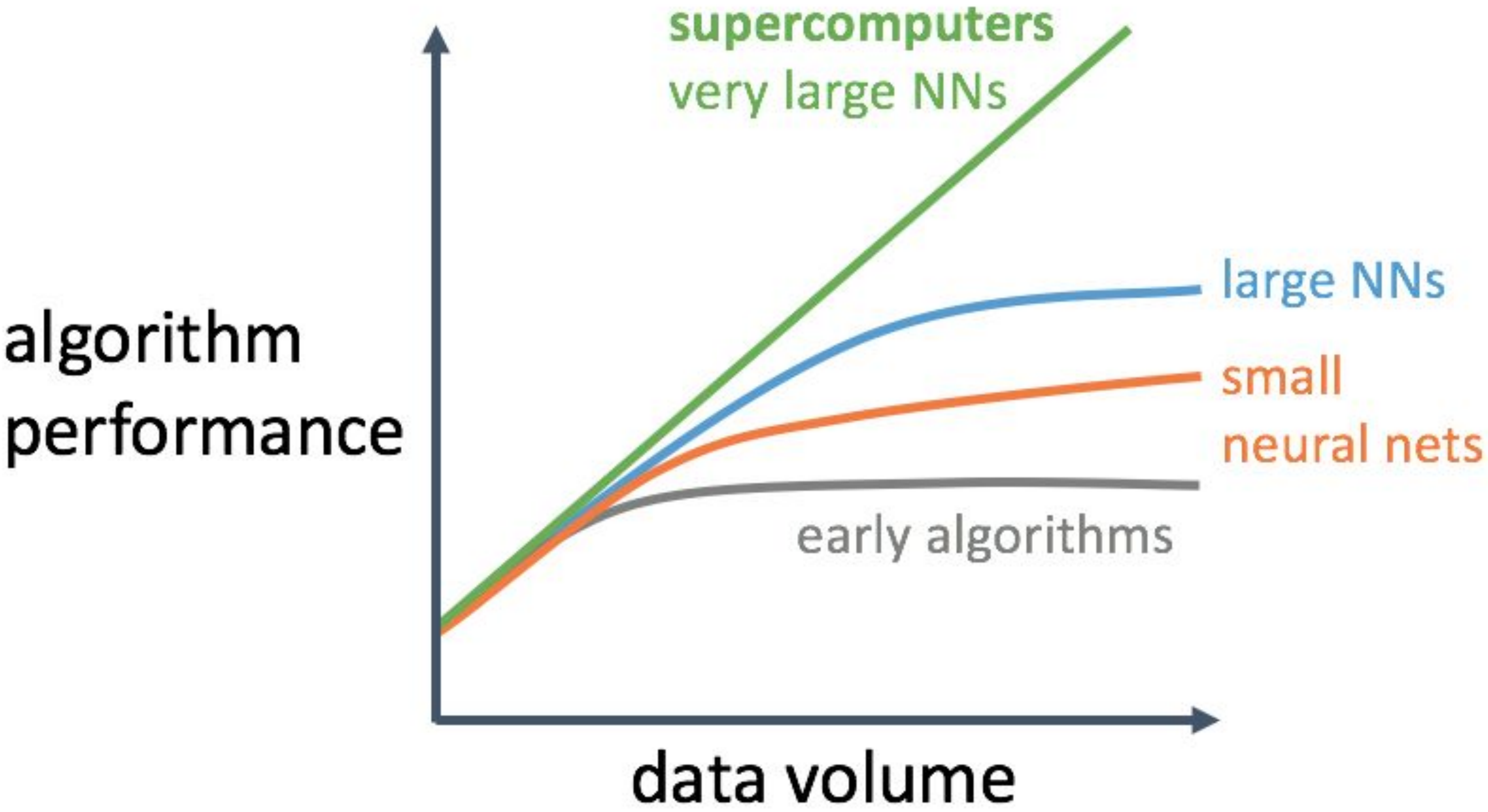
overfitting



underfitting



Good balance



Andrew Ng, AI is the New Electricity

Conclusion: more data makes sense for high variance (semi-structured or unstructured) problem domains like text and images. Sampling makes sense for high bias domains such as structured problem domains.

Should we always use more data with deep learning?

No! Transfer learning on smaller data often beats training nets from scratch on larger data-sets.

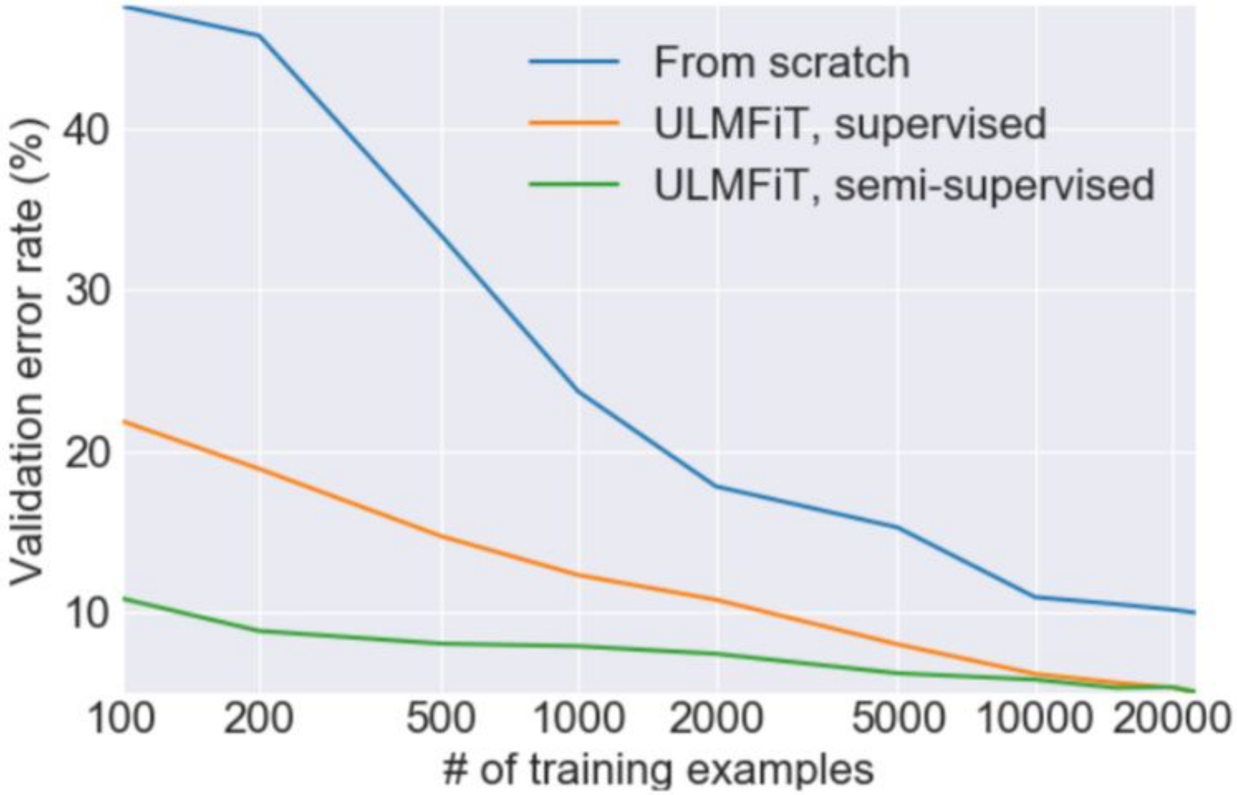
Open AI pointed out that while the amount of compute has been a key component of AI progress, “Massive compute is certainly not a requirement to produce important results.” (source)

In a benchmark run by our very own Matei Zaharia at Stanford, Fast.ai was able to win both fastest and cheapest image classification:

Imagenet competition, our results were:

- Fastest on publicly available infrastructure, fastest on GPUs, and fastest on a single machine (and faster than Intel’s entry that used a cluster of 128 machines!)
- Lowest actual cost (although DAWNBench’s official results didn’t use our actual cost, as discussed below).Overall, our findings were:
- **Algorithmic creativity is more important than bare-metal performance**

(source)



[Introducing state of the art text classification with universal language models,](#)

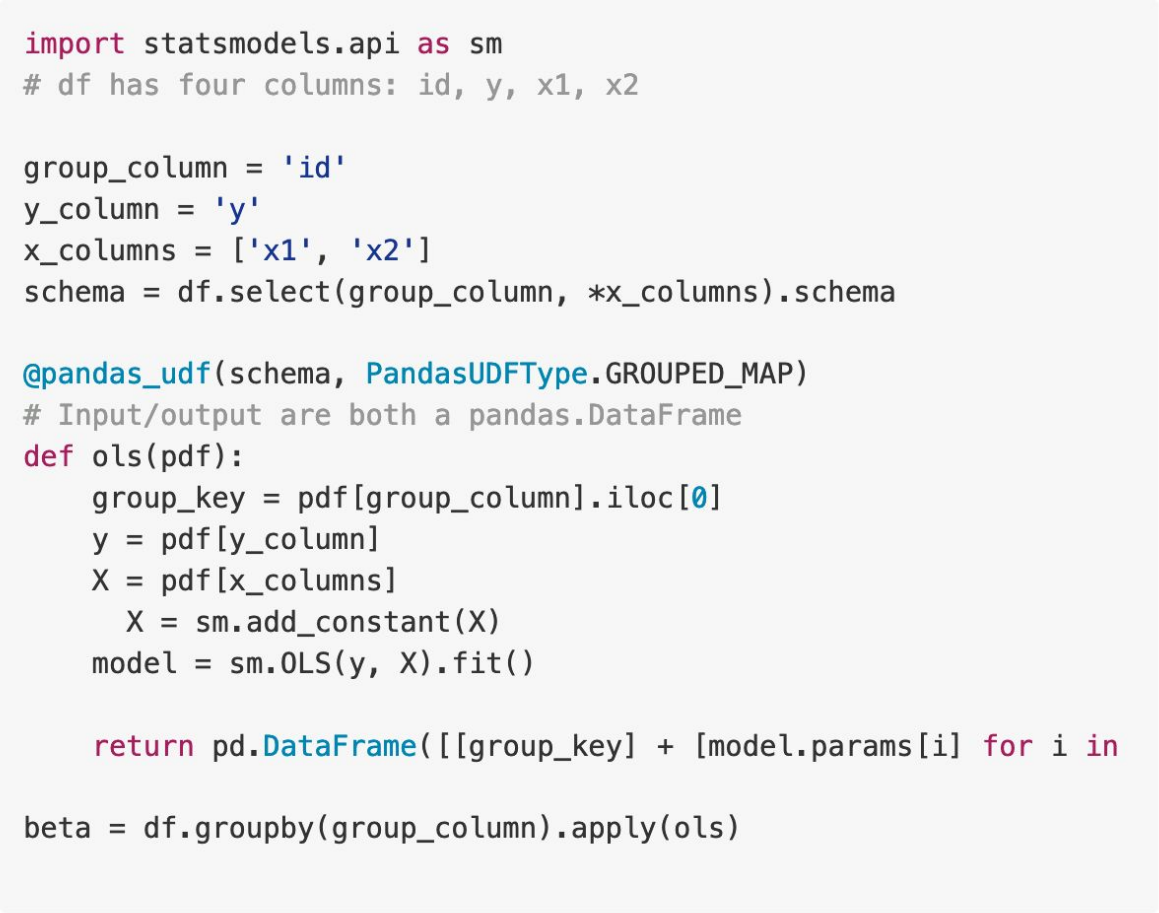
[Jeremy Howard and Sebastian Ruder](#)

Take-away: Even in the case of deep learning, if an established model exists, it’s better to use transfer learning on small data then train from scratch on larger data

So where does databricks fit into this story?

Training models (including hyperparameter search and cross validation) is embarrassingly parallel

Shift from distributed data to distributed models



```
import statsmodels.api as sm
# df has four columns: id, y, x1, x2

group_column = 'id'
y_column = 'y'
x_columns = ['x1', 'x2']
schema = df.select(group_column, *x_columns).schema

@pandas_udf(schema, PandasUDFType.GROUPED_MAP)
# Input/output are both a pandas.DataFrame
def ols(pdf):
    group_key = pdf[group_column].iloc[0]
    y = pdf[y_column]
    X = pdf[x_columns]
      X = sm.add_constant(X)
    model = sm.OLS(y, X).fit()

    return pd.DataFrame([[group_key] + [model.params[i] for i in

beta = df.groupby(group_column).apply(ols)
```

Introducing Pandas UDF for PySpark: How to run your native Python code with PySpark, fast.

The goal of experimentation is to understand the effect of model factors and obtain conclusions which we can consider statistically significant

This is challenging for distributed learning!

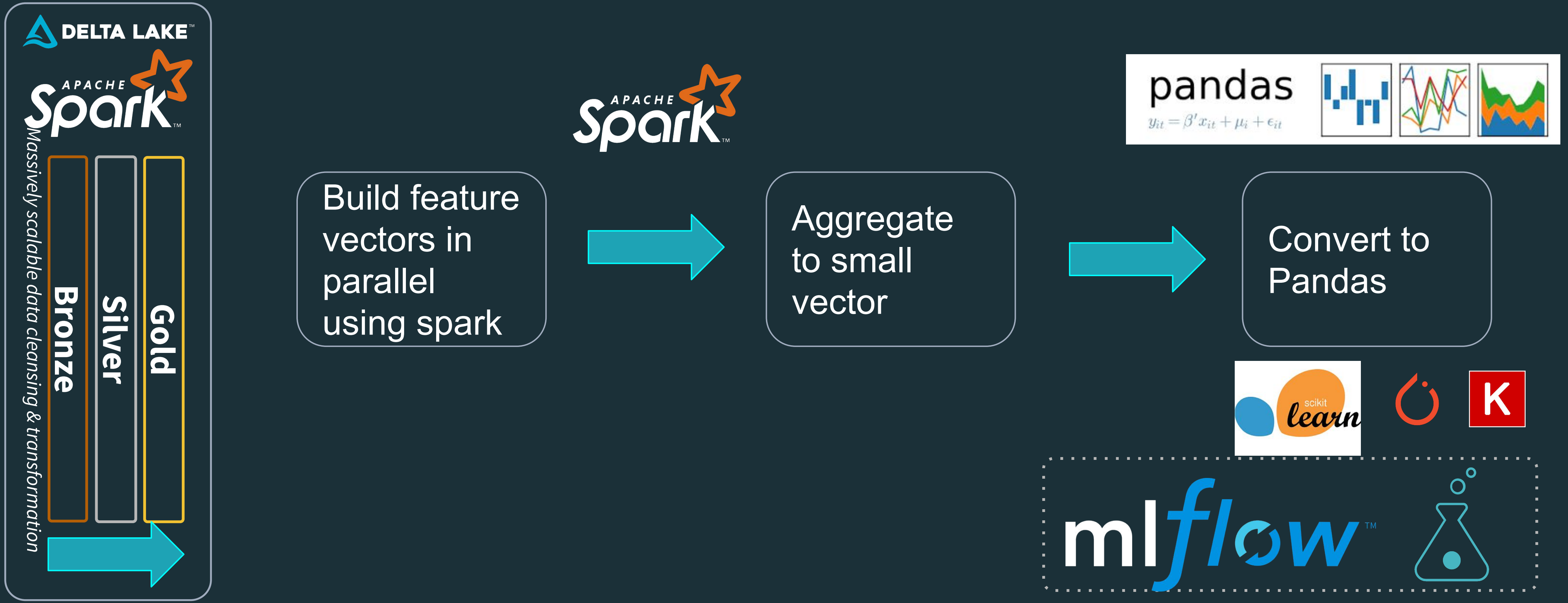
Data Scientists spend lots of time setting up their environment

4 ways to parallelise ML:

1. Parallelise Feature Engineering
2. Parallelise Hyperparameter Tuning
3. Parallelise Single Model Training
4. Train lots of models in parallel

4 ways to parallelise ML:

1 Parallelise Feature Engineering



Massively scalable data cleansing & transformation

Bronze

Silver

Gold

Build feature vectors in parallel using spark

Aggregate to small vector

Convert to Pandas

4 ways to parallelise ML:

2 Parallelise Hyperparameter Tuning

Hyper-parameter tuning methods

Non-adaptive methods:

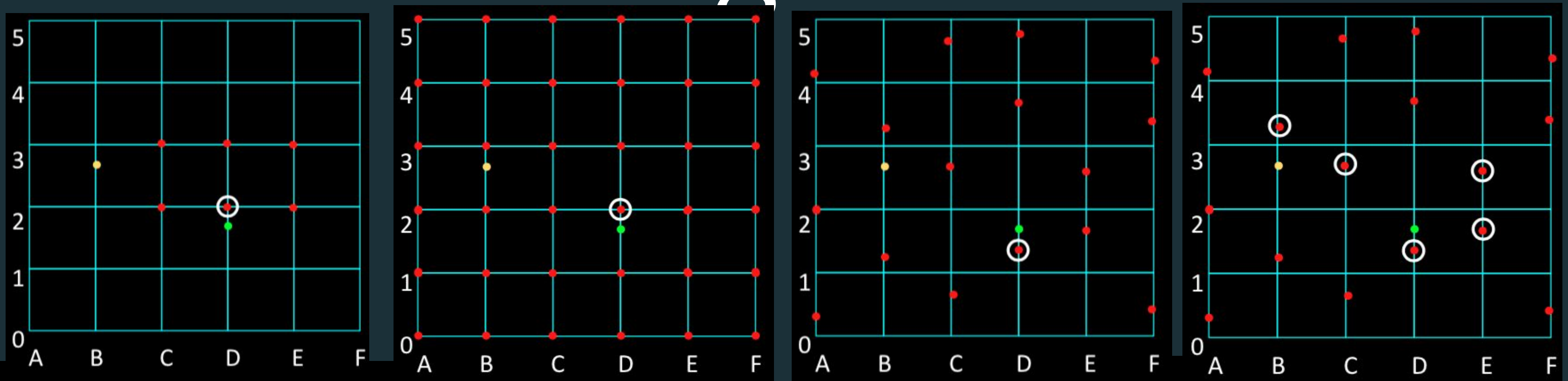
- Manual - a.k.a “baby-sitting”
- Grid search - brute force, exponential in the number of parameters
- Random search - less brutal, but not adaptive

Adaptive methods:

- Population based - genetic methods, Databricks AutoML
- Bayesian optimisation - uses an explicit model, normally linear

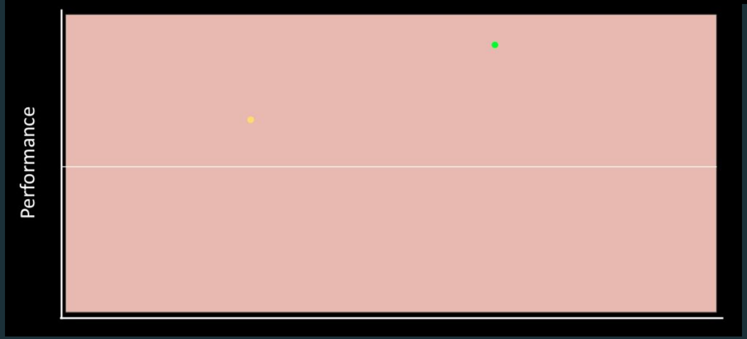
Hyper parameter tuning overview

- Manual
- Grid
- Random
- Population

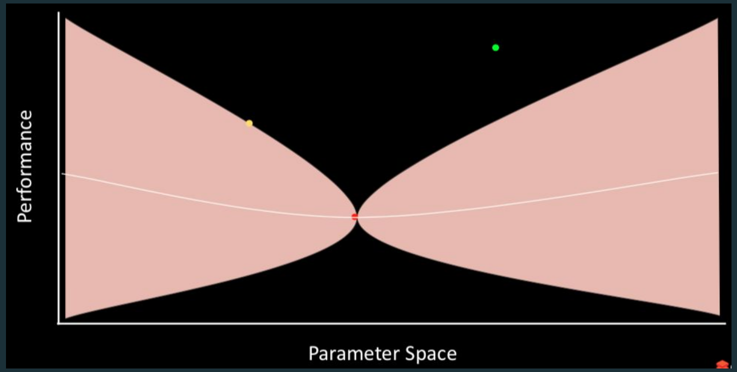


- Bayesian

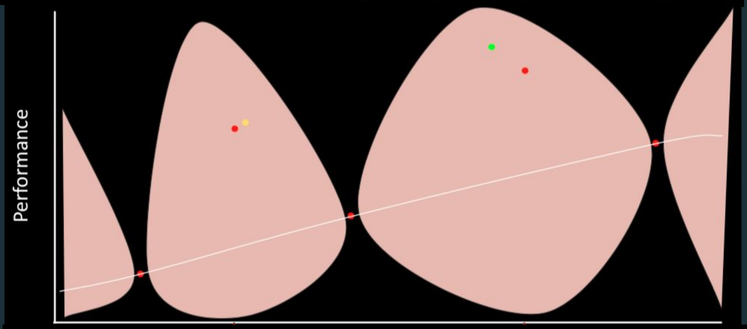
1.



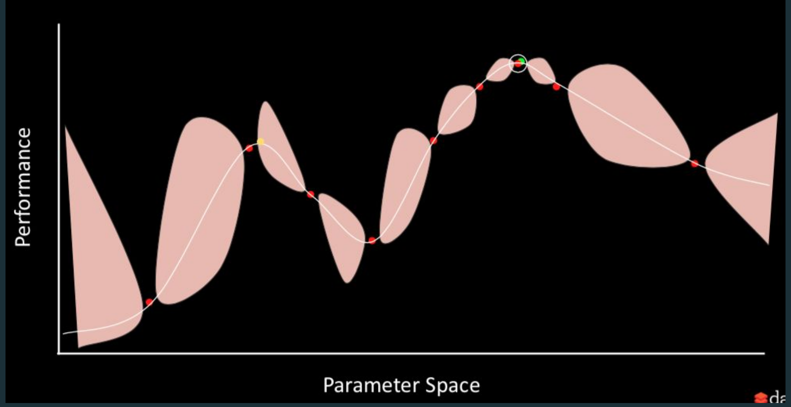
2.



3.



4.

