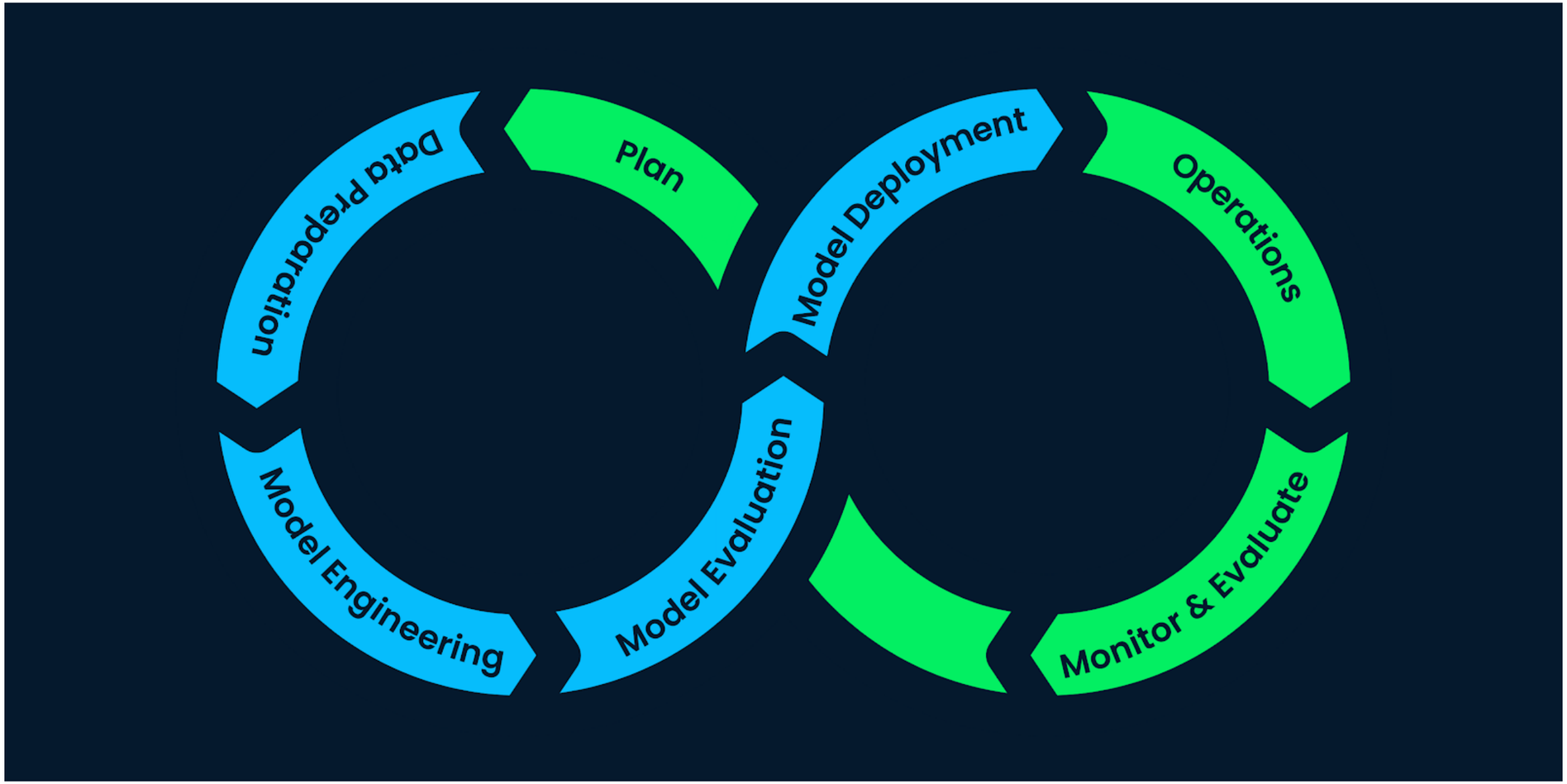
What is MLflow?

INTRODUCTION TO MLFLOW

Weston Bassler
Senior MLOps Engineer

datacamp

The machine learning lifecycle



[1] datacamp.com

Difficulties of machine learning

- **Tracking:**
  - Models and model metrics
- **Reproducibility:**
  - Platforms, environments and workspaces
- **Deployment:**
  - So many options and no standards

What is MLflow?

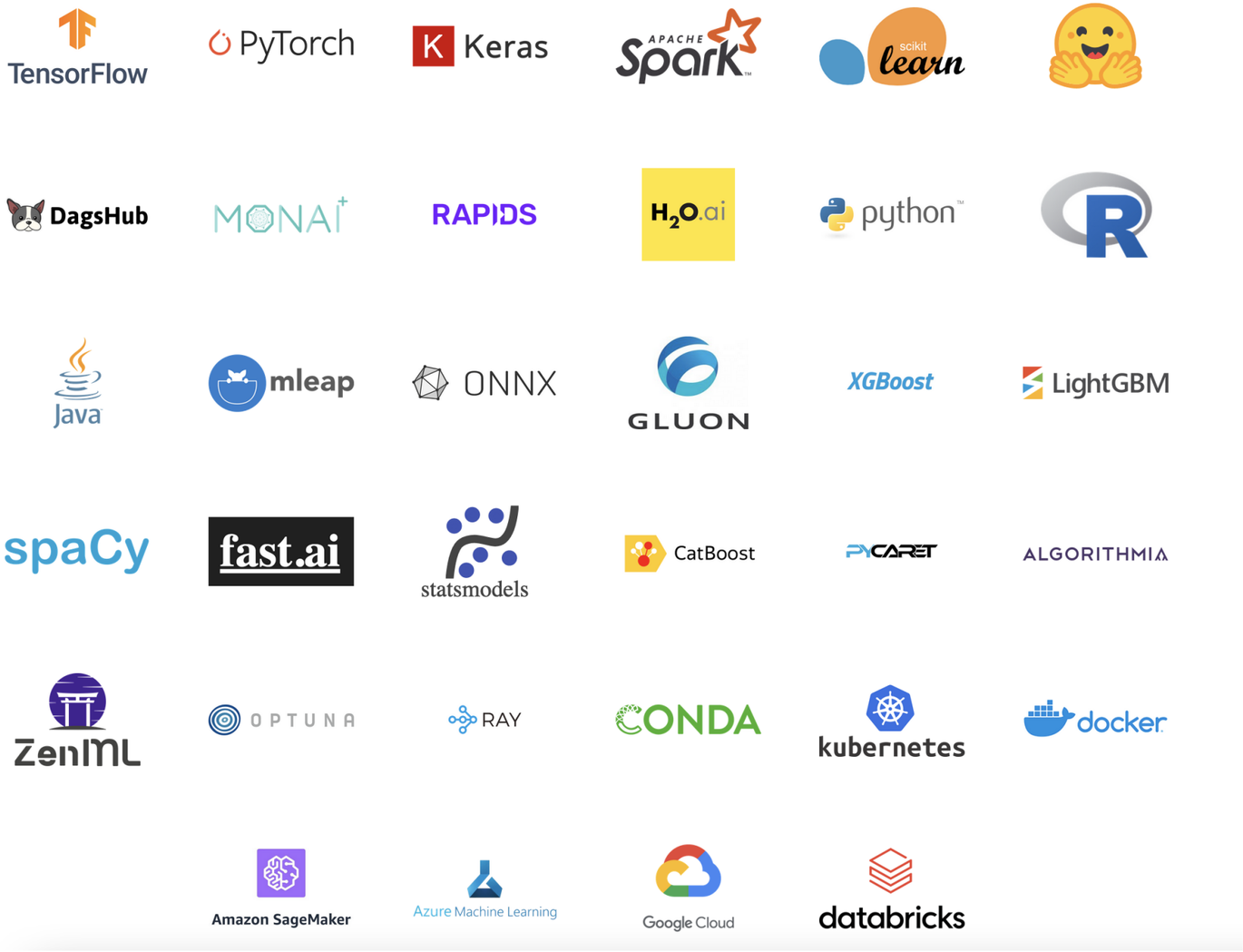
"An open source platform for the machine learning lifecycle" - **MLflow.org**