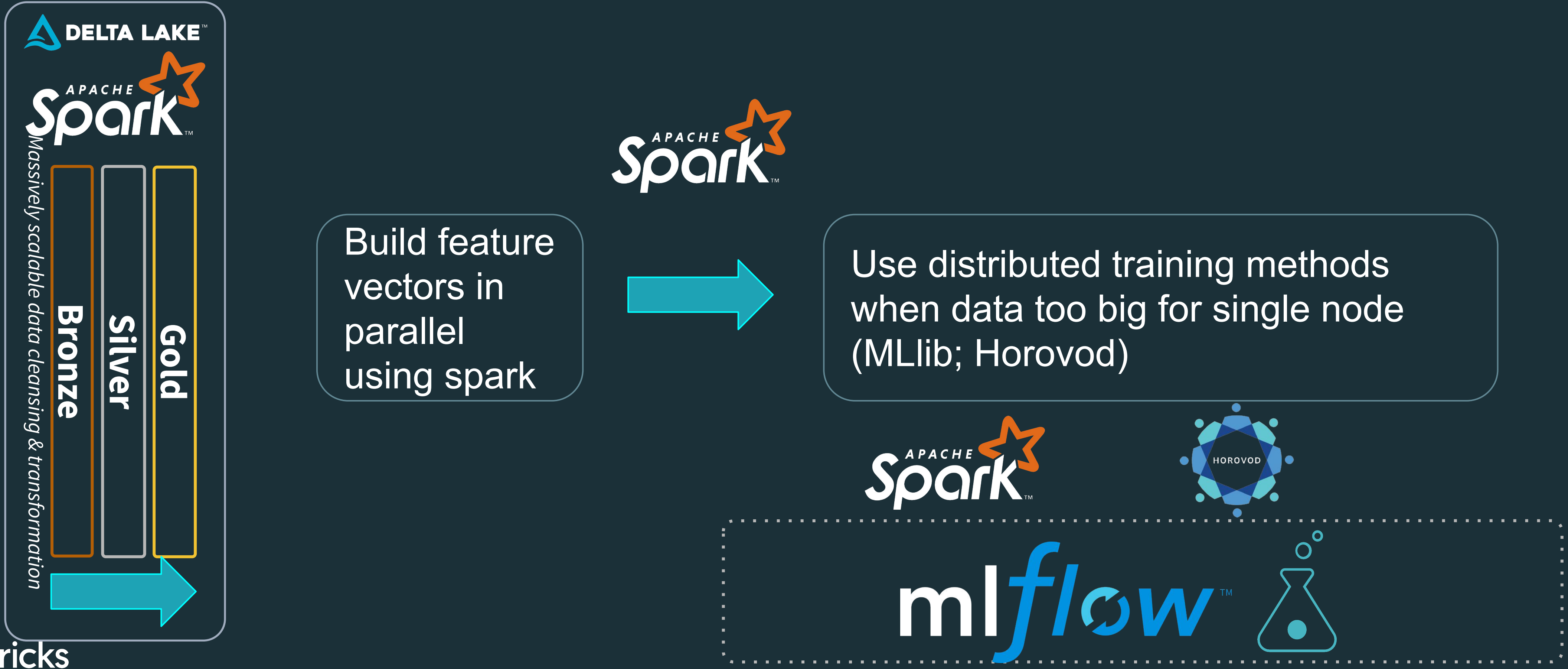


4 ways to parallelise ML:

3 Parallelise Single Model Training

How to parallelise ML:

3 Parallelise Single Model Training



DELTA LAKE

APACHE Spark

Massively scalable data cleansing & transformation

Bronze

Silver

Gold

Build feature vectors in parallel using spark

APACHE Spark

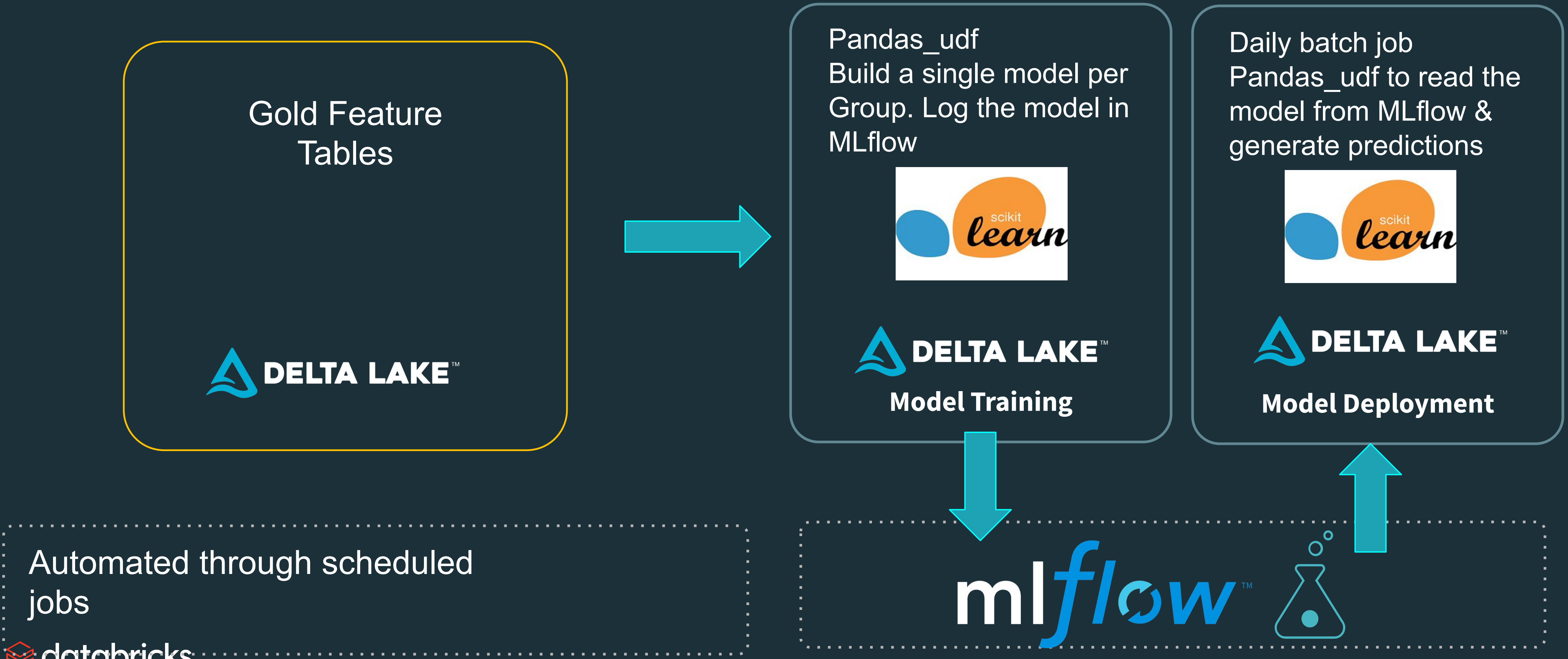
Use distributed training methods when data too big for single node (MLlib; Horovod)

databricks

4 ways to parallelise ML:

4 Train lots of models in parallel

Different model for each group



Gold Feature Tables

DELTA LAKE

Pandas_udf
Build a single model per Group. Log the model in MLflow

DELTA LAKE

Model Training

Daily batch job
Pandas_udf to read the model from MLflow & generate predictions

DELTA LAKE

Model Deployment

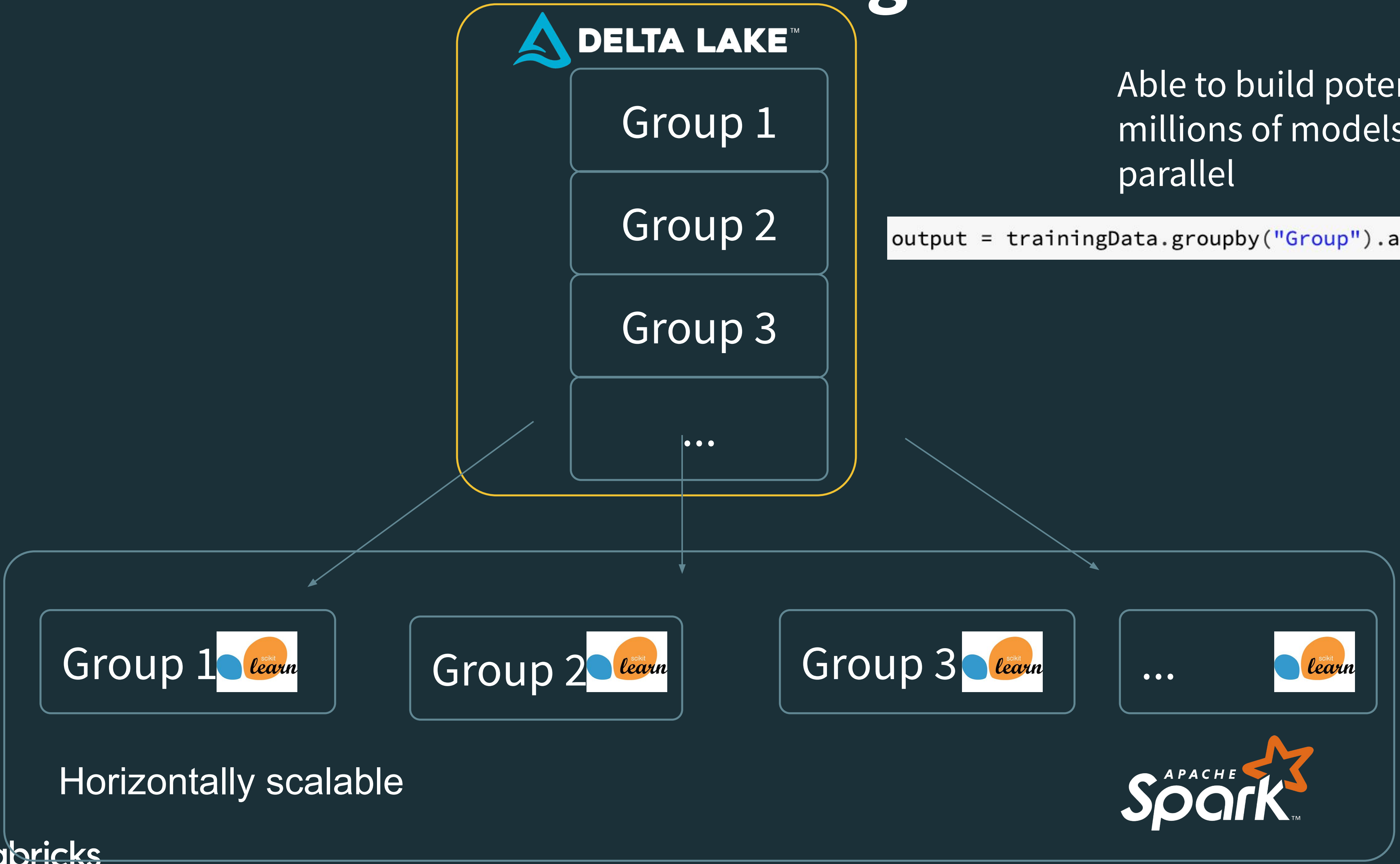
Automated through scheduled jobs

mlflow

Building Models in Parallel

Able to build potentially millions of models in parallel

```
output = trainingData.groupby("Group").apply(training_function)
```



Horizontally scalable

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
Model Registry

Centralized and collaborative model lifecycle management

new!

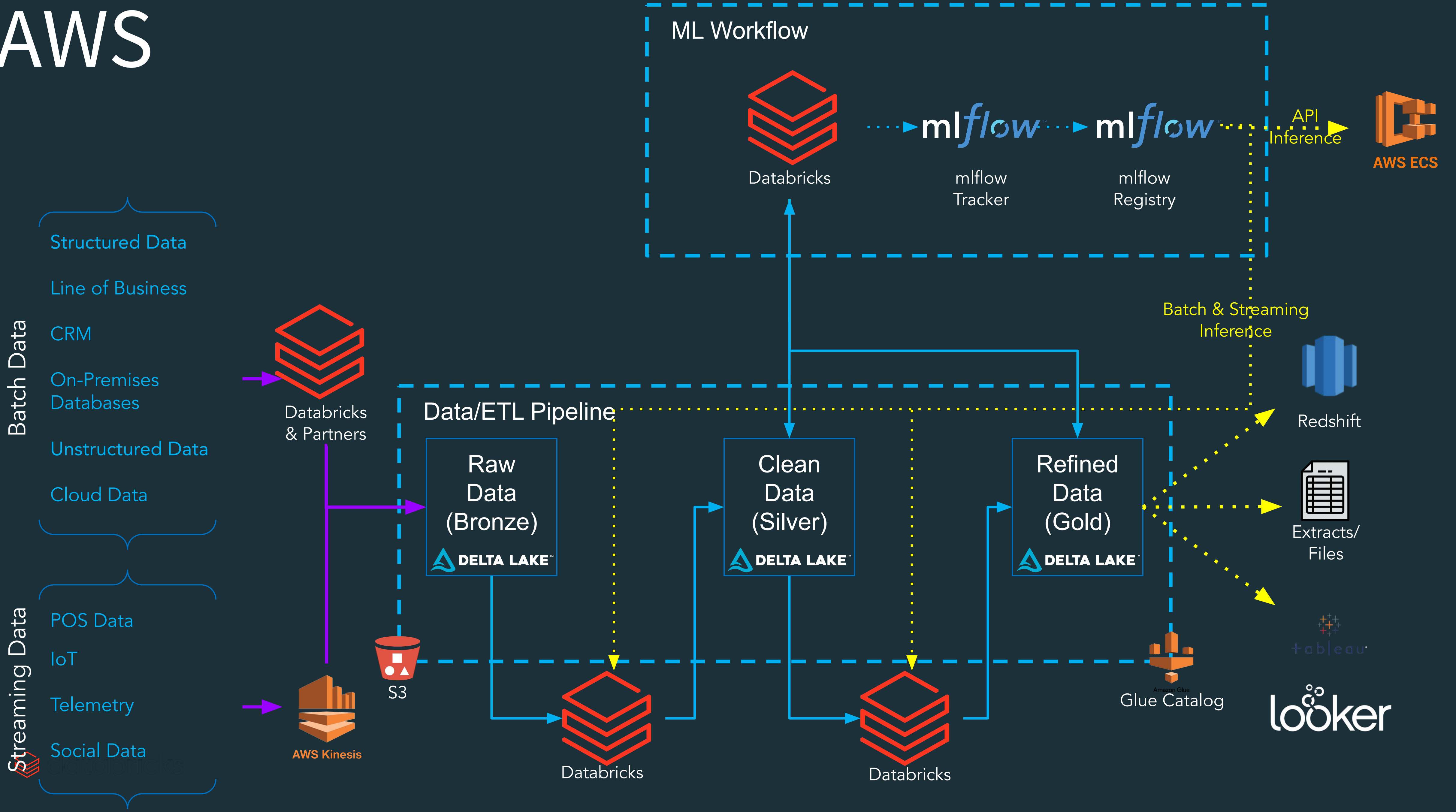
How it all fits together

- MLflow to track, analyze, reproduce, and deploy models
- Spark to accelerate model development
  - Data or Model Parallel training
  - Distributed Hyperparameter search
  - Distributed AutoML
- Delta to manage your data lake
  - Optimized data format reduces data load time for faster training.
  - Time Travel to isolate datasets during training
  - Support for both tabular and binary data

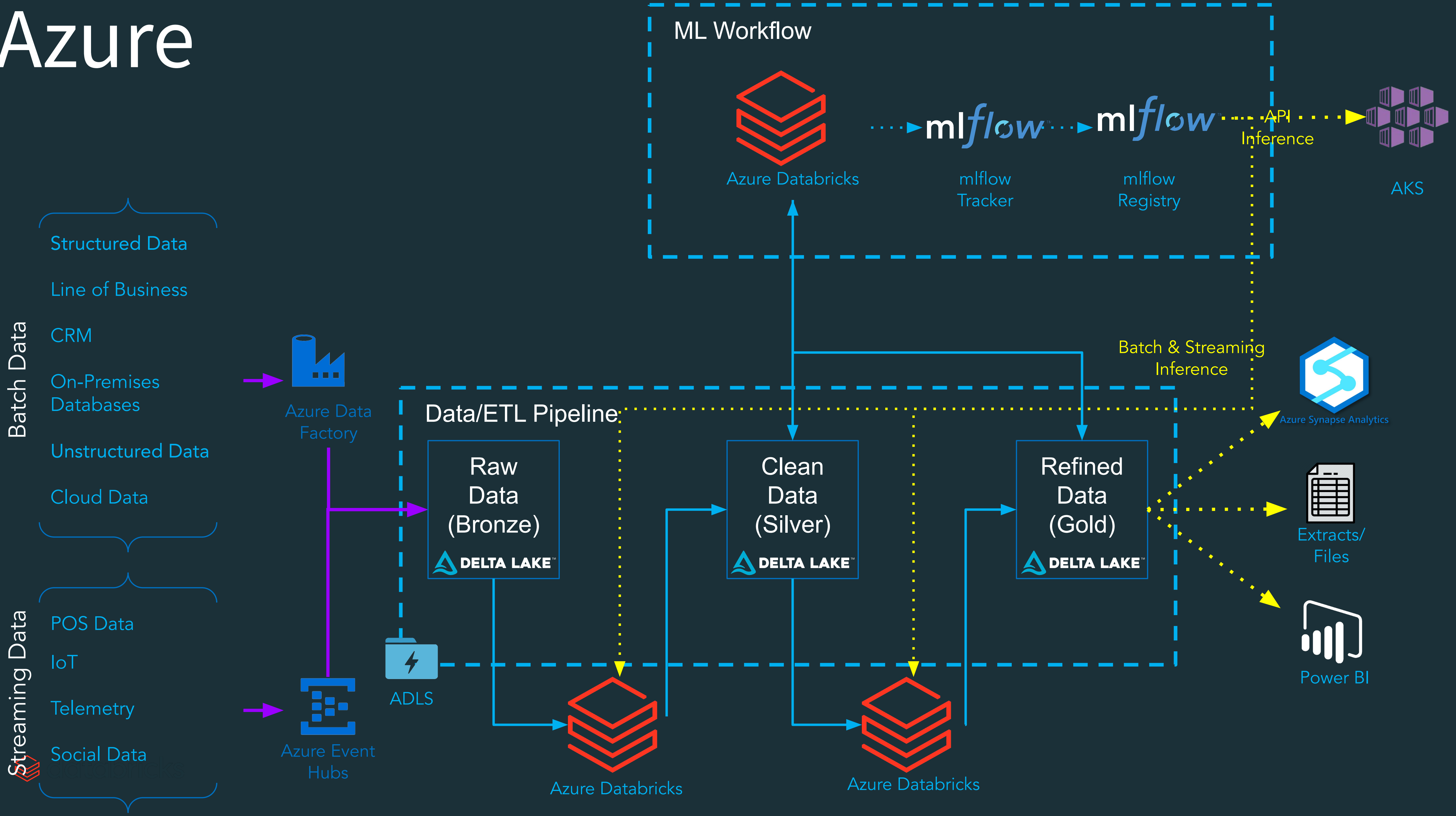
Databricks

- MLflow + Spark + Delta in one seamless environment
- Simple, secure, scalable, workspace with self-serve access to compute
- Highly collaborative environment to accelerate onboarding, and foster team based innovation

AWS


ML Workflow
Databricks
mlflow
mlflow
mlflow
Tracker
mlflow
Registry
API
Inference
AWS ECS
Batch Data
Structured Data
Line of Business
CRM
On-Premises
Databases
Unstructured Data
Cloud Data
Streaming Data
POS Data
IoT
Telemetry
Social Data
Databricks
& Partners
AWS Kinesis
S3
Data/ETL Pipeline
Raw
Data
(Bronze)
DELTA LAKE
Clean
Data
(Silver)
DELTA LAKE
Refined
Data
(Gold)
DELTA LAKE
Databricks
Databricks
Batch & Streaming
Inference
Redshift
Extracts/
Files
tableau
Amazon Glue
Glue Catalog
looker

Azure



ML Workflow

Azure Databricks

mlflow Tracker

mlflow Registry

API Inference

AKS

Batch & Streaming Inference

Batch Data

Structured Data

Line of Business

CRM

On-Premises Databases

Unstructured Data

Cloud Data

Streaming Data

POS Data

IoT

Telemetry

Social Data

Azure Data Factory

Azure Event Hubs

ADLS

Data/ETL Pipeline

Raw Data (Bronze)

DELTA LAKE

Clean Data (Silver)

DELTA LAKE

Refined Data (Gold)

DELTA LAKE

Azure Databricks

Azure Databricks

Azure Synapse Analytics

Extracts/ Files

Power BI

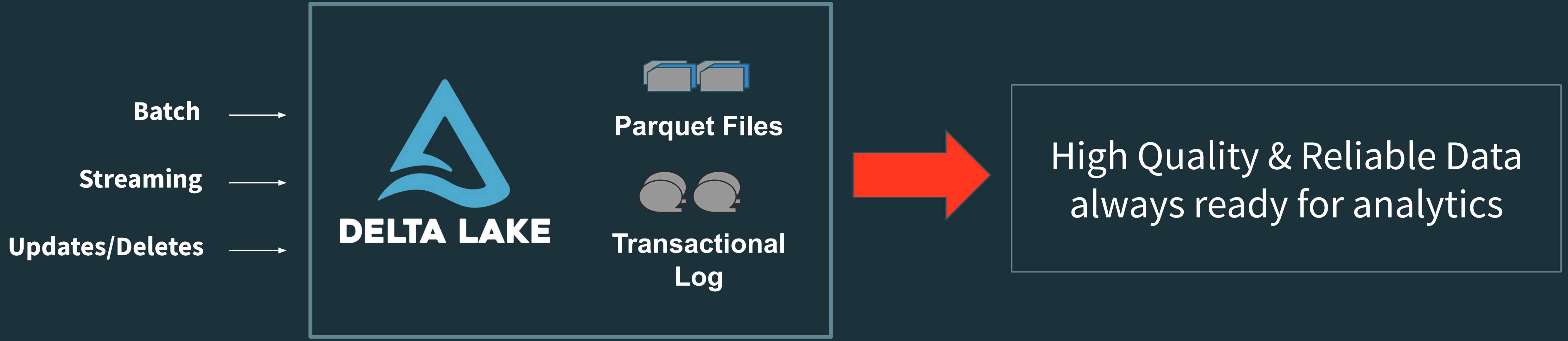
Make the Transition Easy

Common Tools	Single Node on Databricks	Distributed on Databricks
Jupyter Notebook	*Same*, Databricks Notebook	Databricks Notebook
Pandas DataFrame		*Same*, Spark DataFrame, Koalas DataFrame
Matplotlib		
scikit-learn		*Same*, sparkML
xgboost		*Same*, xgboost4j, lightgbm
Keras/Tensorflow/PyTorch		*Same* (+ HorovodRunner)
R or RStudio		SparkR or RStudio

A Deeper Look

DELTA LAKE

Delta Lake ensures data reliability



Key Features

- ACID Transactions
- Schema Enforcement
- Unified Batch & Streaming
- Time Travel/Data Snapshots

Time Travel

Reproduce experiments & reports

```
SELECT count(*) FROM events
TIMESTAMP AS OF timestamp

SELECT count(*) FROM events
VERSION AS OF version

spark.read.format("delta").option("timestampAsOf",
timestamp_string).load("/events/")
```

Rollback accidental bad writes

```
INSERT INTO my_table
    SELECT * FROM my_table TIMESTAMP AS
OF
    date_sub(current_date(), 1)
```

BinaryFile Support

```
df = spark.read.format("binaryFile").option("pathGlobFilter", "*.jpg").load("/path/to/dir")
```

Write arbitrary file types to Parquet

Creates four columns: file path, file date, file size, serialized content of file

Why?