[1] www.mlflow.org

Components of MLflow

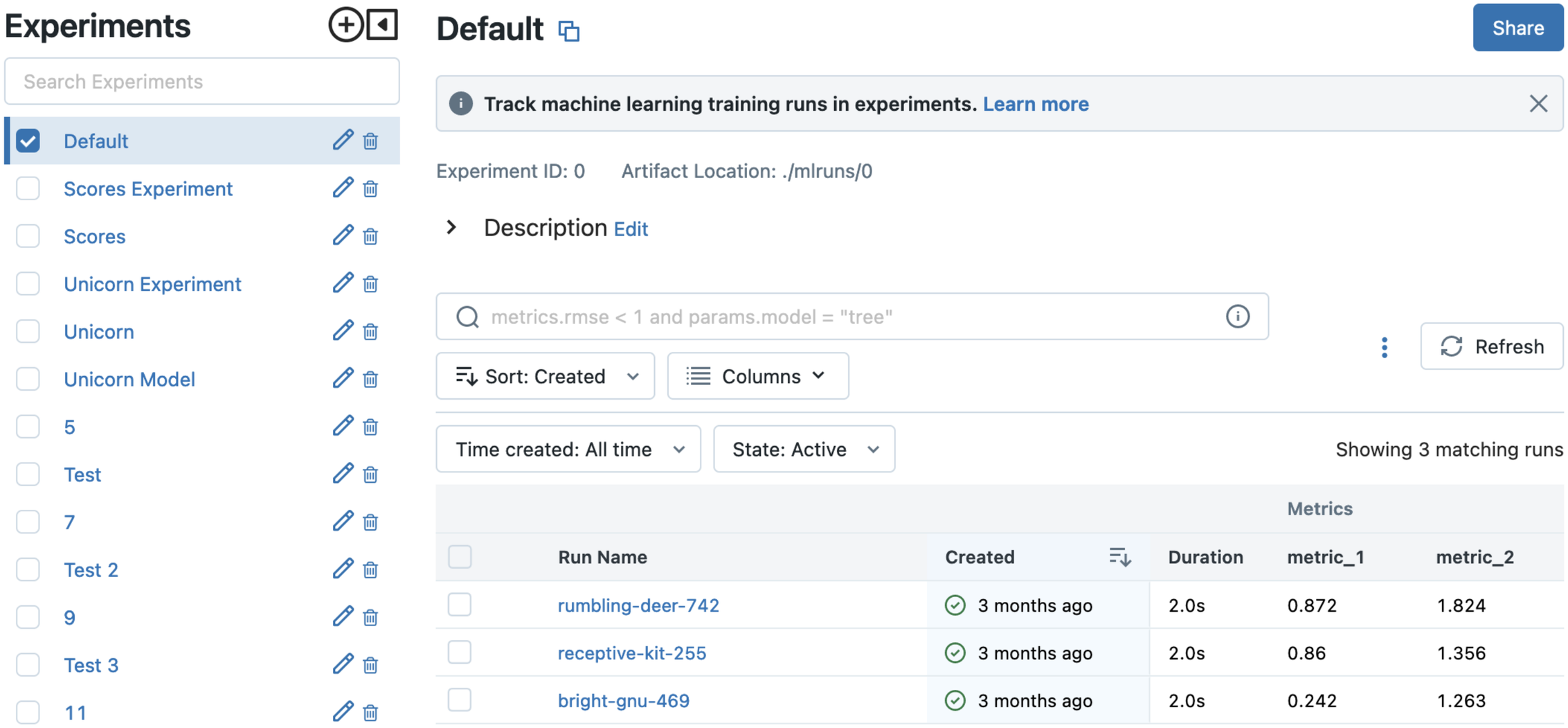
- **MLflow Tracking:**
  - Record metrics and parameters from training runs
  - Query data from experiments
  - Store models, artifacts and code
- **MLflow Models:**
  - Standardize models for deployment
  - Build customized models
- **Model Registry:**
  - Store and version ML models
  - Load and deploy ML models
- **MLflow Projects:**
  - Package ML code for reproducibility
  - Package ML code for repeatability

Integrations with:



[1] www.mlflow.org

MLflow experiments



Working with experiments

MLflow Client

- Create Experiments

```
client.create_experiment("Name")
```

- Tag Experiments

```
client.set_experiment_tag("Name",
k, v)
```

- Delete Experiments

```
client.delete_experiment("Name")
```

MLflow module

- Create Experiments

```
mlflow.create_experiment("Name")
```

- Tag Experiments

```
mlflow.set_experiment_tag(k, v)
```

- Delete Experiments

```
mlflow.delete_experiment("Name")
```

- Set Experiment

```
mlflow.set_experiment("Name")
```

Starting a new experiment

```
import mlflow
# Create new Experiment
mlflow.create_experiment("My Experiment")
# Tag new experiment
mlflow.set_experiment_tag("scikit-learn", "lr")
# Set the experiment
mlflow.set_experiment("My Experiment")
```

Let's practice!