- Solves “small file problem” with mature big data standard
- Can append additional columns to the dataframe
- Can use Delta + TimeTravel to manage binary data

Machine learning with Databricks

How Databricks helps Data Scientists

Distributed Machine Learning

Spark MLlib for distributed models

Migrate **Single Node to distributed** with just a few lines of code changes:

- Distributed **hyperparameter search** (Hyperopt, Gridsearch)
- **PandasUDF** to distribute models over subsets of data or hyperparameters
- **Koalas**: Pandas DataFrame API on Spark

Deep Learning distributed training. (**HorovodRunner**)

Use your own tools

Multiple languages in Databricks Notebooks (Python, R, Scala, SQL)

Databricks Connect: connect external tools with Databricks (IDEs, RStudio, Jupyter...)

R support

Native R notebooks on Databricks

Python (Scikit-Learn, Pandas)

RStudio & RStudio Server integrations

Scaling and parallelizing with SparkR & SparklyR

Upcoming features

Data Science Workspace

- Project-based Git integration
- Share & reproduce Conda environments

Hosted JupyterLab

Hosted Shiny Apps

ML Runtime Optimizations

Reliable and secure distribution of open source ML frameworks

Packages and optimizes most common ML Frameworks

XGBoost · K · scikit learn · …

Built-in Optimization for Distributed Deep Learning .

HOROVOD

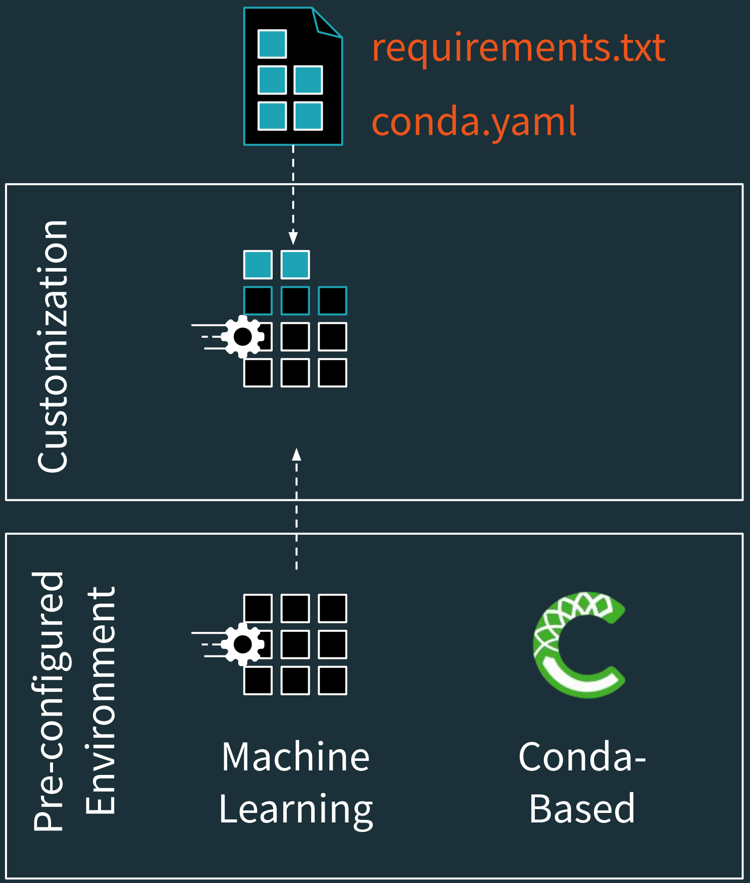
Distribute and Scale any Single-Machine ML Code to 1,000's of machines.

Built-In AutoML and Experiment Tracking

mlflow™

AutoML and Tracking / Visualizations with MLflow

Customized Environments using Conda



Petastorm

(Roadmap)

- Allows you to load Parquet directly into Deep Learning frameworks
- Supports TensorFlow and PyTorch
- Works for Single Node & Distributed

Meet Horovod

- Distributed training framework for TensorFlow
- Inspired by work of Baidu, Facebook, et al.
- Uses bandwidth-optimal communication protocols
    - Makes use of RDMA (RoCE, InfiniBand) if available
- Seamlessly installs on top of TensorFlow via `pip install horovod`
- Named after traditional Russian folk dance where participants dance in a circle with linked hands

Hyperopt

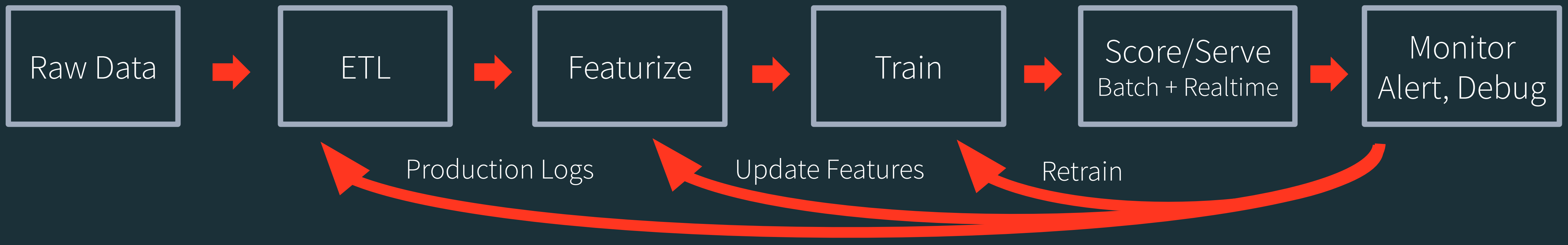
- Open Source Bayesian HyperParameter Optimizer
- Uses TPE (Tree of Parzen Estimators) to model the prior
  - Scales better to high dimensional problems than GPs
- Have created a new “SparkTrials” class
  - Supports parallel suggestion serving out of the box on spark
  - Contributing back to the open source

mlflow
TM

ML Lifecycle and Challenges

mlflow

An open source platform for the machine learning lifecycle



kafka · S3 · Delta · Microsoft Azure Blob Storage · hadoop · mongoDB · Spark · pandas · SQL · learn · PYTORCH · learn · TensorFlow · R · Spark · dmlc XGBoost · Azure Machine Learning · Amazon SageMaker · docker · Spark

Zoo of Ecosystem Frameworks

Tuning

Deploy

Model Mgmt

Collaboration

Scale

Governance

Feature Repository

Experiment Tracking

AutoML, Hyper-p. search

Remote Cloud Execution

Project Mgmt (scale teams)

Model Exchange

A/B Testing

CI/CD/Jenkins push to prod

Orchestration (Airflow, Jobs)

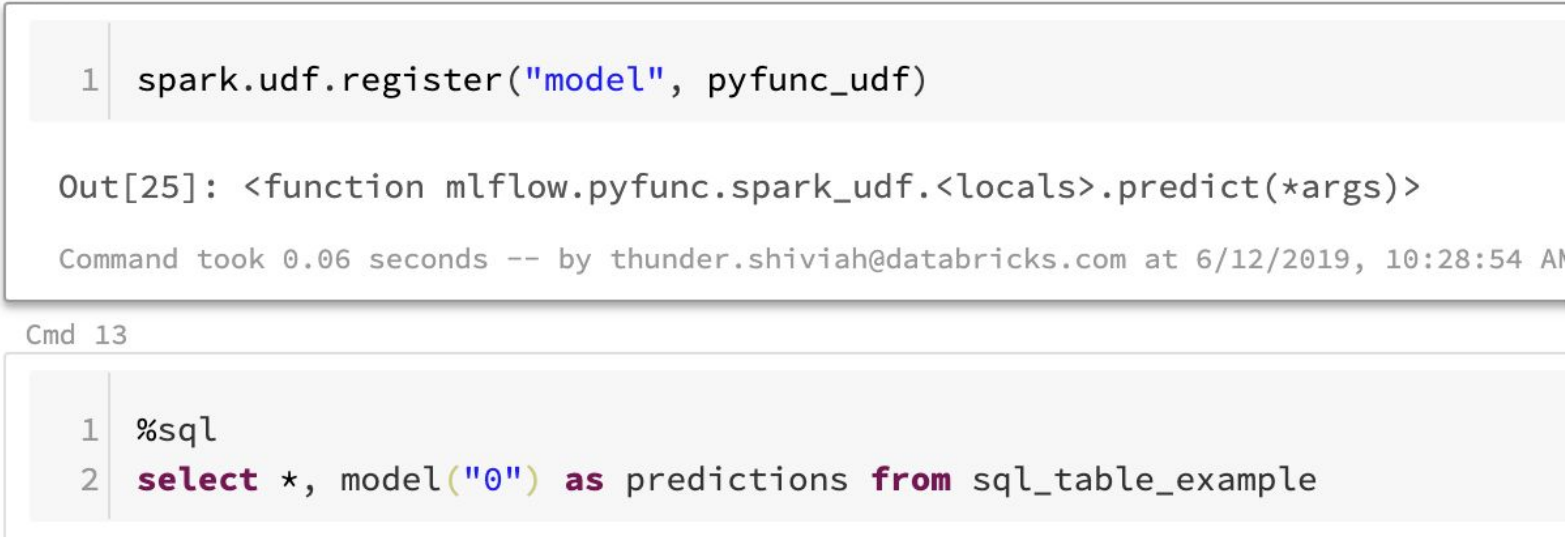
Lifecycle mgmt.

Data Drift

Model Drift

databricks

Use MLflow + spark UDFs to democratize ML within the org.



```python
spark.udf.register("model", pyfunc_udf)
```

```
Out[25]: <function mlflow.pyfunc.spark_udf.<locals>.predict(*args)>

Command took 0.06 seconds -- by thunder.shiviah@databricks.com at 6/12/2019, 10:28:54 AM
```

Cmd 13

```sql
%sql
select *, model("0") as predictions from sql_table_example
```

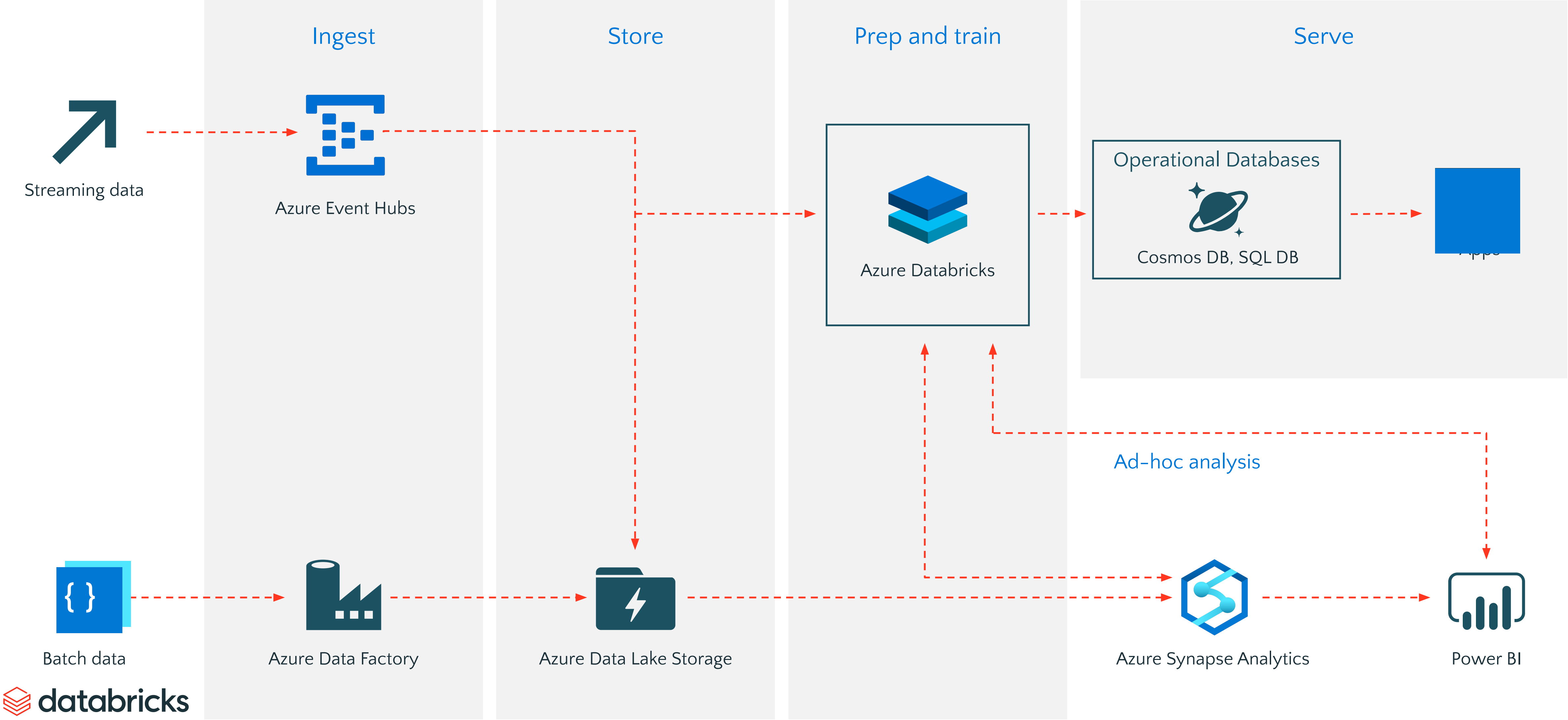
See my mlflow deployment example notebook.

Enable Every Level of Expertise

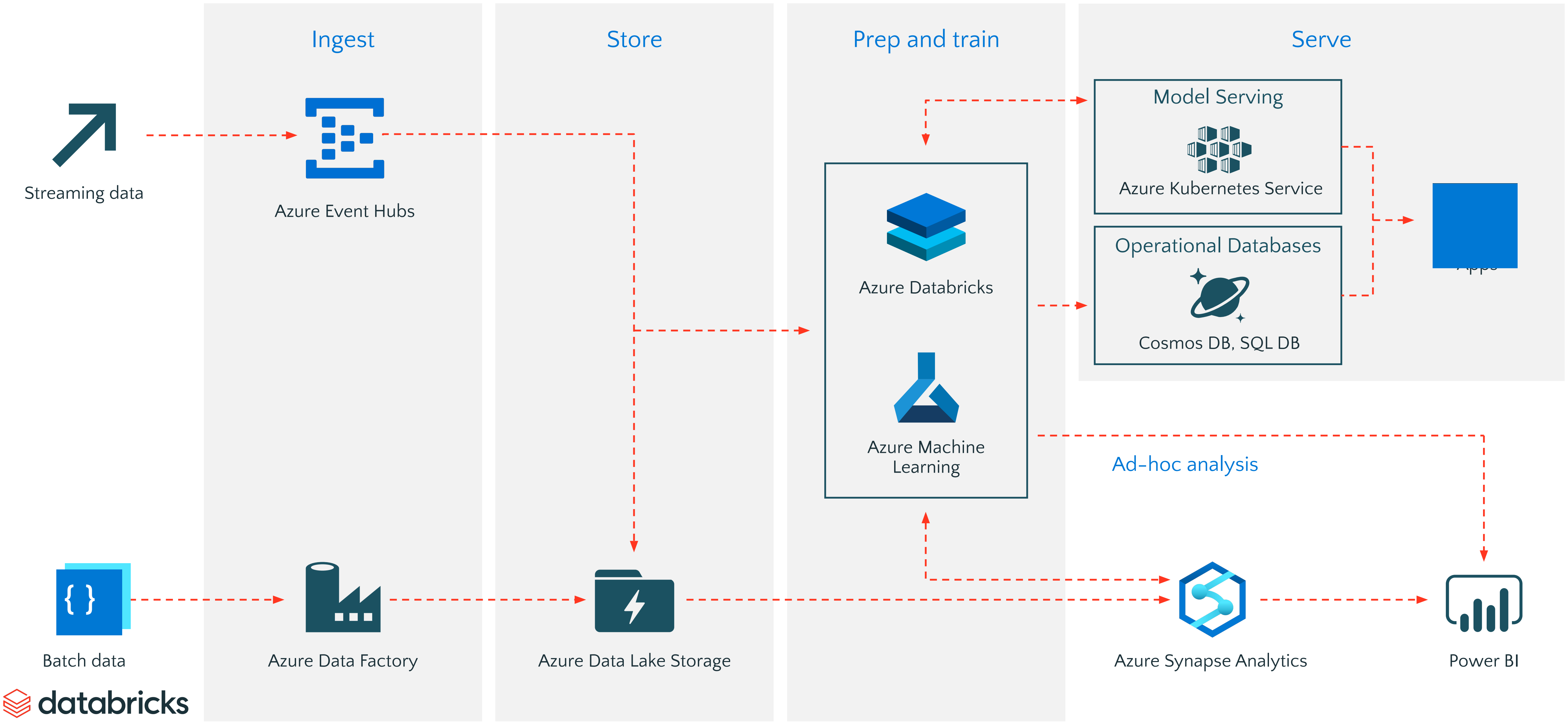
User	Offering	Details
Citizen Data Scientist	databricks AutoML Toolkit	/ Feature Factory / Feature Importance / Evolutionary Model Search
Engineer	mlflow ML Runtime and MLflow	/ Hyperparameter Tuning / Model Architecture Search
ML Expert / Researcher	HOROVOD mlflow ML Runtime Optimizations	/ Distributed Execution of Libraries / Latest AutoML Libraries (e.g. Uber's Ludwig)

Machine learning with Azure Databricks and Azure Machine Learning

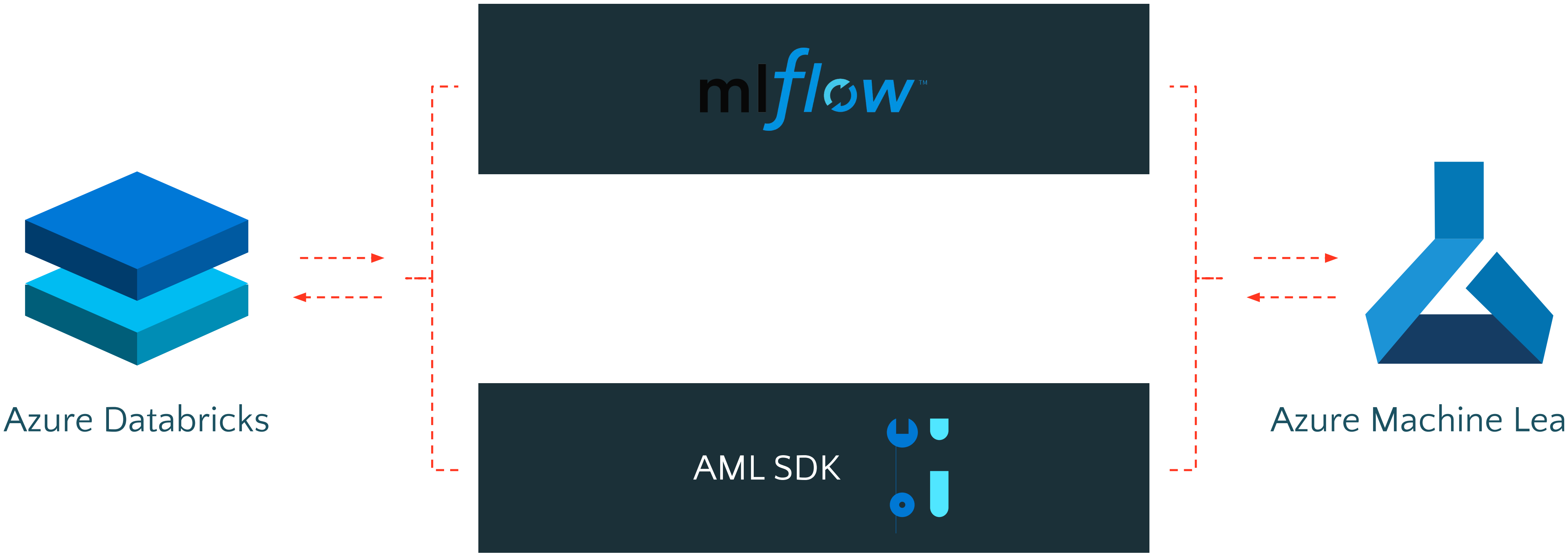
Machine learning



Machine learning



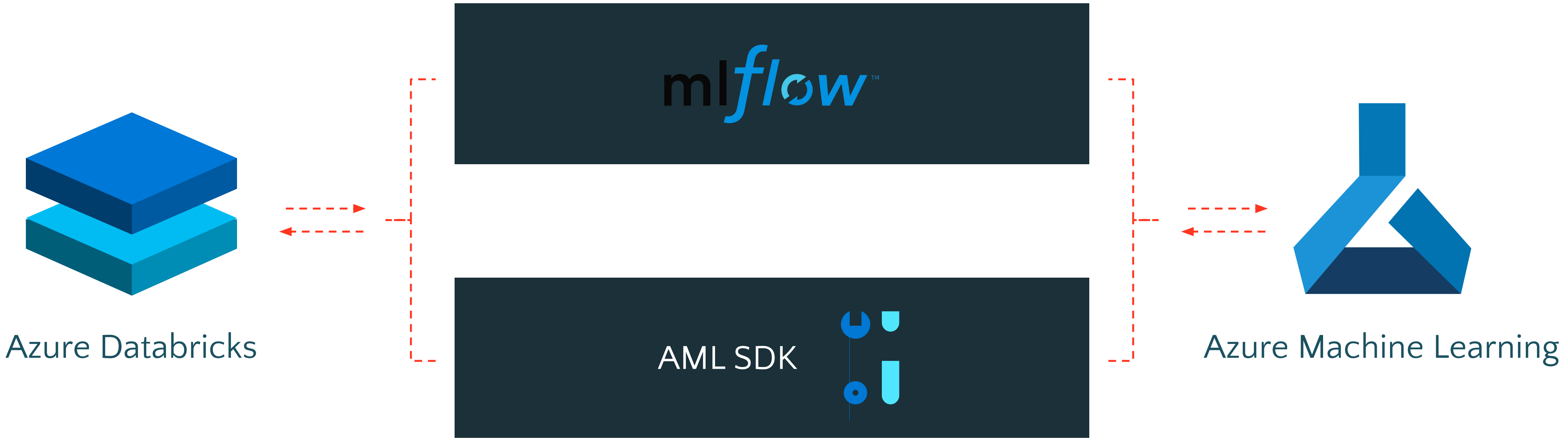
Azure Databricks and Azure ML are better together