INTRODUCTION TO MLFLOW

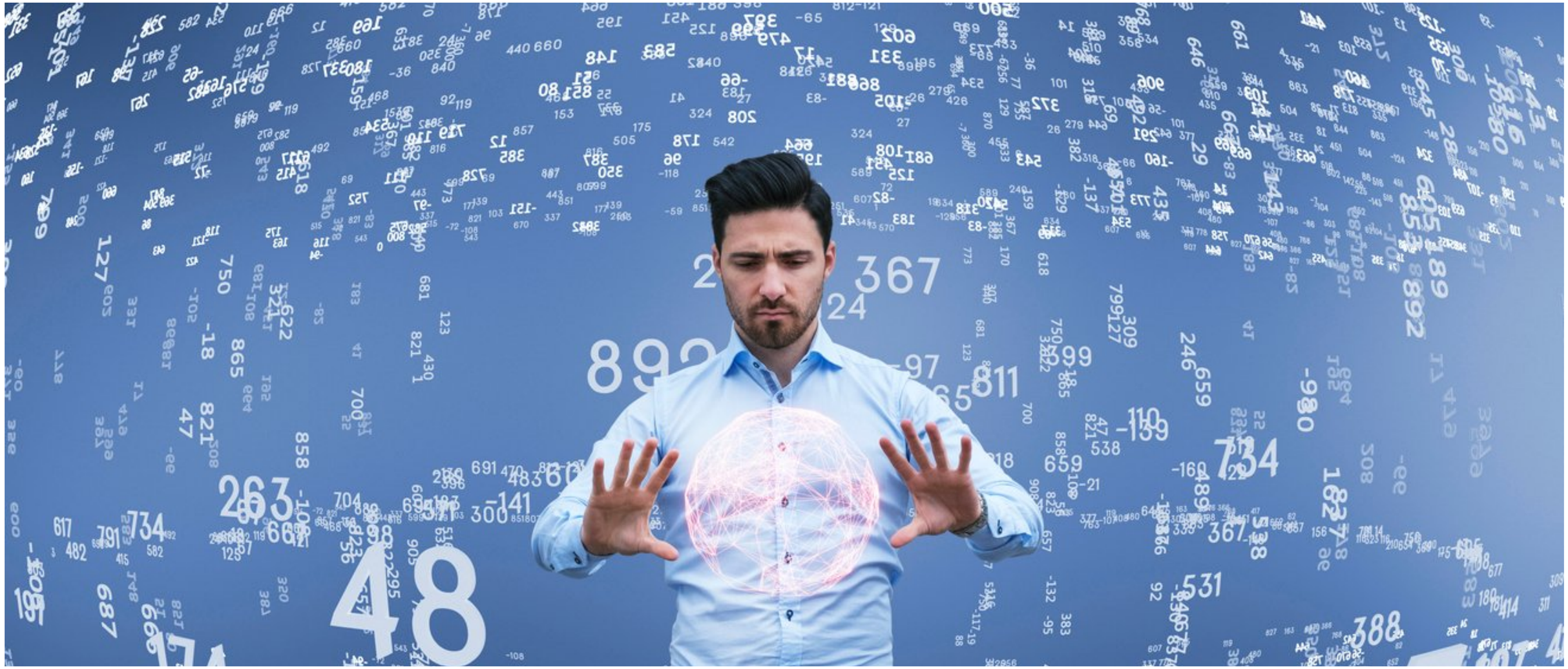
MLflow Tracking

INTRODUCTION TO MLFLOW

Weston Bassler
Senior MLOps Engineer

datacamp

Tracking data about models



[1] istock.com

What is MLflow Tracking?

- Model Metrics
  - F1, Recall, Accuracy, MSE, etc...
- Parameters
  - library specific
- code
  - `train.py`
- other artifacts
  - tokenizers, pickle, etc...

Training runs

- How MLflow is organized
- New run equals new model training
- A run is placed within an experiment
- Invoked via `mlflow.start_run()`

[1] unsplash.com

Starting a training run

```
import mlflow


# Start a run
mlflow.start_run()
```

```
<ActiveRun: >
```

```
# End a run
mlflow.end_run()
```

Setting a training run variable

```
import mlflow
# Set experiment
mlflow.set_experiment("My Experiment")
# Start a run
run = mlflow.start_run()
# Print run info
run.info
```

```
<RunInfo: artifact_uri='./mlruns/0/9de5df4d19994546b03dce09aefb58af/artifacts',
  end_time=None, experiment_id='31', lifecycle_stage='active',
  run_id='9de5df4d19994546b03dce09aefb58af', run_name='big-owl-145',
  run_uuid='9de5df4d19994546b03dce09aefb58af', start_time=1676838126924,
  status='RUNNING', user_id='user'>
```

Logging to MLflow Tracking

- **Metrics**
  - `log_metric("accuracy", 0.90)`
  - `log_metrics({"accuracy": 0.90, "loss": 0.50})`
- **Parameters**
  - `log_param("n_jobs", 1)`
  - `log_params({"n_jobs": 1, "fit_intercept": False})`
- **Artifacts**
  - `log_artifact("file.py")`
  - `log_artifacts("./directory/")`

Logging a run

```
import mlflow
# Set Experiment
mlflow.set_experiment("LR Experiment")

# Start a run
mlflow.start_run()

# Model Training Code here
lr = LogisticRegression(n_jobs=1)

# Model evaluation Code here
lr.fit(X, y)
score = lr.score(X, y)
```

```
# Log a metric
mlflow.log_metric("score", score)

# Log a parameter
mlflow.log_param("n_jobs", 1)

# Log an artifact
mlflow.log_artifact("train_code.py")
```

Open MLflow UI

```
# Open MLflow Tracking UI
mlflow ui
```

Go to: http://localhost:5000

Tracking UI experiment view

LR Experiment

Share

Experiment ID: 37 Artifact Location: ./mlruns/37

Description Edit

metrics.rmse < 1 and params.model = "tree"

Sort: Created

Refresh

Columns

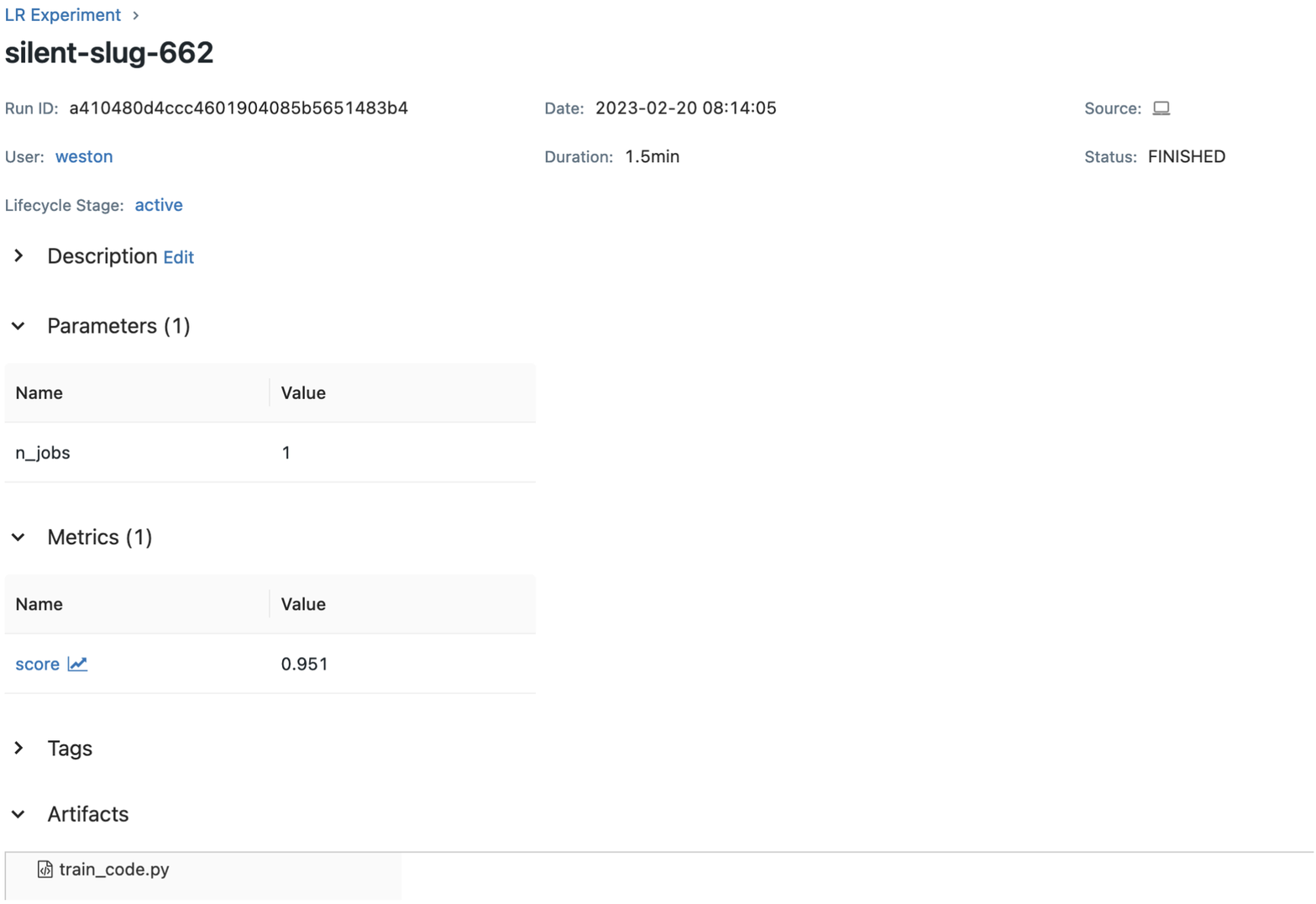
Time created: All time

State: Active

Showing 1 matching run

				Metrics	Parameters
	Run Name	Created	Models	score	n_jobs
	silent-slug-662	1 minute ago	-	0.951	1

Tracking UI run view



Let's practice!

INTRODUCTION TO MLFLOW

Querying runs

INTRODUCTION TO MLFLOW

Weston Bassler
Senior MLOps Engineer

Model data

[1] istock.com

Runs data

Insurnace Experiment

Experiment ID: 27 Artifact Location: ./mlruns/27

Description Edit

metrics.rmse < 1 and params.model = "tree" Sort: accuracy_score Columns

Time created: All time State: Active

			Metrics						
	Run Name	Cr Models	accuracy_scor	example_count	f1_score	false_negatives	false_positives	precision_scor	recall_score
	wise-mole-318	shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
	invincible-lark-929	shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
	sedate-fawn-631	shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
	awesome-moth-44	shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618
	fun-grub-597	shap, 1 mo	0.621	335	0.623	65	62	0.629	0.618

Searching runs

```
mlflow.search_runs()
```

[1] https://pandas.pydata.org/

Output format

```
#   Column                  Non-Null Count  Dtype
0   run_id                  6 non-null      object
1   experiment_id           6 non-null      object