Azure Databricks

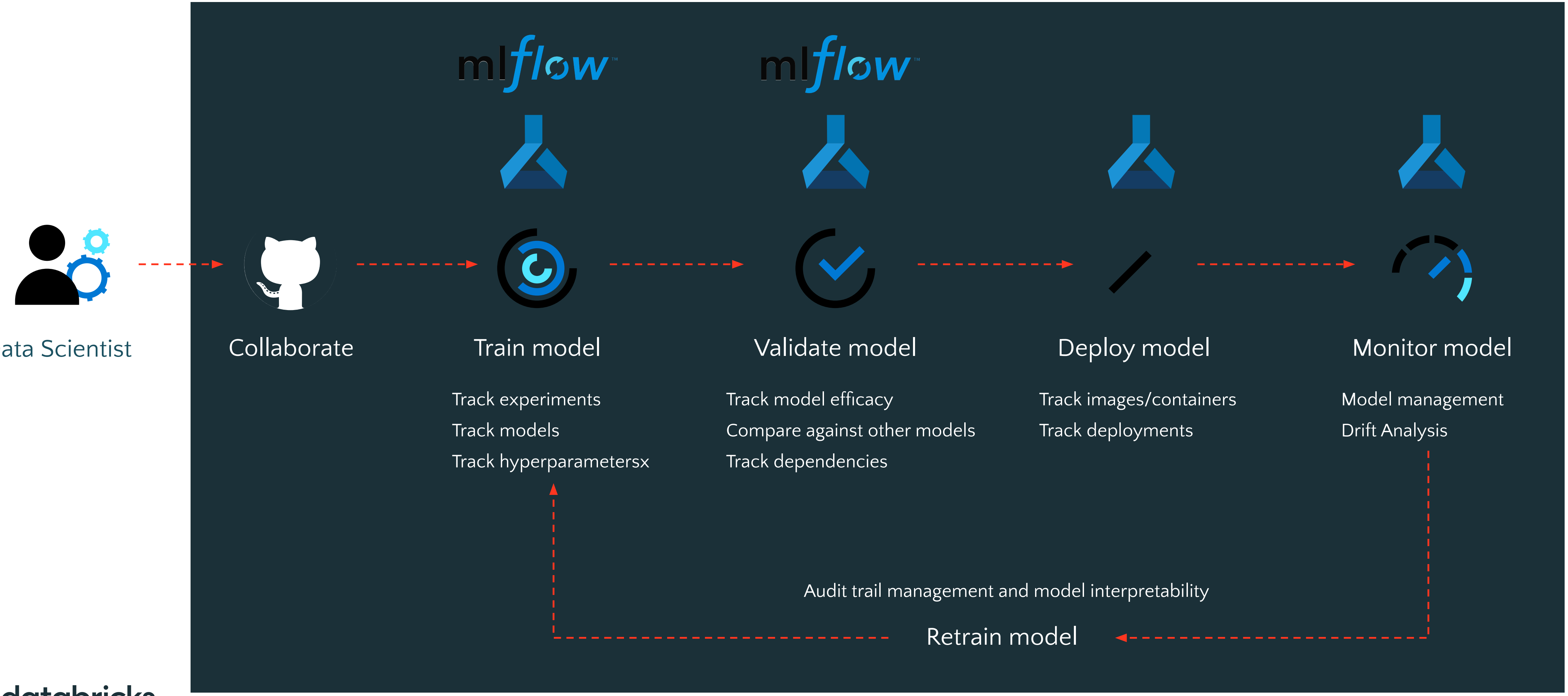
Azure Machine Learning

Azure Databricks and Azure ML are better together



- Log experiments and models in a central place
- Maintain audit trails centrally
- Deploy models seamlessly via Azure Machine Learning
- Implement robust MLOps

MLflow and Azure Machine Learning



Data Scientist

Collaborate

Train model

- Track experiments
- Track models
- Track hyperparametersx

Validate model

- Track model efficacy
- Compare against other models
- Track dependencies

Deploy model

- Track images/containers
- Track deployments

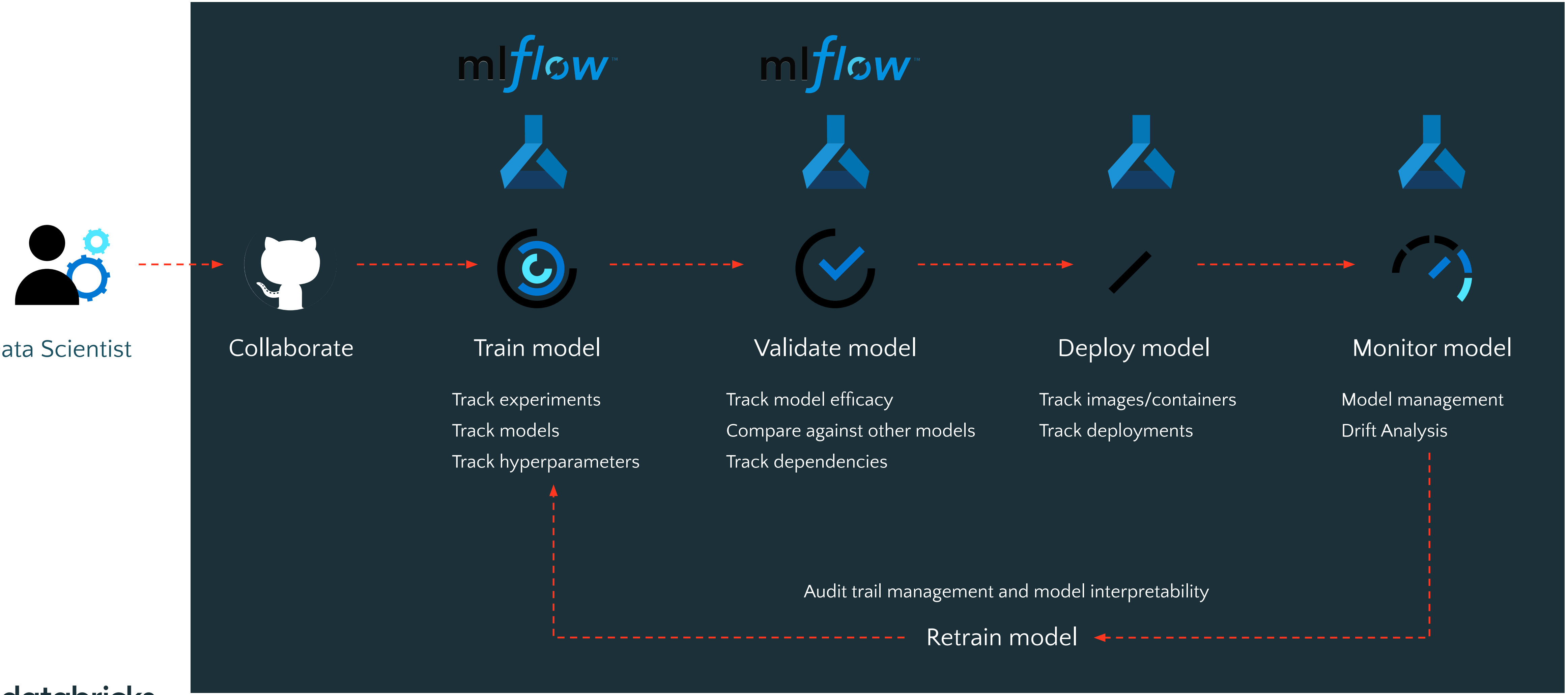
Monitor model

- Model management
- Drift Analysis

Audit trail management and model interpretability

Retrain model

MLflow and Azure Machine Learning



Data Scientist

Collaborate

Train model

- Track experiments
- Track models
- Track hyperparameters

Validate model

- Track model efficacy
- Compare against other models
- Track dependencies

Deploy model

- Track images/containers
- Track deployments

Monitor model

- Model management
- Drift Analysis

Audit trail management and model interpretability

Retrain model

Databricks with Azure Machine Learning

Azure Databricks → Engineered integration → Azure Machine Learning

Open & extensible

- Leverage the latest libraries and frameworks
- Perform distributed training across CPUs and GPUs
- Dedicated ML runtime with pre-built optimizations

MLflow integration

- Common experiment tracking and results backend
- Store models in a central model registry across Azure
- Combined view of all ML activity within Azure

ML & ML management

- Package and deploy models for inferencing at scale
- Leverage automated ML to design a model factory
- Create CI/CD pipelines for retraining with drift tracking and audit trails

Azure DevOps

Implement MLOps with Azure DevOps

Appendix

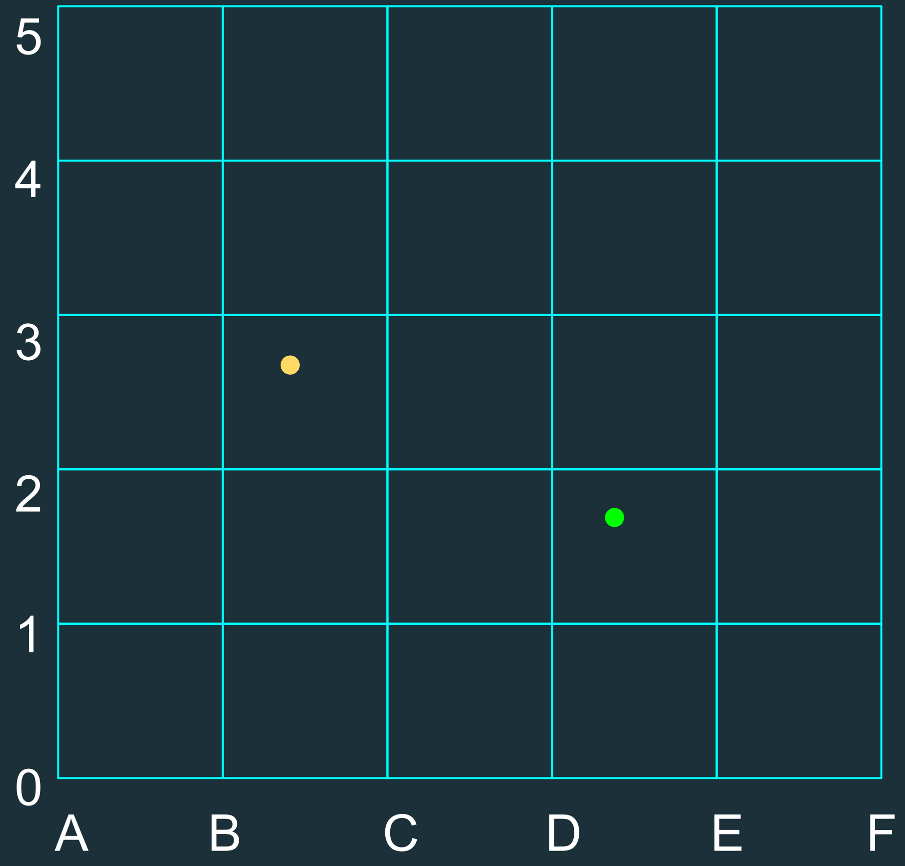
databricks

Review: Bayesian Optimization

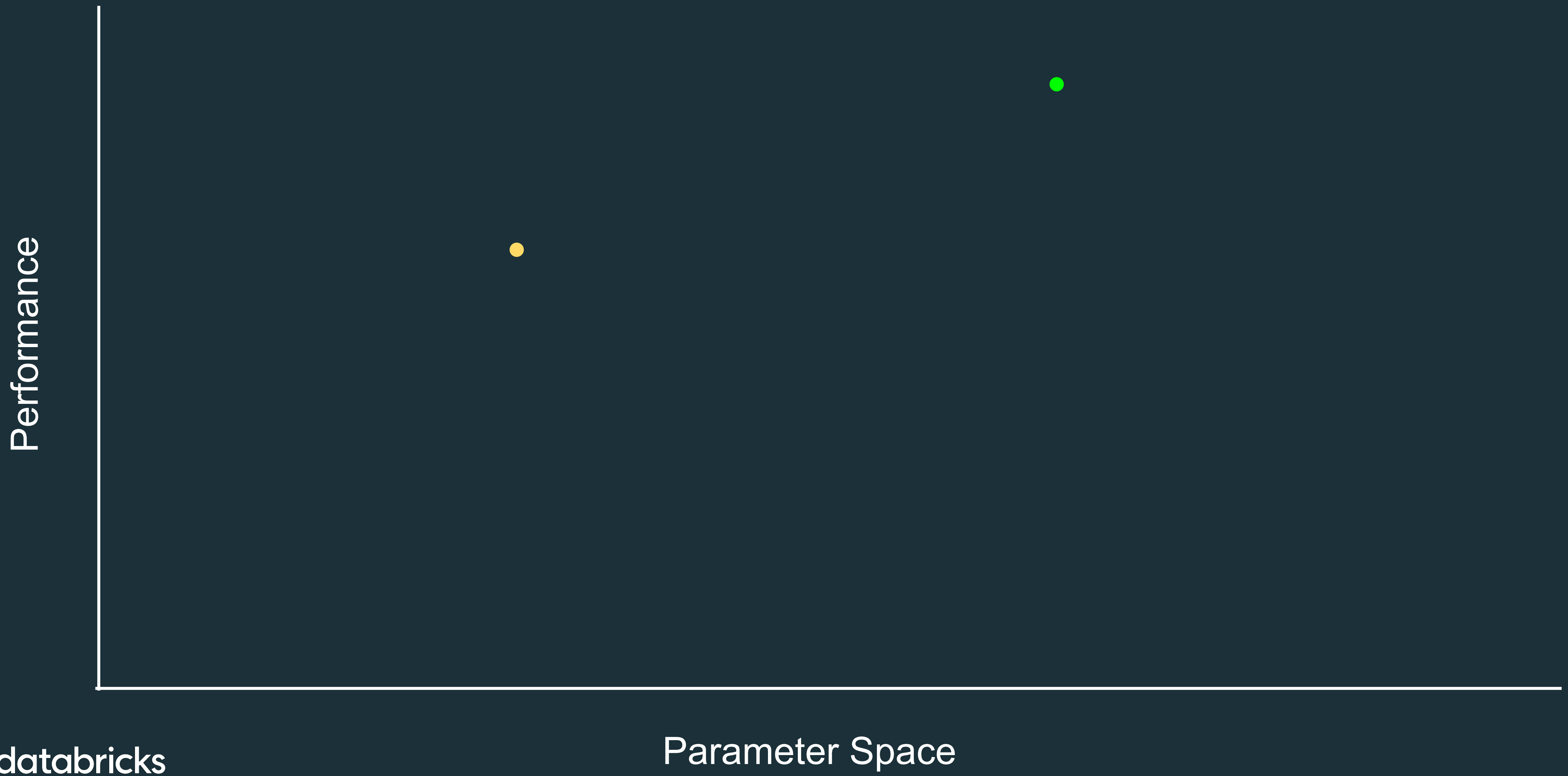
Create ranges, and trade off between exploration and exploitation to search space

param 1: a-f

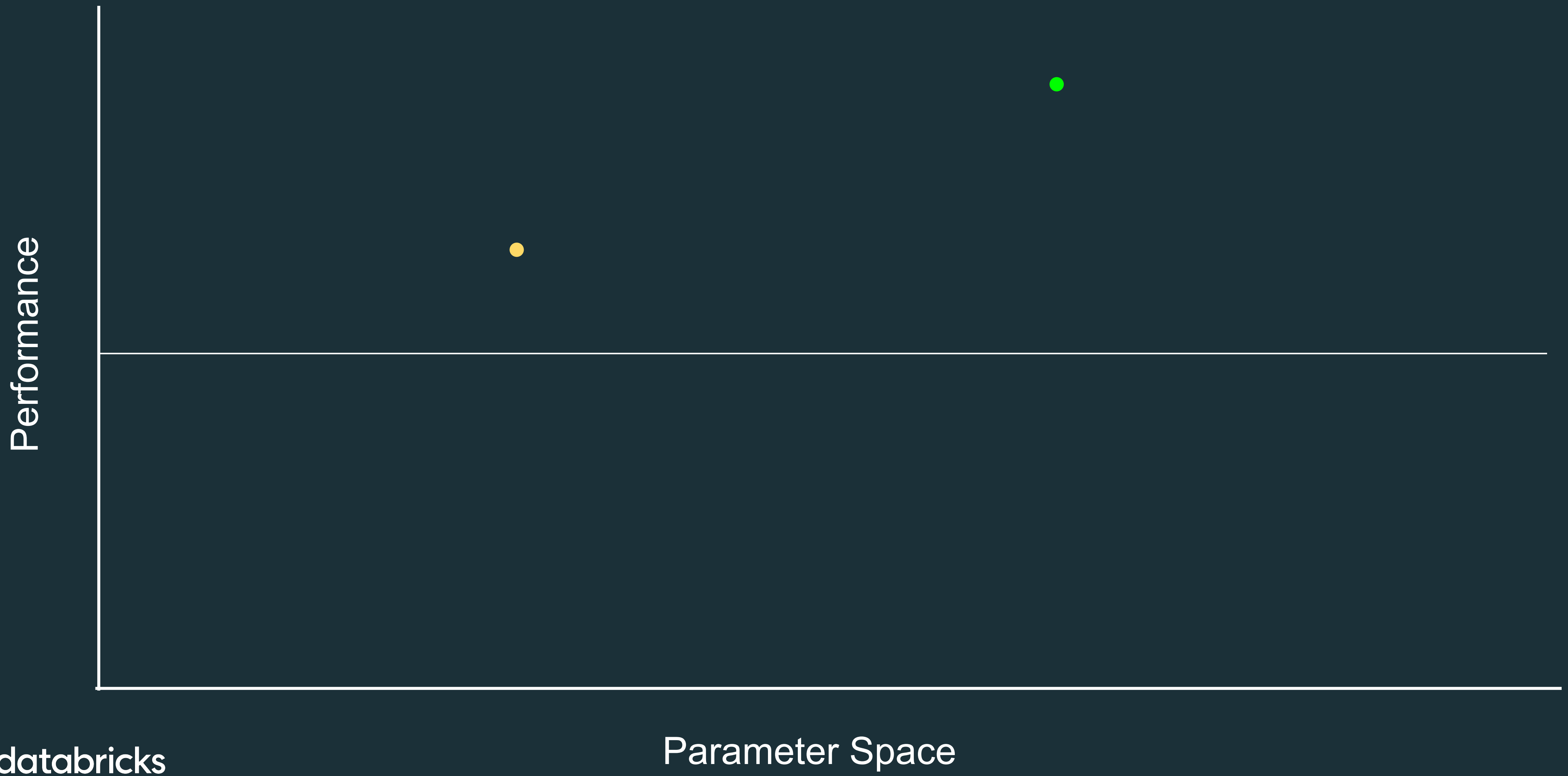
param 2: 0-5



Bayesian Optimization



Bayesian Optimization



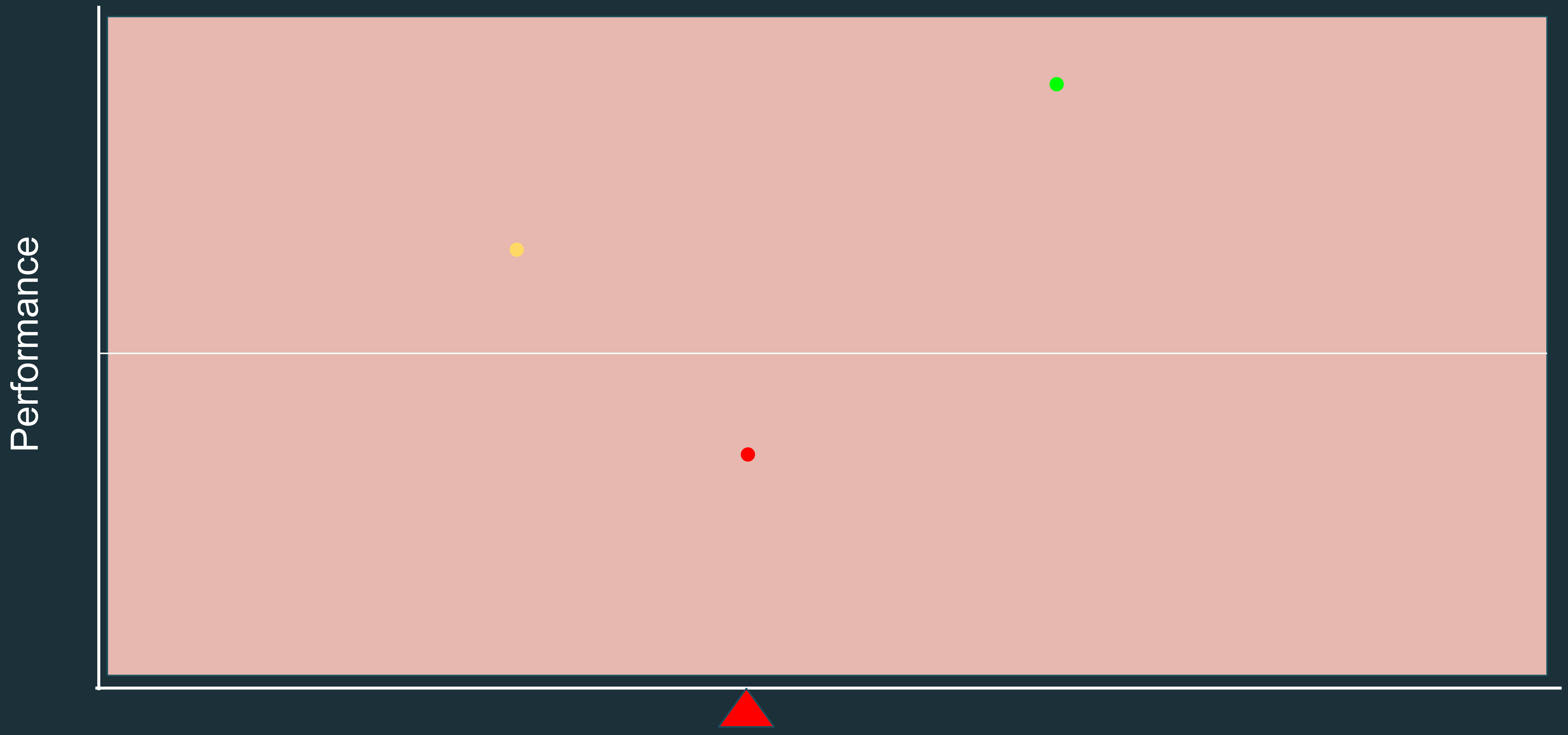
databricks

Bayesian Optimization

Performance
Parameter Space

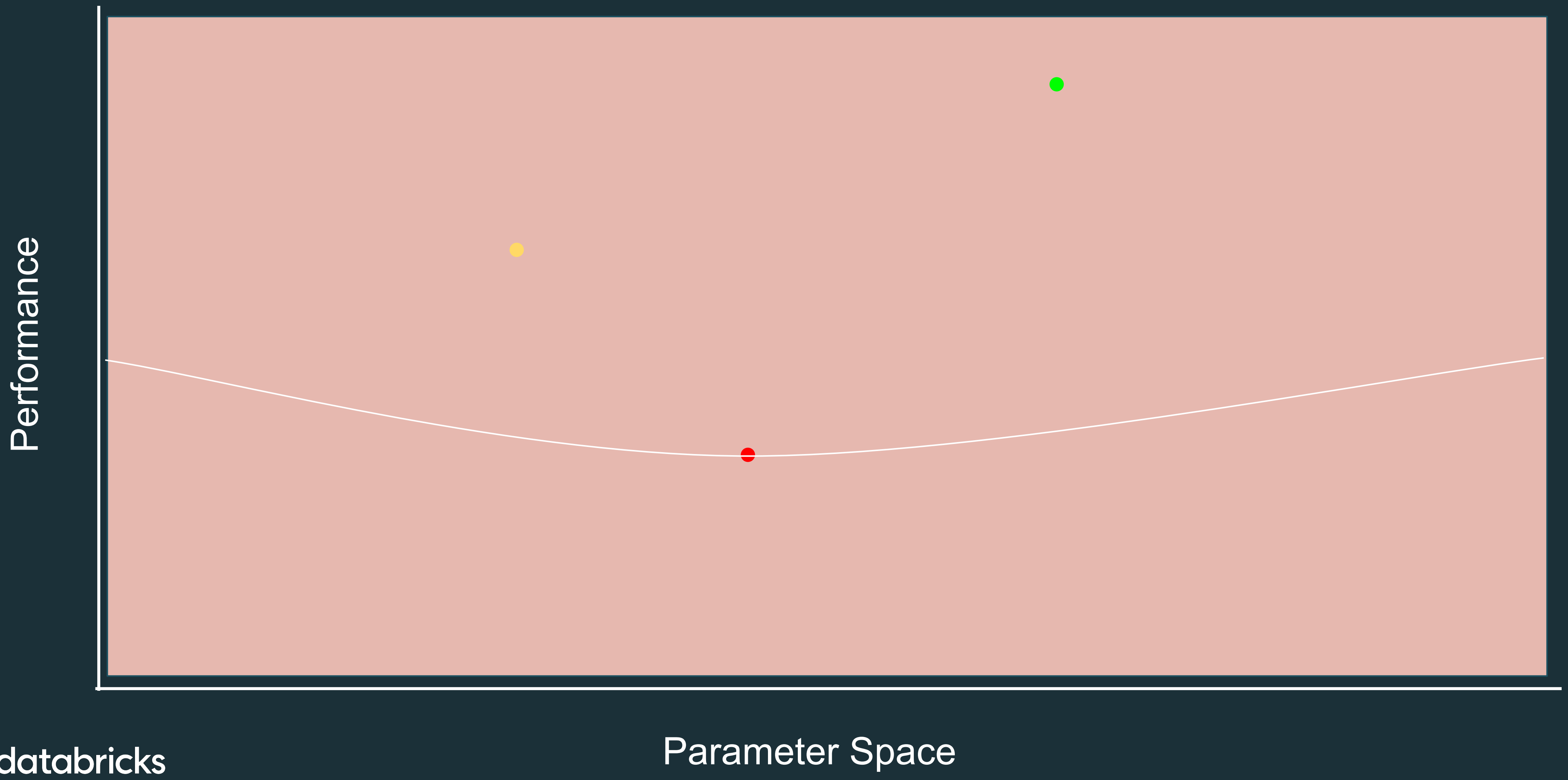
databricks

Bayesian Optimization



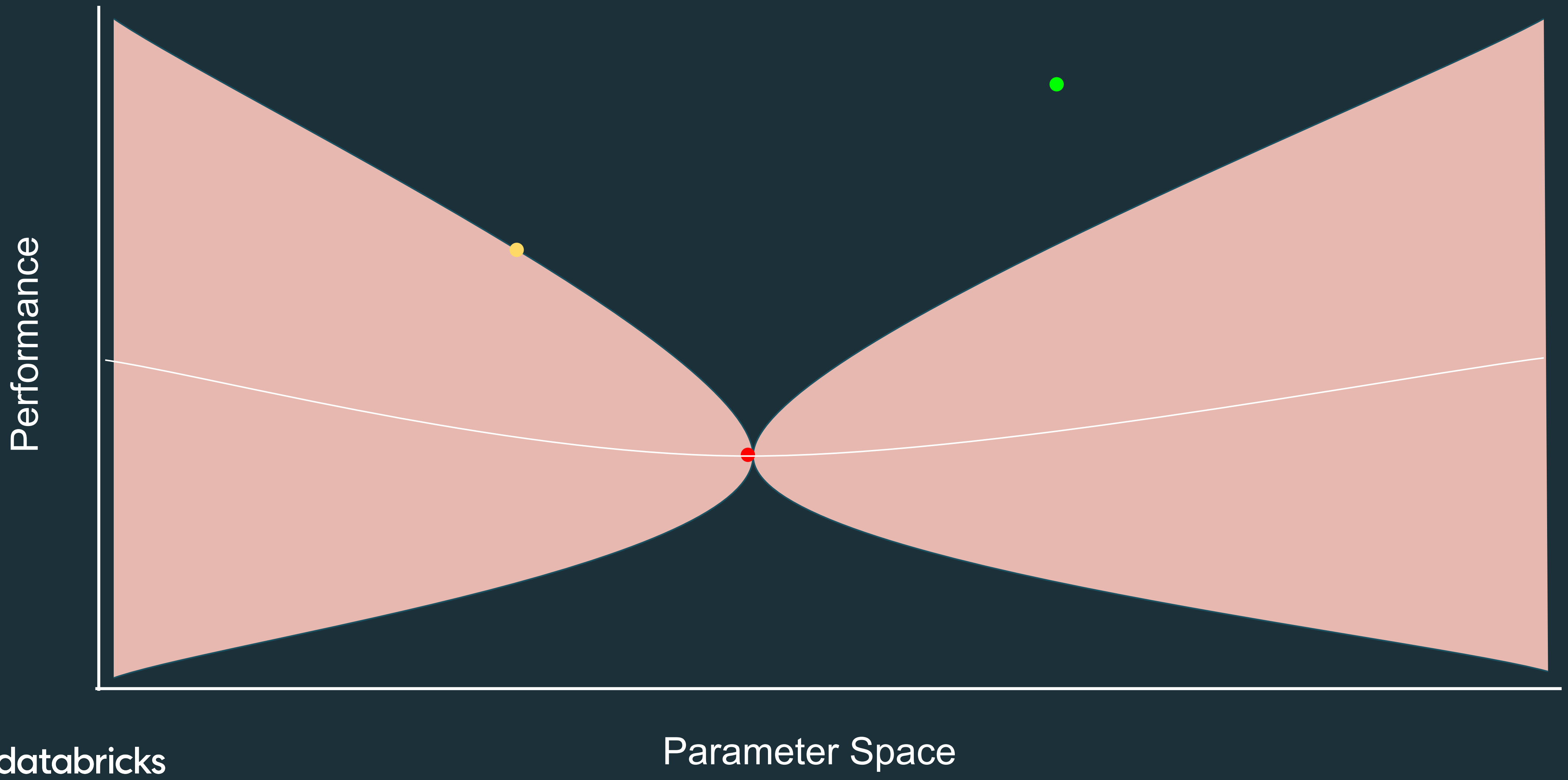
databricks

Bayesian Optimization



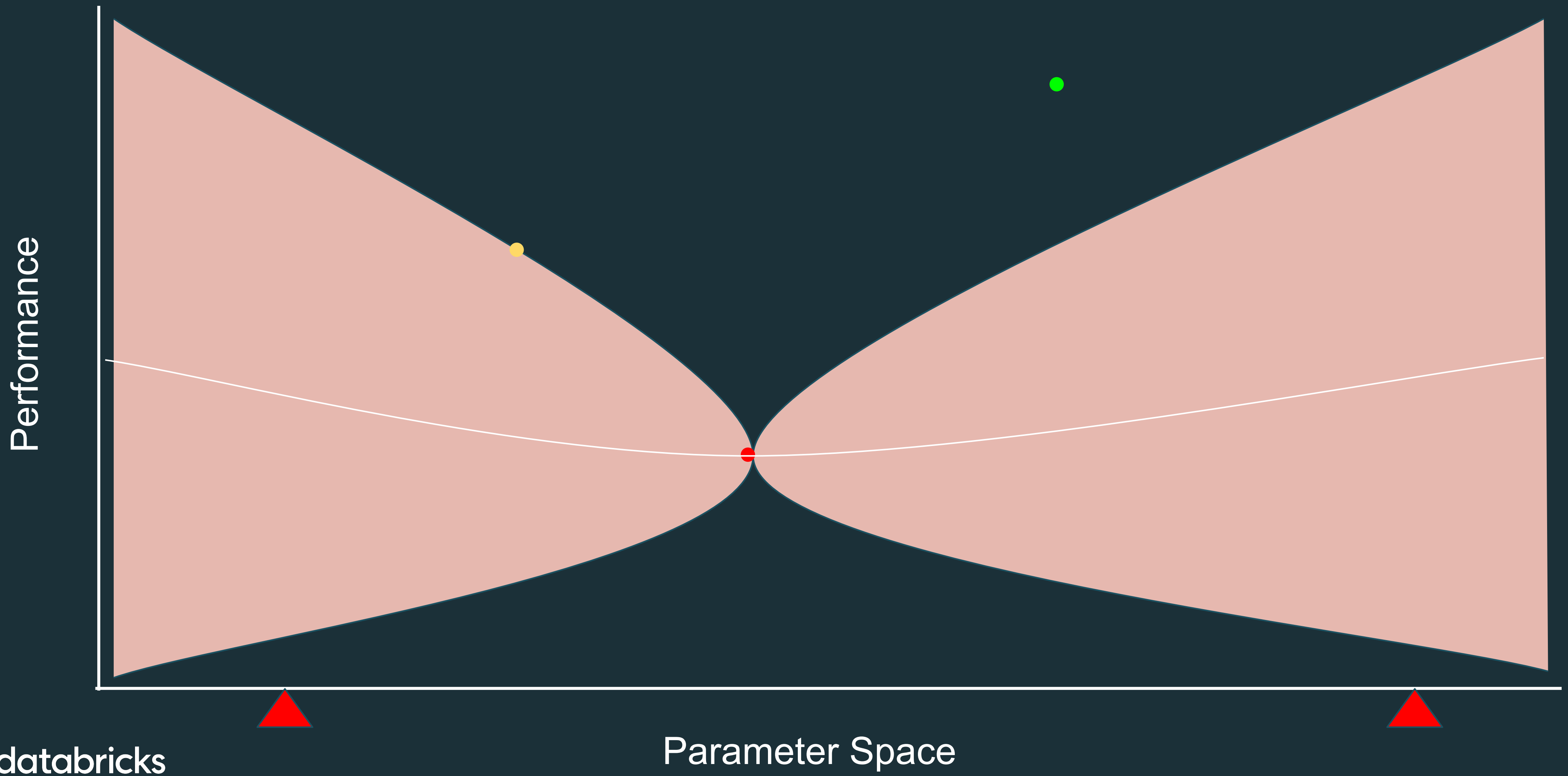
databricks

Bayesian Optimization

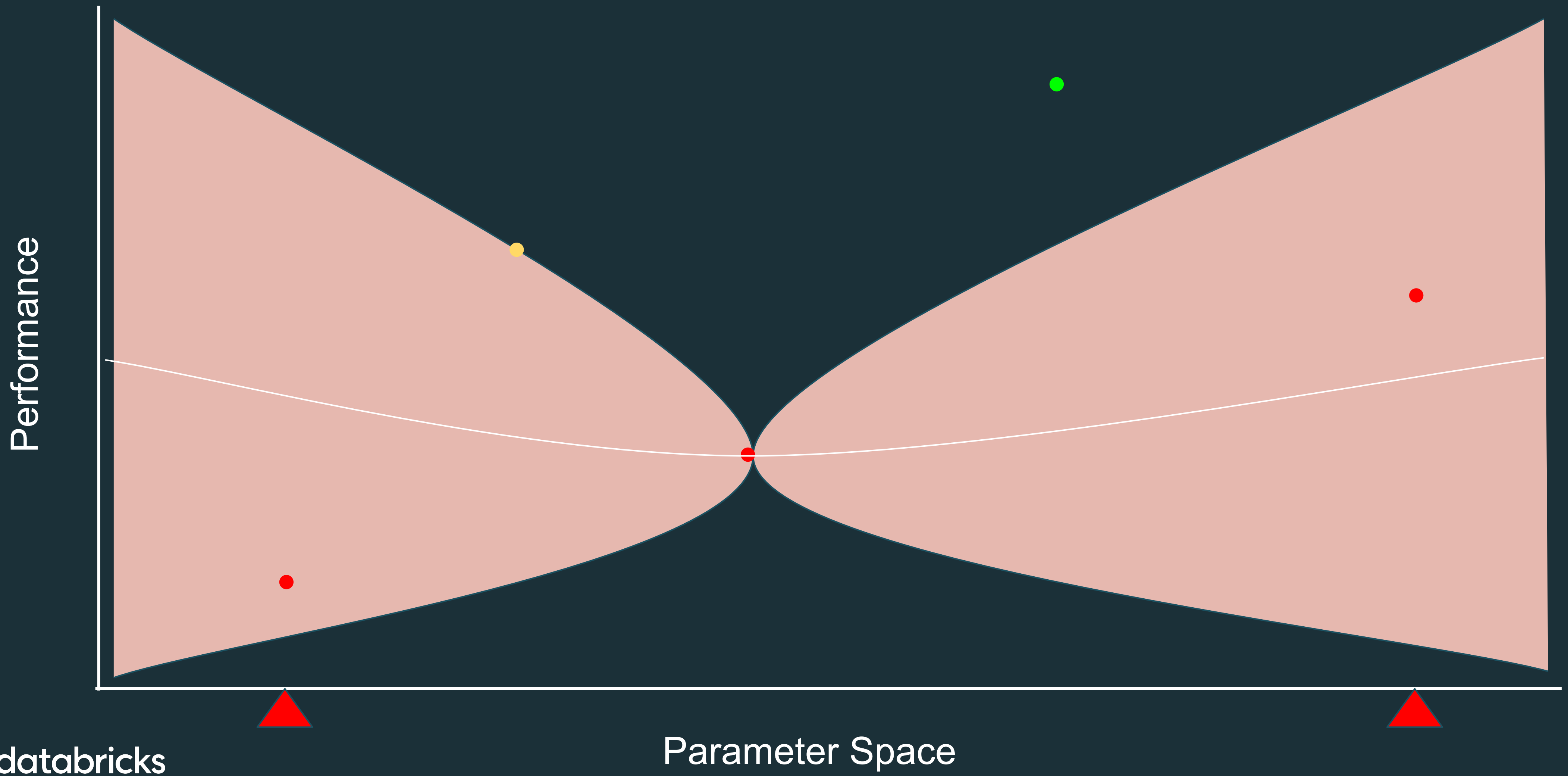


databricks

Bayesian Optimization

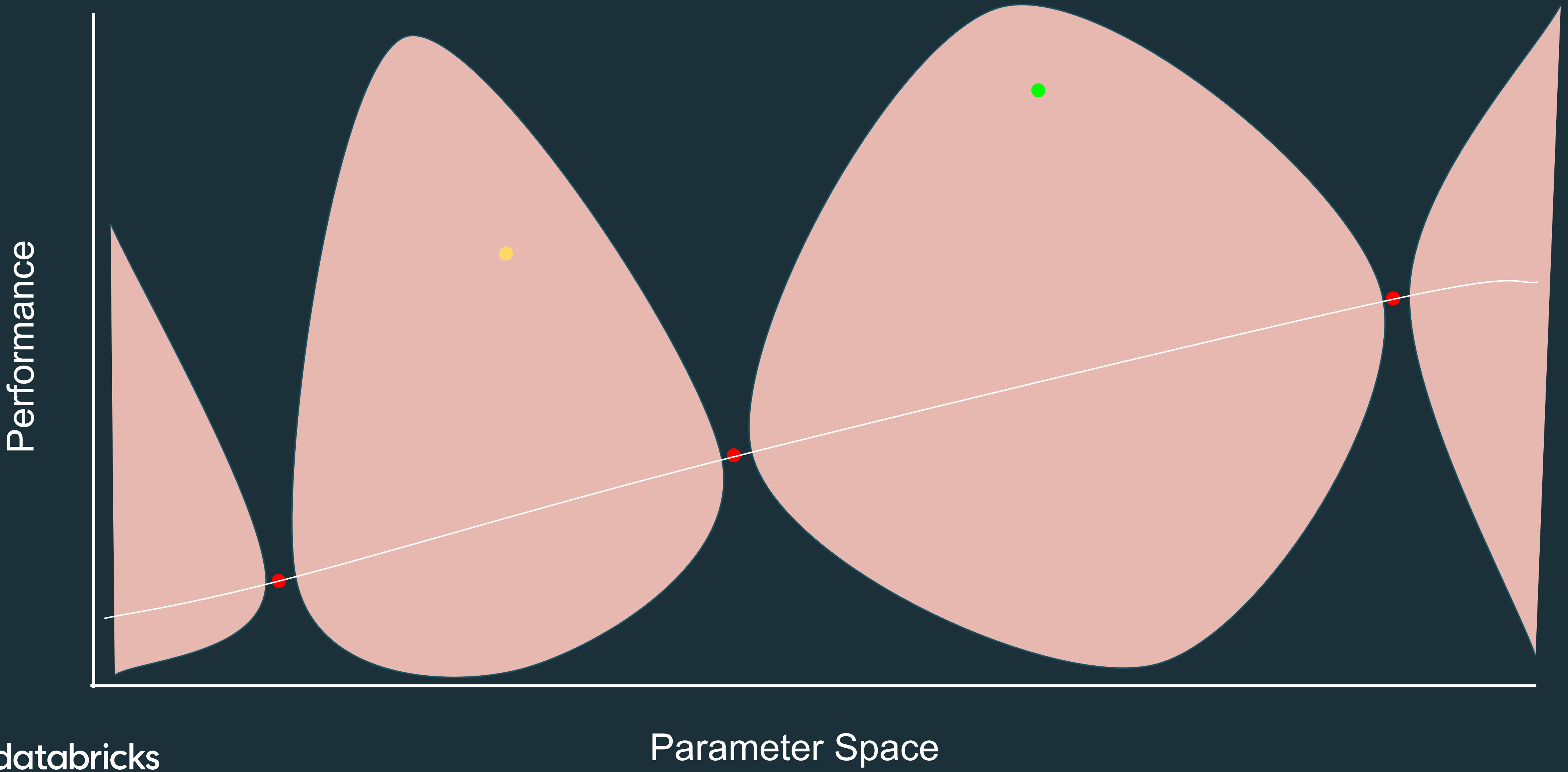