2   status                  6 non-null      object
3   artifact_uri            6 non-null      object
4   start_time              6 non-null      datetime64[ns, UTC]
5   end_time                5 non-null      datetime64[ns, UTC]
6   metrics.test            1 non-null      float64
7   metrics.metric_2        3 non-null      float64
8   metrics.metric_1        3 non-null      float64
9   params.param_1          3 non-null      object
10  params.random_state     3 non-null      object
11  params.n_estimators     3 non-null      object
12  tags.mlflow.user        6 non-null      object
13  tags.mlflow.runName     6 non-null      object
14  tags.mlflow.source.type 6 non-null      object
15  tags.mlflow.source.name 6 non-null      object
```

Filtering run searches

- `max_results` - maximum number of results to return.
- `order_by` - column(s) to sort in `ASC` ending or `DESC` ending order.
- `filter_string` - string based query.
- `experiment_names` - name(s) of experiments to query.

Tracking UI

Insurance Experiment

Experiment ID: 27 Artifact Location: ./mlruns/27

Description Edit

metrics.rmse < 1 and params.model = "tree" Sort: accuracy_score Columns

Time created: All time State: Active S

				Metrics				
	Run Name	Created	Duration	accuracy_score	f1_score	false_negative	false_positives	precision_score
	wise-mole-318	3 months ago	10.4s	0.621	0.623	65	62	0.629
	powerful-shoat-853	3 months ago	5.0s	0.621	0.623	65	62	0.629
	amazing-penguin-22	3 months ago	4.6s	0.537	0.485	97	58	0.557
	traveling-snipe-808	3 months ago	5.6s	0.537	0.485	97	58	0.557

Search runs example

```
import mlflow
# Filter string
f1_score_filter = "metrics.f1_score > 0.60"
# Search runs
mlflow.search_runs(experiment_names=["Insurance Experiment"],
    filter_string=f1_score_filter,
    order_by=["metrics.precision_score DESC"])
```

Example output

```
# Search runs from Insurance Experiment
mlflow.search_runs(experiment_names=["Insurance Experiment"],
    filter_string=f1_score_filter,
    order_by=["metrics.precision_score DESC"])
```

```
                            run_id experiment_id  ... tags.mlflow.source.type tags.mlflow.user
0  90407e29a5aa4a31954bed874c7d4337            27  ...                   LOCAL             user
1  c335c0b16a5d4cf398aaa7189362b577            27  ...                   LOCAL             user
```

Let's practice!

INTRODUCTION TO MLFLOW