Bayesian Optimization

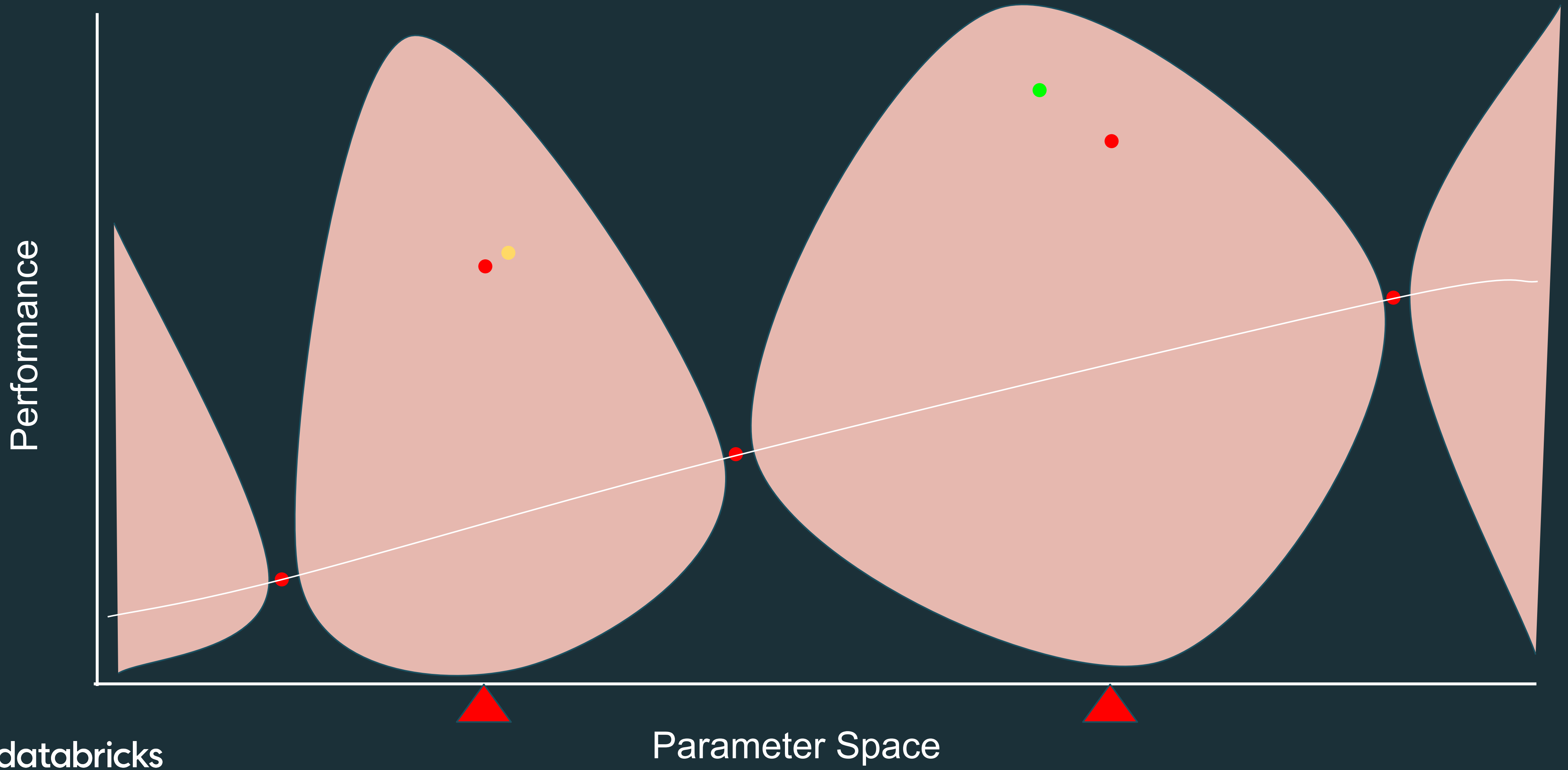


databricks

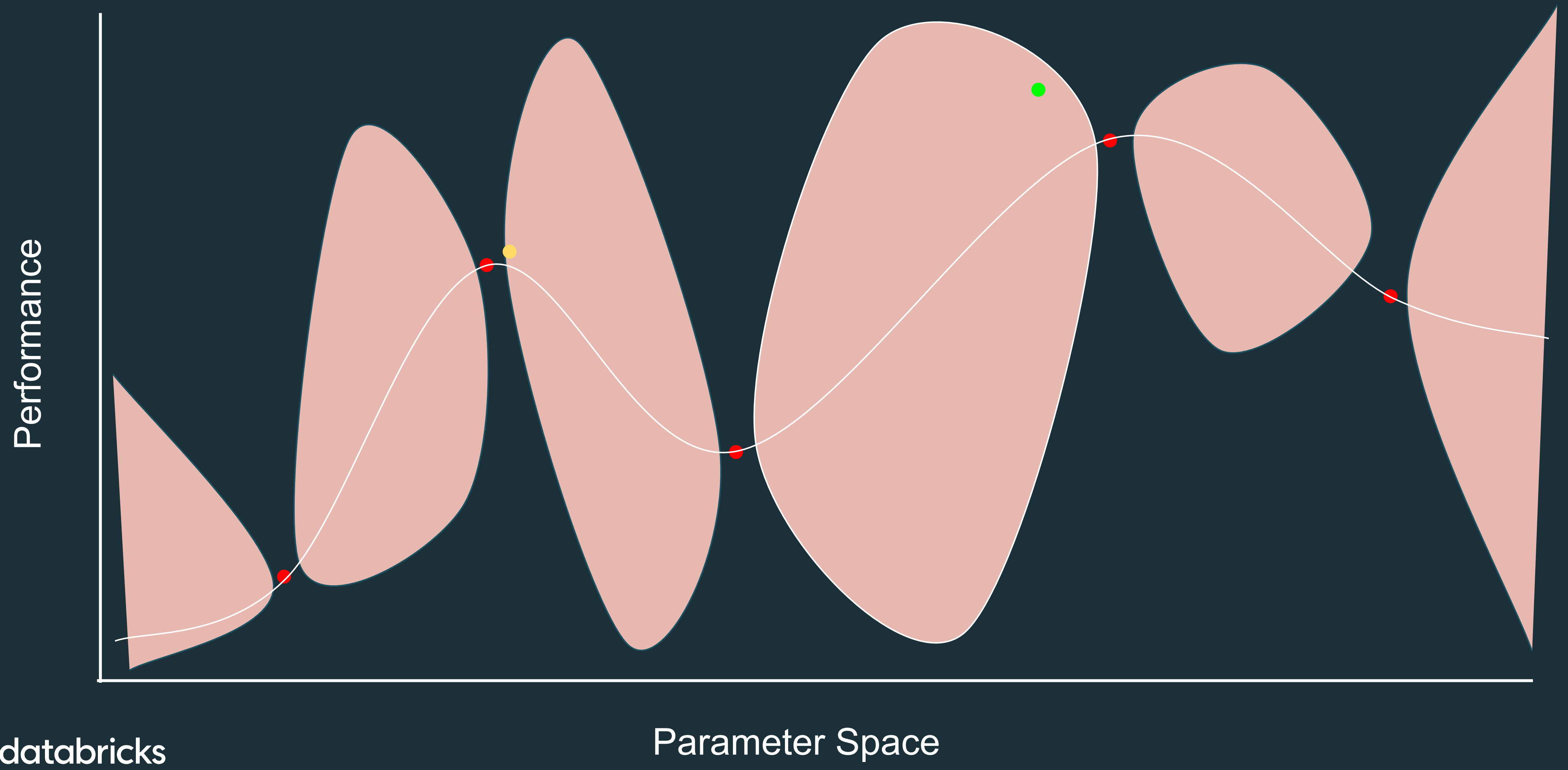
Bayesian Optimization



Bayesian Optimization

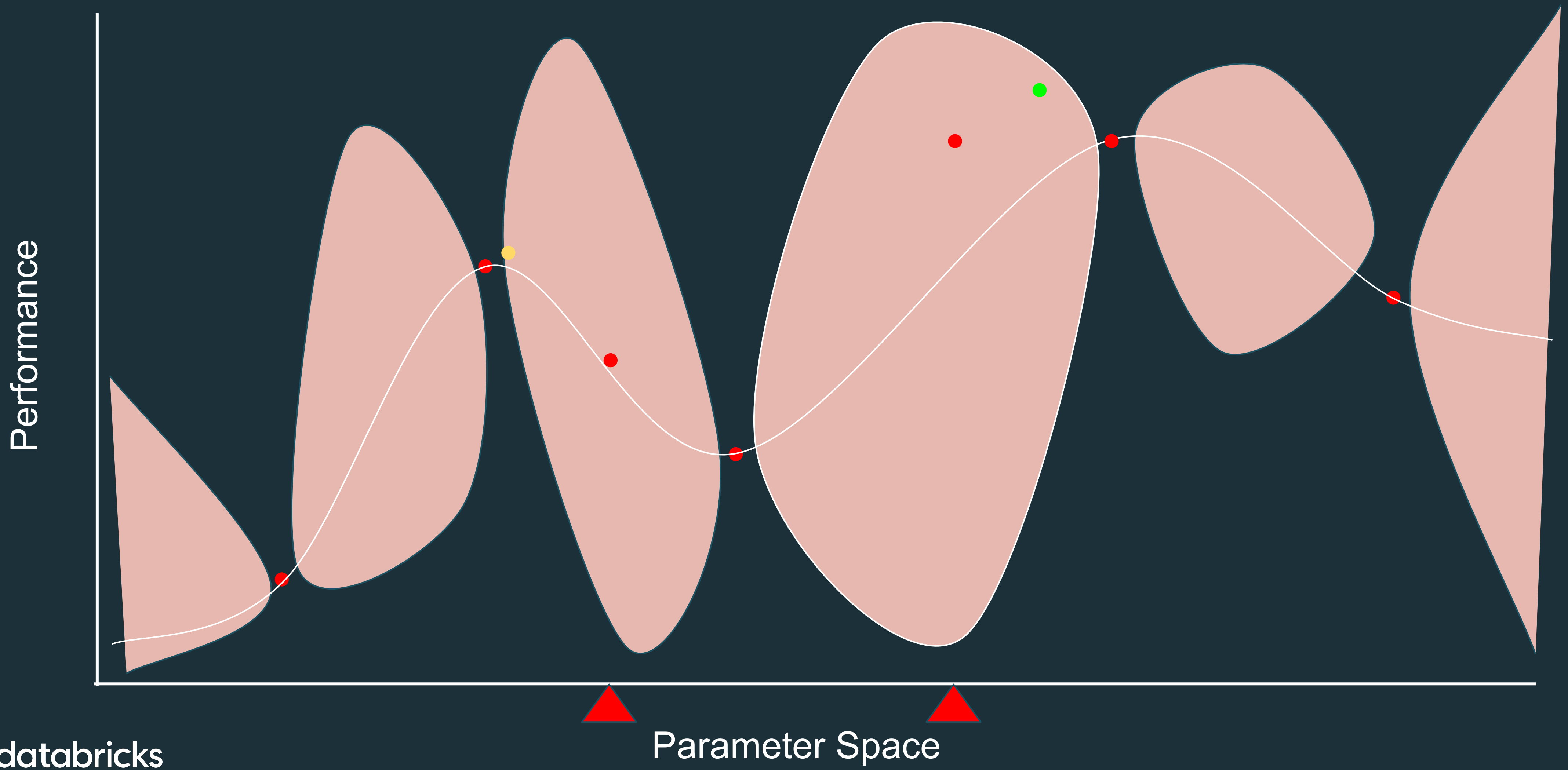


Bayesian Optimization



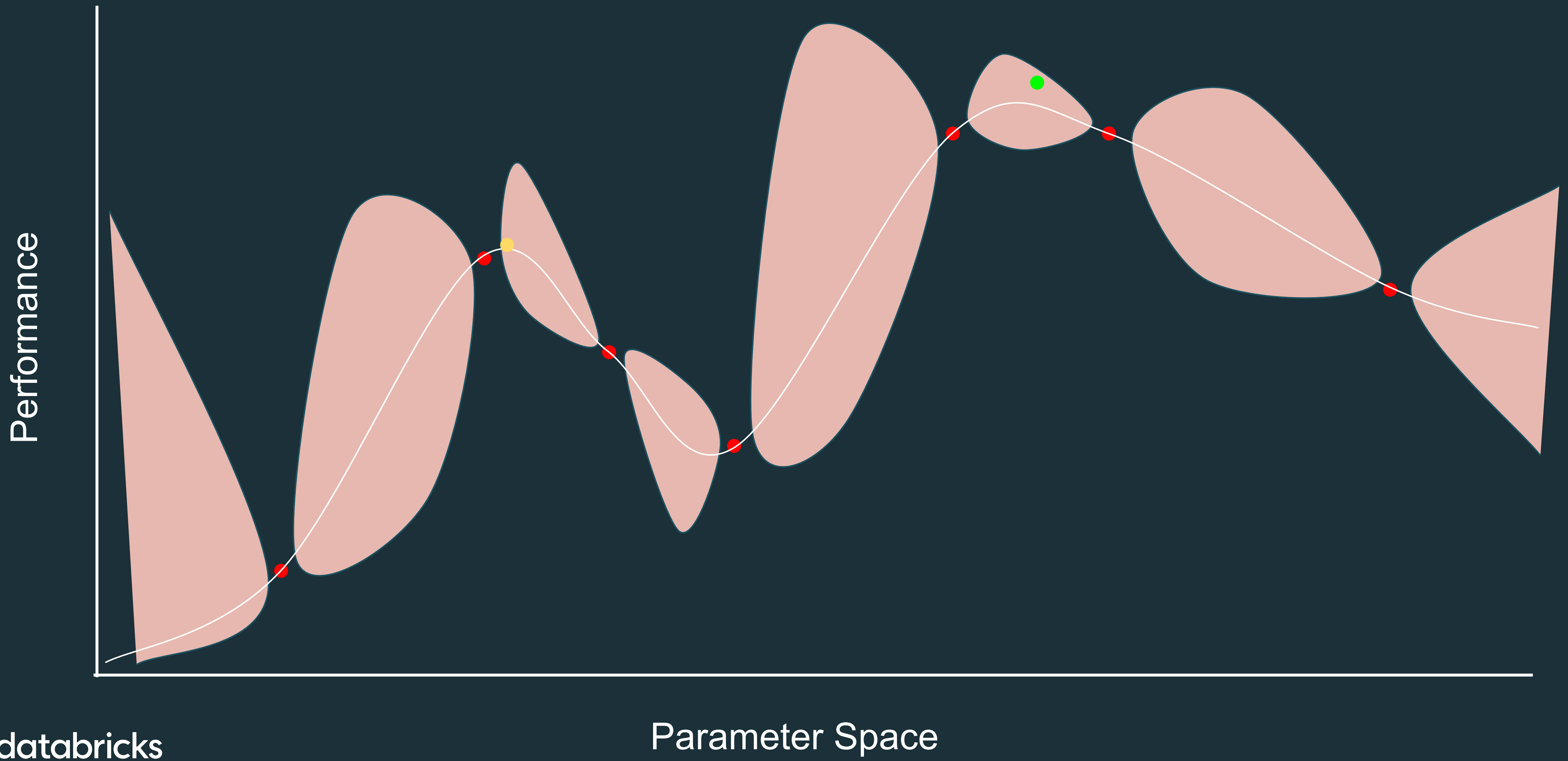
databricks

Bayesian Optimization



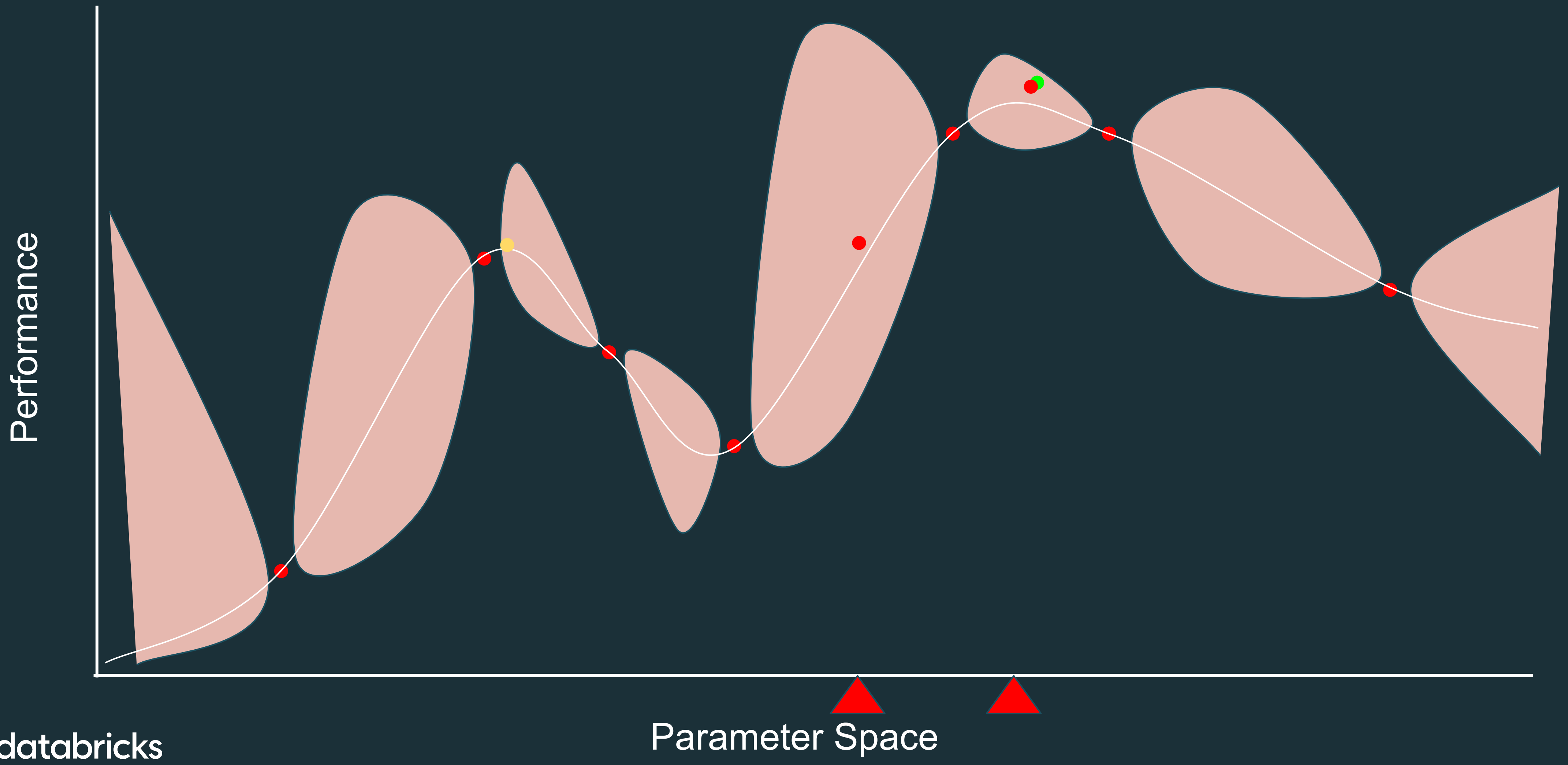
databricks

Bayesian Optimization



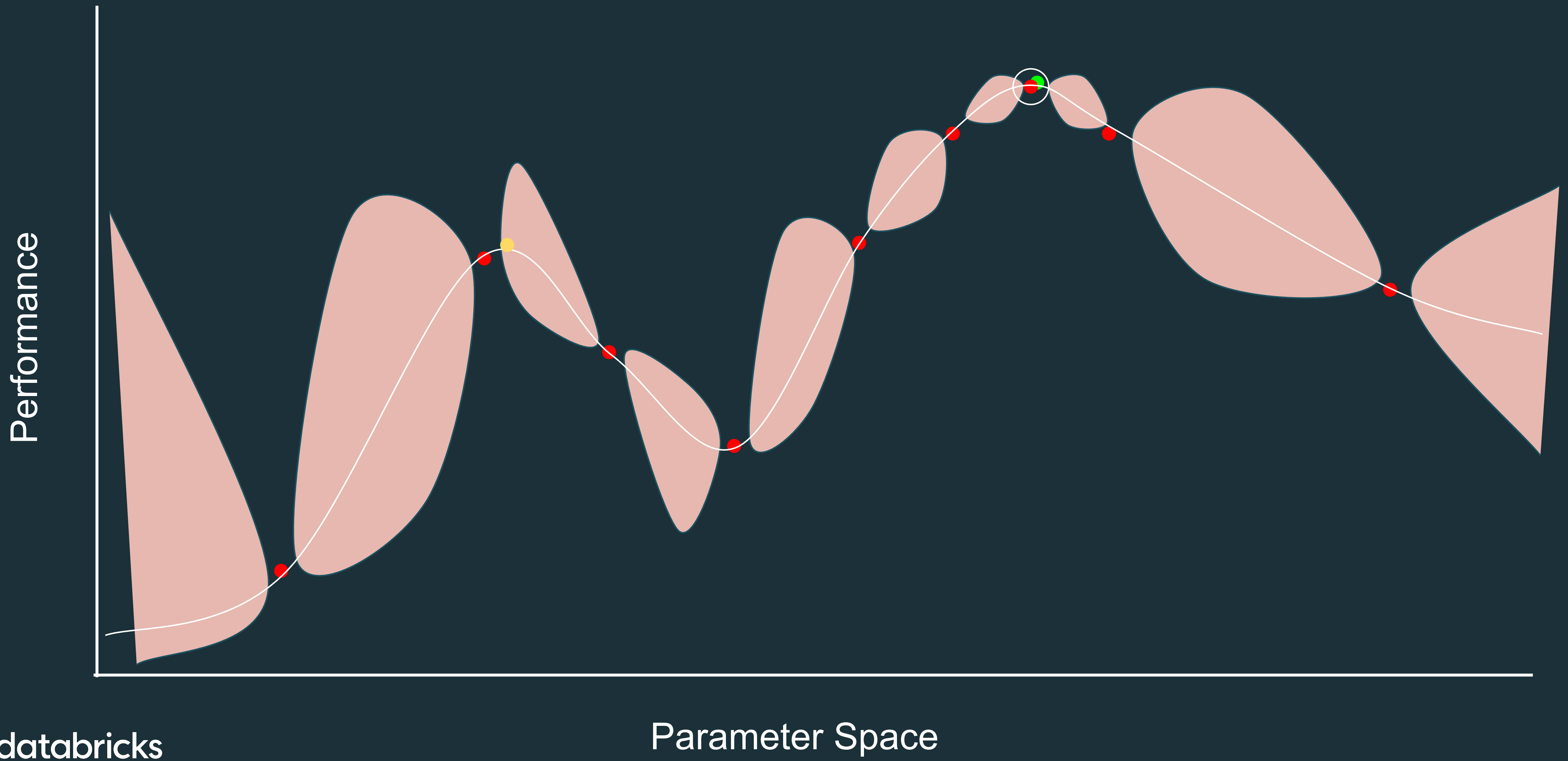
databricks

Bayesian Optimization



databricks

Bayesian Optimization



databricks

Working with Time Series

Approach

No Magic...

- Use standard frameworks
  - Use spark based frameworks (eg. Flint) where available!
- Use FCN/CNN when you can, use RNNs/LSTMs/GRUs if you have to

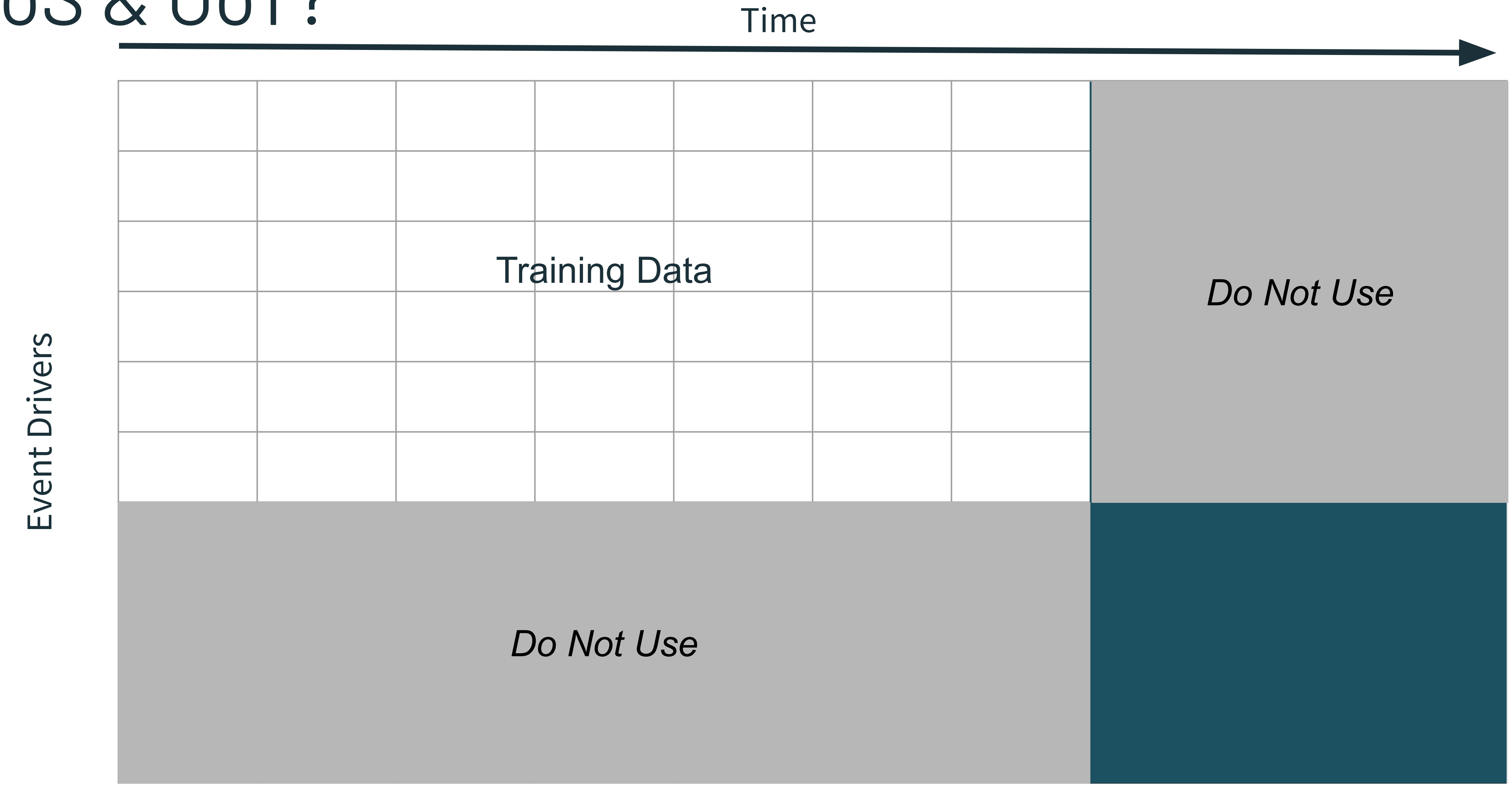
But...

- Our best practices can enable you to do this more efficiently

What Kind of Problem Do You Have?

Type	Order of Events	Relative Timing	Absolute Time	Validation Requirements
Sequence	Yes	No	No	OoS
History Match	Yes	Yes	No	OoT
Forecasting	Yes	Yes	Yes	OoS & OoT

OoS & OoT?



Scenario

An AAV needs to travel from point A to point B. It is a non-trivial distance, with significant topological features, and at a minimum will take two days (local time).

There is an instrument that after some accumulated impact requires recalibration. This requires significant downtime and must be avoided.

The objective is to get from A to B as fast as possible, without having to recalibrate the instrument.

Disclaimer: This scenario is completely contrived, based entirely on my own ignorance, and used only because Thomas mentioned “Rovers” on our prep call — and I miss @SarcasticRover. Please don’t throw rocks at me if I say something stupid!

Sequence

Order Matters, but Time (Relative or Absolute) does not

Scenario modification: None

Explanation: The scenario as described is a Sequence problem. At any given point, I could stop the AAV dead in its tracks, and nothing in the model's expected route and velocity predictions would change.

History Match

Order and Relative Time Matter, Absolute Time does not

Scenario modification: The instrument is able to “heal” itself at some rate

Explanation: Now the time between impacts matters because of the ability to heal, so my model should account for both the impact, and time between impacts, when determining the route and velocity.

Forecasting

Order, Relative Time, and Absolute Time all matter

Scenario modification: The healing rate is heavily dependent on the amount of solar power available to the instrument.

Explanation: Now, in addition to the time between impacts, my model also needs access to absolute time and potentially external data (expected weather) to calculate solar power availability, and thus the modified healing rate, to predict a route and velocity.

Conclusion

- Databricks offers a mix of tooling and best practices to solve these problems more efficiently
- For Sequence and History Match problems, there is literature suggesting you can use CNNs to get similar results to RNNs
- For Forecasting, RNN/LSTM/GRU are probably the way to go
- Just as important: match your validation strategy to the problem type!

SAS Migration Slides

How to Migrate from SAS to Databricks

Replace

Migrate SAS code to PySpark/SparkR/SQL:

- Translate SAS code to use Spark native APIs
- Good for small teams, new applications

Integrate

SAS can connect to Databricks via SQL:

- Reduce barrier to get started by accelerating existing SAS code
- Good for large teams, validated applications

How to Migrate from SAS to Databricks

1

IDENTIFY NOVEL USE CASES

Identify a current initiative where SAS is a poor fit and where the team has skills for a non-SAS implementation

2

PORT EASY-TO-MIGRATE USE CASES

Identify SAS workloads that leverage lots of SQL commands which can easily be ported via ProcSQL and other SAS built-ins or analogs in R

3

PORT HEAVIER USE CASES

For use cases that are embedded in SAS, invest in larger code translation and staff retraining initiative

Why companies migrate from SAS to □

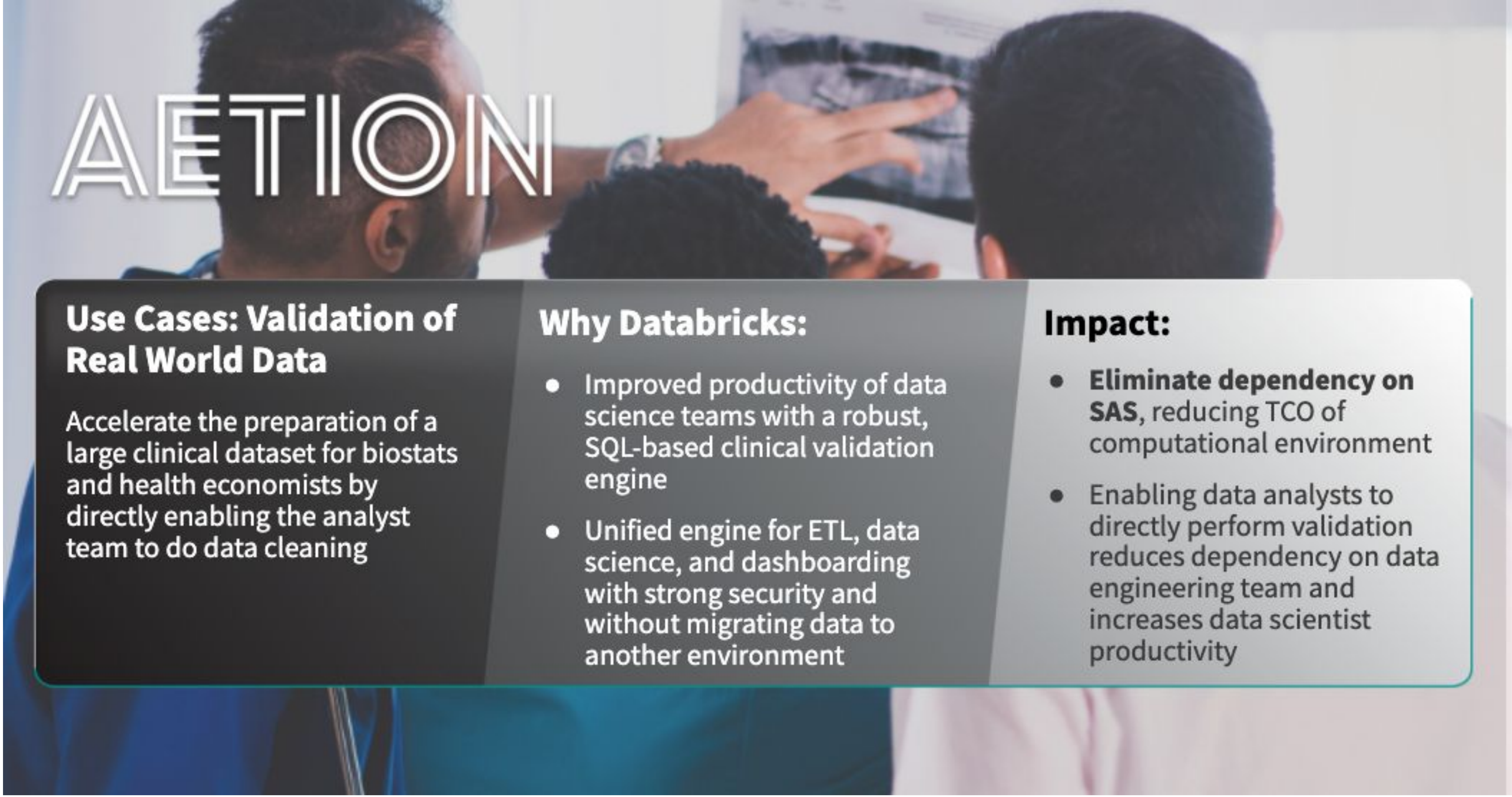
1. GROWTH

(Revenue impact)

- ❖ Deliver ML innovations to market faster
- ❖ Provide data insights for better business decision

AETION

Robust rule validation framework allows Aetion to build larger data assets, accelerating sales to external data vendors



AETION

Use Cases: Validation of Real World Data

Accelerate the preparation of a large clinical dataset for biostats and health economists by directly enabling the analyst team to do data cleaning

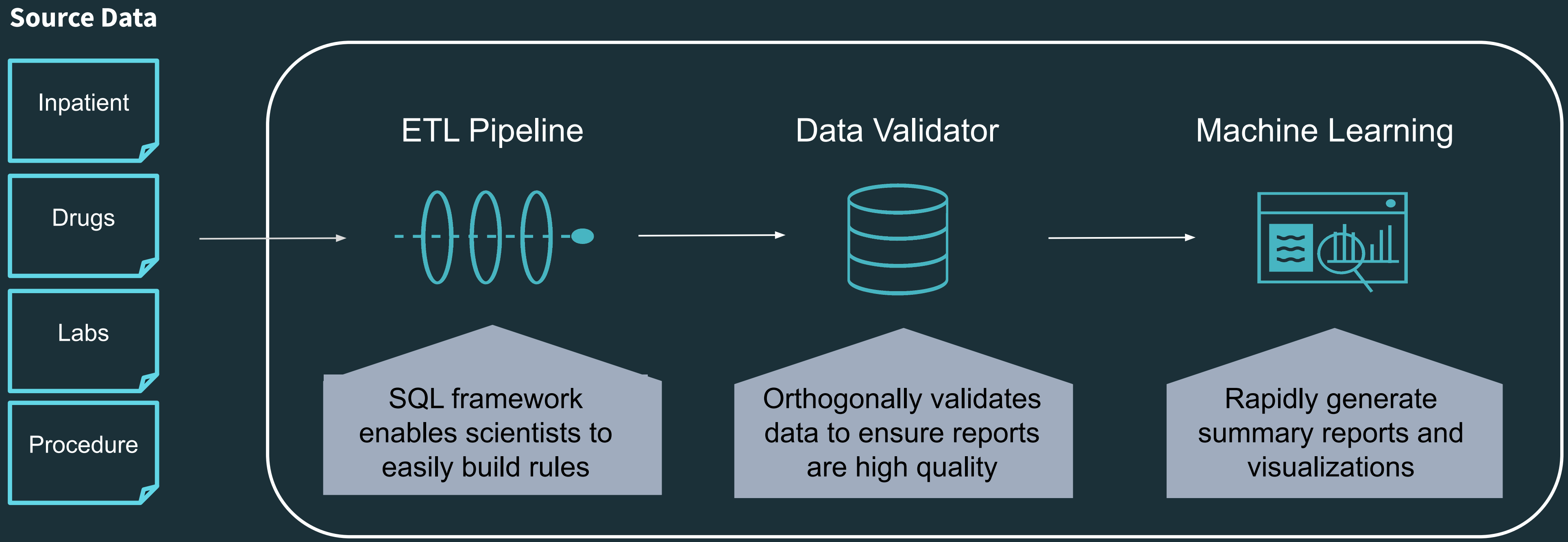
Why Databricks:

- Improved productivity of data science teams with a robust, SQL-based clinical validation engine
- Unified engine for ETL, data science, and dashboarding with strong security and without migrating data to another environment

Impact:

- **Eliminate dependency on SAS**, reducing TCO of computational environment
- Enabling data analysts to directly perform validation reduces dependency on data engineering team and increases data scientist productivity

Replacement: Rapid Rule Engine for RWD



Source Data

Inpatient

Drugs

Labs

Procedure

ETL Pipeline

SQL framework enables scientists to easily build rules

Data Validator

Orthogonally validates data to ensure reports are high quality

Machine Learning

Rapidly generate summary reports and visualizations

RESULTS

- Before Databricks: SAS environment limited where analysis could be run and inhibited building a robust RVF engine
- On Databricks: Spark SQL enables robust data validation while powering both interactive analytics and ML

databricks

Blog at https://medium.com/aetion-technology/building-a-rule-based-validation-framework-rvf-for-real-world-healthcare-data-3dc05d661382

Why companies migrate from SAS to

1. GROWTH

(Revenue impact)

- ❖ Deliver ML innovations to market faster
- ❖ Provide data insights for better business decision

2. PROFIT

(Bottom Line Savings)

- ❖ Unlock infrastructure savings through automation
- ❖ Increase data team productivity through one unified platform

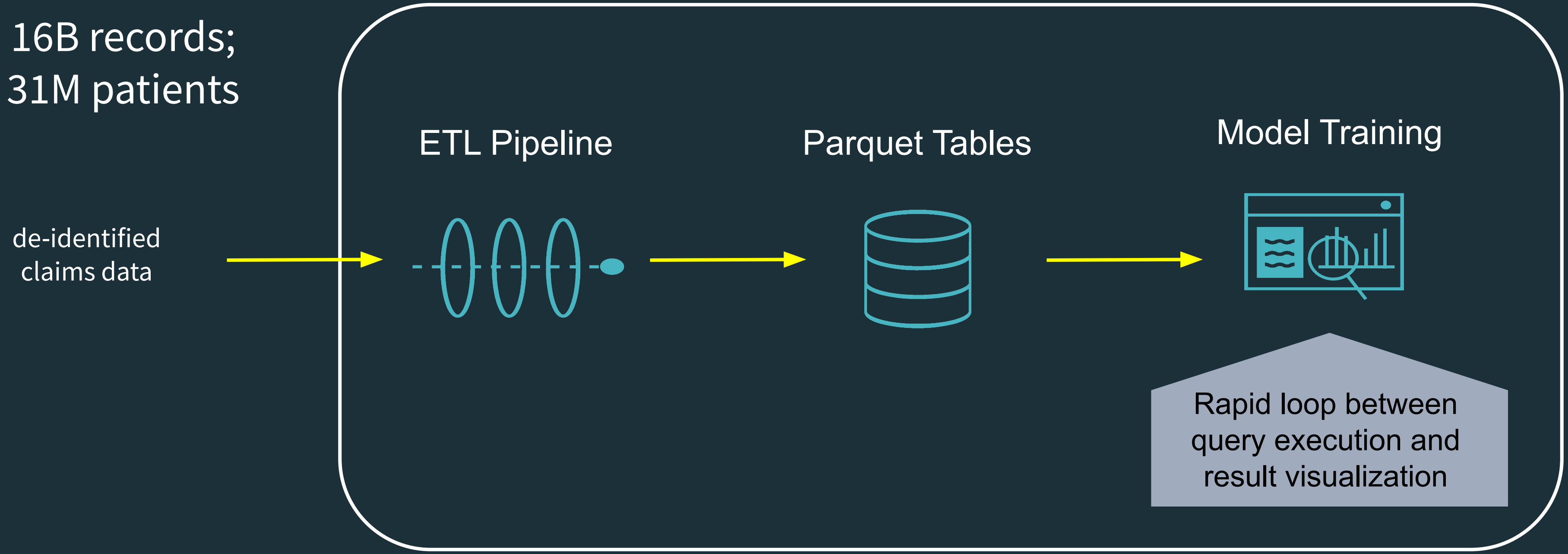
AETION

Robust rule validation framework allows Aetion to build larger data assets, accelerating sales to external data vendors

Major Biopharma

Eliminated dependency on failure-prone HPC system, saving $75k in outage costs per month and 4 months of schedule risk, while also eliminating homegrown system for managing PHI

Replacement: Interactive Query on Terabyte-scale RWE


16B records;
31M patients
de-identified claims data
ETL Pipeline
Parquet Tables
Model Training
Rapid loop between query execution and result visualization

RESULTS

- Prior to Databricks: >1TB dataset failed when importing into SAS
- On Databricks: Common queries (top ICD code by year, subselect patient cohorts) execute interactively, data can be segregated using ACLs to meet data use requirements

Why companies migrate from SAS to

1. GROWTH
(Revenue impact)

- ❖ Deliver ML innovations to market faster
- ❖ Provide data insights for better business decision

2. PROFIT
(Bottom Line Savings)

- ❖ Unlock infrastructure savings through automation
- ❖ Increase data team productivity through one unified platform

3. RISK
(Insulation and reduction)

- ❖ Threat and fraud detection, and response at scale
- ❖ Comprehensive cloud based data security, governance and certifications

AETION

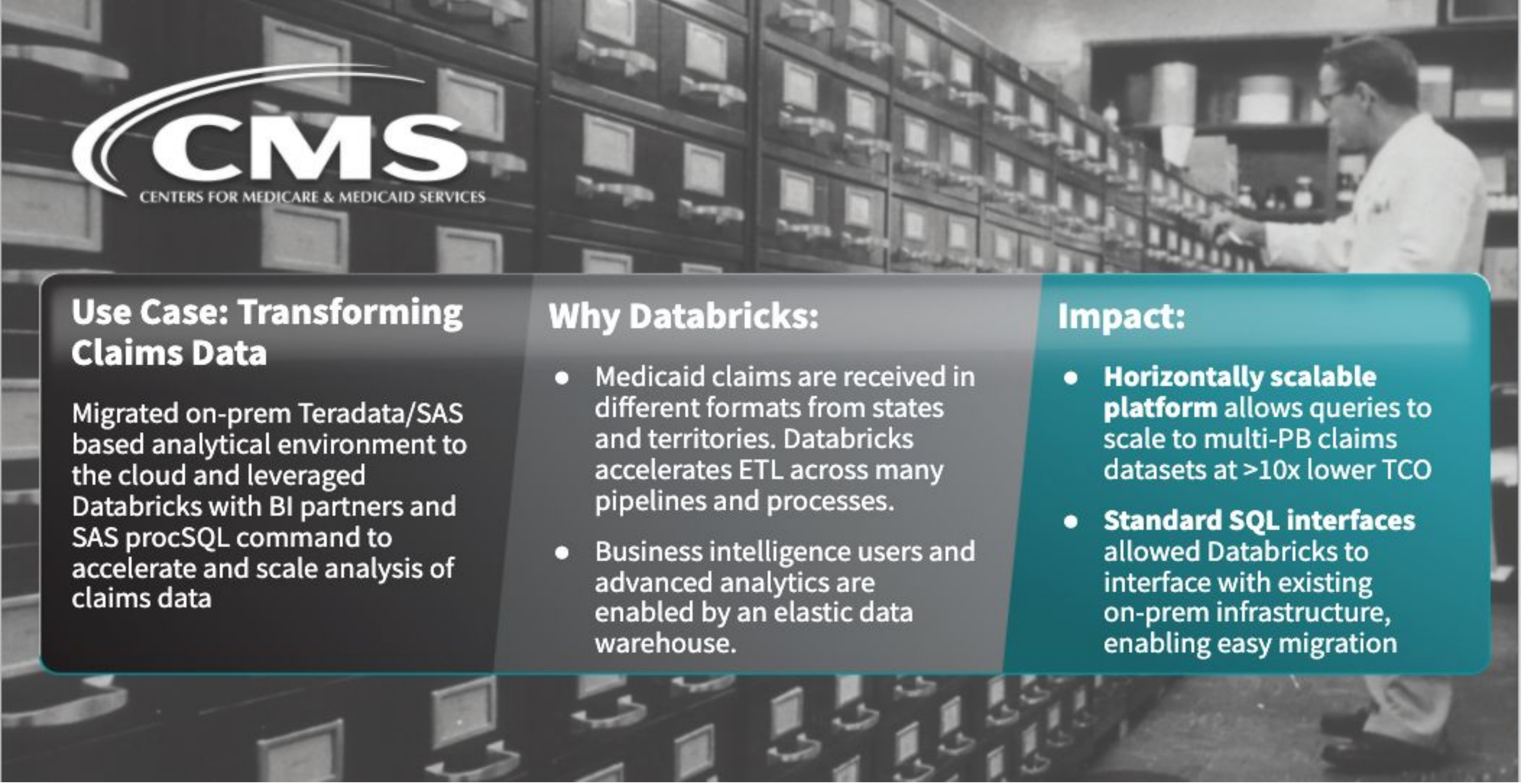
Robust rule validation framework allows Aetion to build larger data assets, accelerating sales to external data vendors

Major Biopharma

Eliminated dependency on failure-prone HPC system, saving $75k in outage costs per month and 4 months of schedule risk, while also eliminating homegrown system for managing PHI

CMS
CENTERS FOR MEDICARE & MEDICAID SERVICES

Databricks platform meets the complex compliance needs of working on large scale medicare/medicaid datasets



CMS

CENTERS FOR MEDICARE & MEDICAID SERVICES

Use Case: Transforming Claims Data

Migrated on-prem Teradata/SAS based analytical environment to the cloud and leveraged Databricks with BI partners and SAS procSQL command to accelerate and scale analysis of claims data

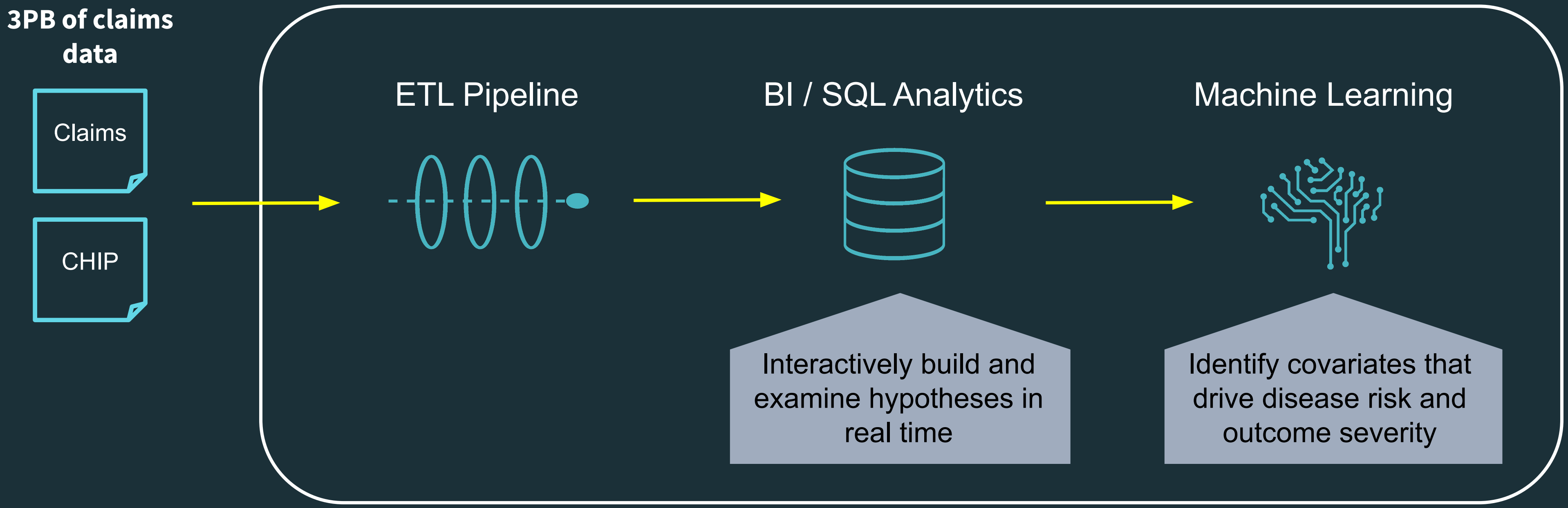
Why Databricks:

- Medicaid claims are received in different formats from states and territories. Databricks accelerates ETL across many pipelines and processes.
- Business intelligence users and advanced analytics are enabled by an elastic data warehouse.

Impact:

- **Horizontally scalable platform** allows queries to scale to multi-PB claims datasets at >10x lower TCO
- **Standard SQL interfaces** allowed Databricks to interface with existing on-prem infrastructure, enabling easy migration

Digital Transformation of Claims Data



3PB of claims data

Claims

CHIP

ETL Pipeline

BI / SQL Analytics

Machine Learning

Interactively build and examine hypotheses in real time

Identify covariates that drive disease risk and outcome severity

RESULTS

- Before Databricks: unable to do advanced visualization or analytics without exporting data from system
- On Databricks: ETL, visualization, and ML are all available within a single platform, can build self-documenting data products

Video at https://youtu.be/M0TcIVEEPZ0?t=7780

How to Migrate from SAS to Databricks

Replace

Partner with Databricks Migration Factory and SI Ecosystem on a strategy:

- Identify best workloads to move
- Spark experts ensure best practices for translating workloads

Integrate

Leverage built-in SQL connectors for rapid integration:

- Works with minimal setup
- Databricks engineers provide assistance with complex setups

How to Migrate from SAS to Databricks

1

PICK A USE CASE(s)

Identify a current initiative the will drive business value and the success metrics to be measured by the business

2

ENABLE DATABRICKS PLATFORM with SECURE DATA

Ensure datasets are in a secure within cloud environment or can be accessed securely from the cloud

3

CO-DEVELOP A MIGRATION PLAN

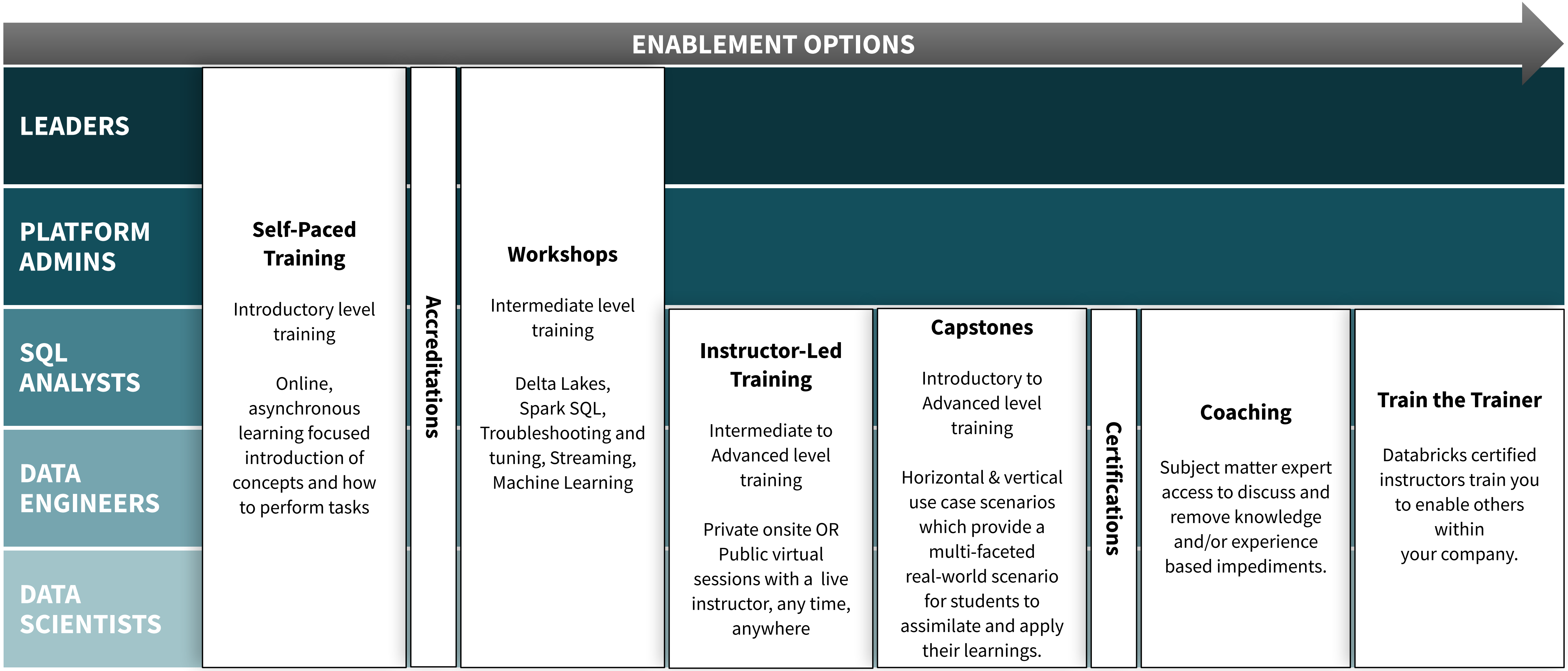
Databricks can integrate as a SQL backend to an existing SAS deployment. For code migration, Databricks and partners can help you to move.

SAS Statements and Pyspark Equivalents

SAS	Description	Python API
DATA Statement	Creates either a SAS data file, a data set that holds actual data, or a SAS data view, a data set that references data that is stored elsewhere	spark.read.<fill in> Ex: spark.read.csv()
PROC CONTENTS	Shows the contents of a SAS data set	df.show(), df.printSchema()
PROC CORR	Computes Pearson correlation coefficients	pyspark.ml.stat.Correlation pyspark.mllib.stat.Statistics.Corr
PROC TRANSPOSE	Creates an output data set by restructuring the values in a SAS data set, transposing selected variables into observations	.groupBy().pivot()
PROC SORT	Orders SAS data set observations by the values of one or more character or numeric variable	.orderBy()
PROC DELETE	Delete a list of data sets	Not necessary
PROC SQL	Can sort, summarize, subset, join (merge), and concatenate datasets, create new variables, and print the results or create a new table or view	Pyspark.sql.functions Ex: df.filter(), df.join(), df.withColumn(), df.concat(), df.agg(sum()) df.orderBy()
PROC SUMMARY	Provides data summarization tools that compute descriptive statistics	pyspark.ml.stat.Summarizer
PROC DATASETS	Utility procedure that manages your SAS files	pyspark.sql.functions
PROC EXPORT	Reads data from a SAS data set and writes it to an external data source	df.write.format() Ex: df.write.format('csv')

Training Roadmaps

Customer / Partner Enablement Journey



ENABLEMENT OPTIONS

LEADERS

PLATFORM ADMINS

SQL ANALYSTS

DATA ENGINEERS

DATA SCIENTISTS

Self-Paced Training

Introductory level training

Online, asynchronous learning focused introduction of concepts and how to perform tasks

Accreditations

Workshops

Intermediate level training

Delta Lakes, Spark SQL, Troubleshooting and tuning, Streaming, Machine Learning

Instructor-Led Training

Intermediate to Advanced level training

Private onsite OR Public virtual sessions with a live instructor, any time, anywhere

Capstones

Introductory to Advanced level training

Horizontal & vertical use case scenarios which provide a multi-faceted real-world scenario for students to assimilate and apply their learnings.

Certifications

Coaching

Subject matter expert access to discuss and remove knowledge and/or experience based impediments.

Train the Trainer

Databricks certified instructors train you to enable others within your company.

databricks

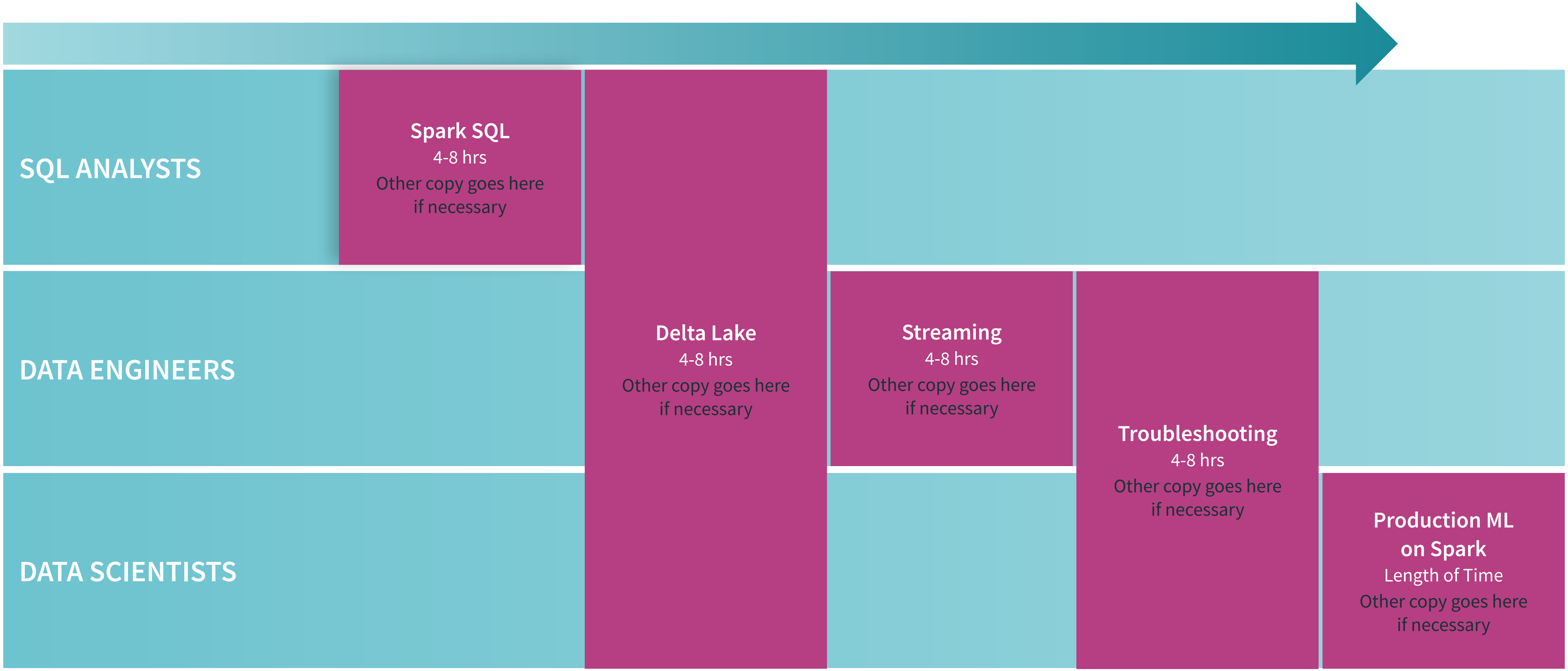
Free eLearning - Upcoming

BUSINESS LEADERS	**Introduction to Data Science** 1 hour Optional Online Cert Exam: Data Science Fundamentals	**Introduction to Unified Analytics** 1 hour Optional Online Cert Exam: Unified Analytics Fundamentals	**Introduction to Delta** 1 hour Optional Online Cert Exam: Delta Fundamentals							
PLATFORM ADMINS				**Databricks Administration: Account Setup** 1 hour	**Databricks Administration: Admin Console** 1 hour	**Databricks Security: Access Control** 1 hour	**Databricks Architecture** 1 hour	**Cluster Best Practices** 0.5 hours		
SQL ANALYSTS				**Getting Started with Apache Spark SQL** 6 hours	**Databricks Platform: Data Analyst** 1 hour		**DataFrames (Intro Spark Programming)** 1 hour	**Fundamentals of SQL on Databricks** 1 hour		
DATA ENGINEERS				**Databricks Platform: Python Developer** 1 hour	**Databricks Platform: Data Engineer** 1 hour	**Databricks Security: Access Control** 1 hour		**Fundamentals of Delta** 1.5 hours Optional Online Cert Exam: Delta Accreditation	**Delta Rapid Start** 1.5 hours	**CI/CD Part 1** 1.5 hours
DATA SCIENTISTS				**Databricks Platform: Data Scientist** 1 hour						

databricks

Free eLearning | Free Workshops | Instructor-Led Training (Paid)

Intermediate Training - Workshops



SQL ANALYSTS

DATA ENGINEERS

DATA SCIENTISTS

Spark SQL
4-8 hrs
Other copy goes here if necessary

Delta Lake
4-8 hrs
Other copy goes here if necessary

Streaming
4-8 hrs
Other copy goes here if necessary

Troubleshooting
4-8 hrs
Other copy goes here if necessary

Production ML on Spark
Length of Time
Other copy goes here if necessary

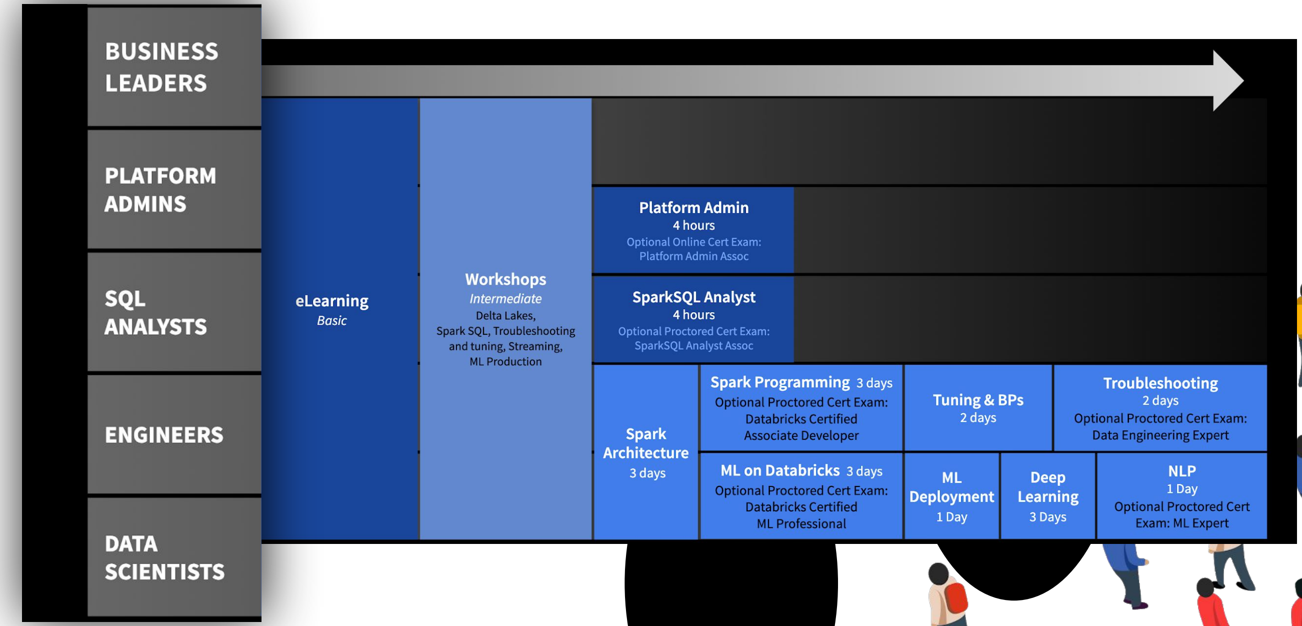
databricks

Free eLearning	Free Workshops	Instructor-Led Training (Paid)

Data + AI Customer Partnerships

Building Data + AI Experts

Persona-Based eLearning, Workshops, and Instructor-Led Training



BUSINESS LEADERS

PLATFORM ADMINS

SQL ANALYSTS

ENGINEERS

DATA SCIENTISTS

eLearning
Basic

Workshops
Intermediate
Delta Lakes,
Spark SQL, Troubleshooting and tuning, Streaming, ML Production

Platform Admin
4 hours
Optional Online Cert Exam:
Platform Admin Assoc

SparkSQL Analyst
4 hours
Optional Proctored Cert Exam:
SparkSQL Analyst Assoc

Spark Architecture
3 days

Spark Programming 3 days
Optional Proctored Cert Exam:
Databricks Certified
Associate Developer

Tuning & BPs
2 days

Troubleshooting
2 days
Optional Proctored Cert Exam:
Data Engineering Expert

ML on Databricks 3 days
Optional Proctored Cert Exam:
Databricks Certified
ML Professional

ML Deployment
1 Day

Deep Learning
3 Days

NLP
1 Day
Optional Proctored Cert
Exam: ML Expert

Data + AI Community

Workshops

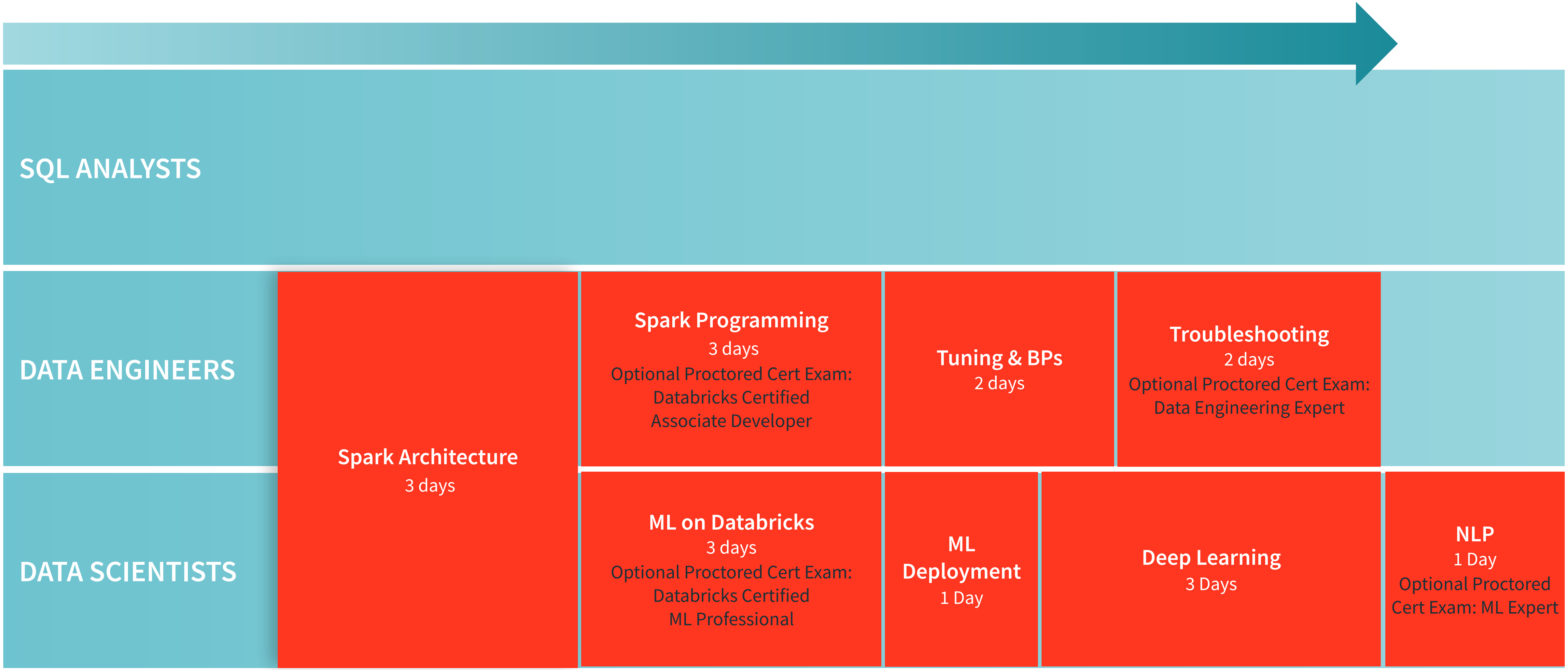
Hackathons

SPARK+AI SUMMIT 2019

Customer Advisory Boards

databricks

Instructor-Led Training



SQL ANALYSTS

DATA ENGINEERS

DATA SCIENTISTS

Spark Architecture
3 days

Spark Programming
3 days
Optional Proctored Cert Exam:
Databricks Certified
Associate Developer

Tuning & BPs
2 days

Troubleshooting
2 days
Optional Proctored Cert Exam:
Data Engineering Expert

ML on Databricks
3 days
Optional Proctored Cert Exam:
Databricks Certified
ML Professional

ML Deployment
1 Day

Deep Learning
3 Days

NLP
1 Day
Optional Proctored
Cert Exam: ML Expert

databricks

Free eLearning | Free Workshops | Instructor-Led Training (Paid)

databricks