Deep Learning with Databricks

Srijith Rajamohan, Ph.D.
John O’Dwyer

databricks

30+ Million

monthly downloads

APACHE Spark

DELTA LAKE

mlflow

Koalas

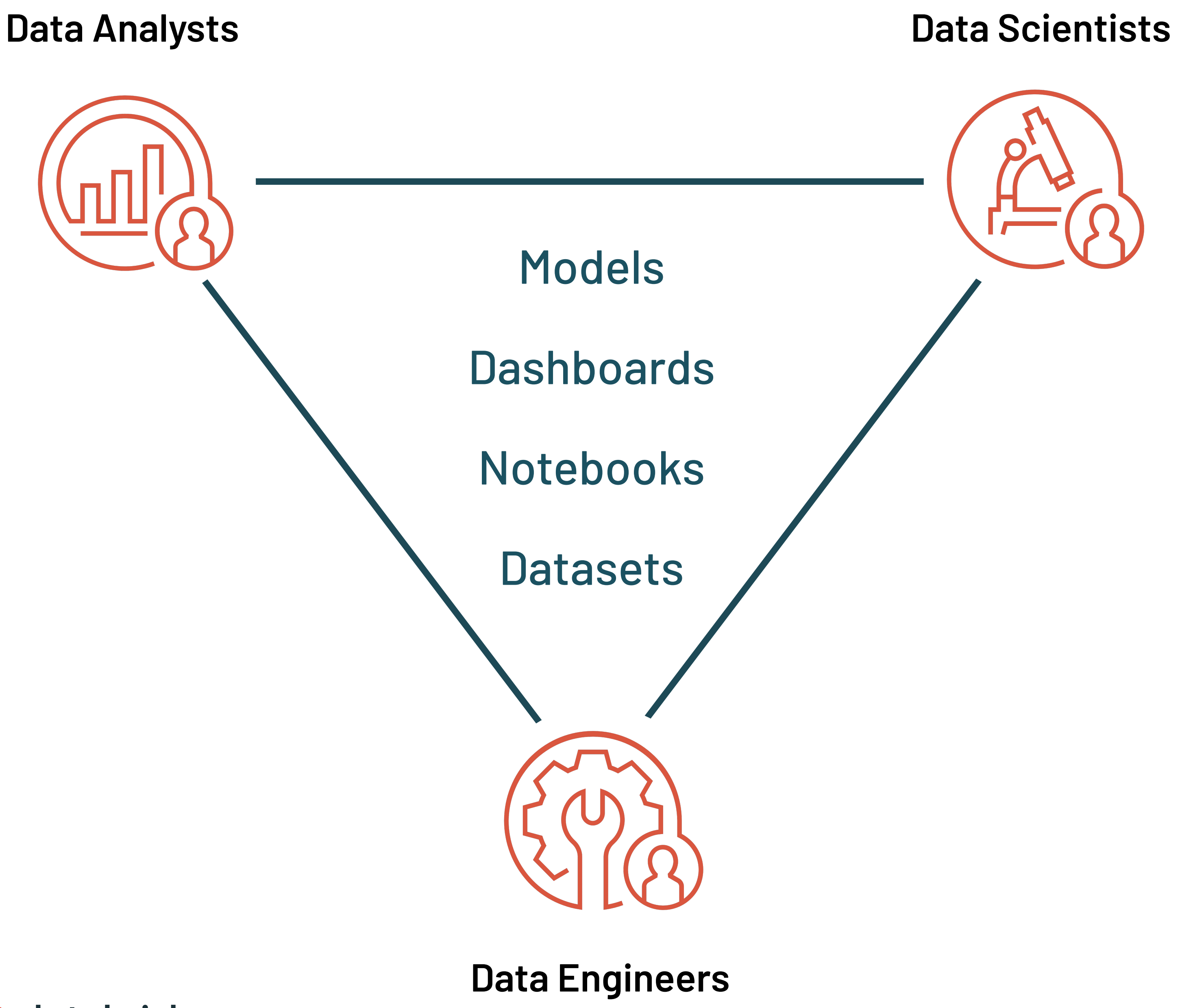
redash

databricks

Open

Unify your data ecosystem with open source, standards and formats

Built on the innovation of some of the most successful open source data projects in the world



Collaborative

Unify your data teams to collaborate across the entire data and AI workflow

Questions for Scalable ML

- Track the provenance and reason for model creation
- What training data was used, if any?
  - Proprietary data, sensitive data, storage, data retention period?
  - Real-time or batch?
- How are the models being used and who is using it?
  - Exploratory analysis and production environment?
- Is model performance being measured regularly and is the model being updated?
- Is the model well documented to ensure reuse?
- Is the model deployment process being automated?
- Institutional adoption and support

Best Practices for ML

- Software engineering practices
  - Code quality best practices
- Validate your data
  - Ensure proper data types and format are fed to your model (Schema validation)
  - Ensure no data drift, can render a supervised model ineffective
- Version and track your experiments like code!
  - Changing hyperparameters, inputs, code etc.
- Monitor predictive performance over time
  - Ensure model performance does not degrade over time
  - Ensure model fairness across different classes of data (bias)

What is MLOps?

MLOps = ML + DataOps + DevOps

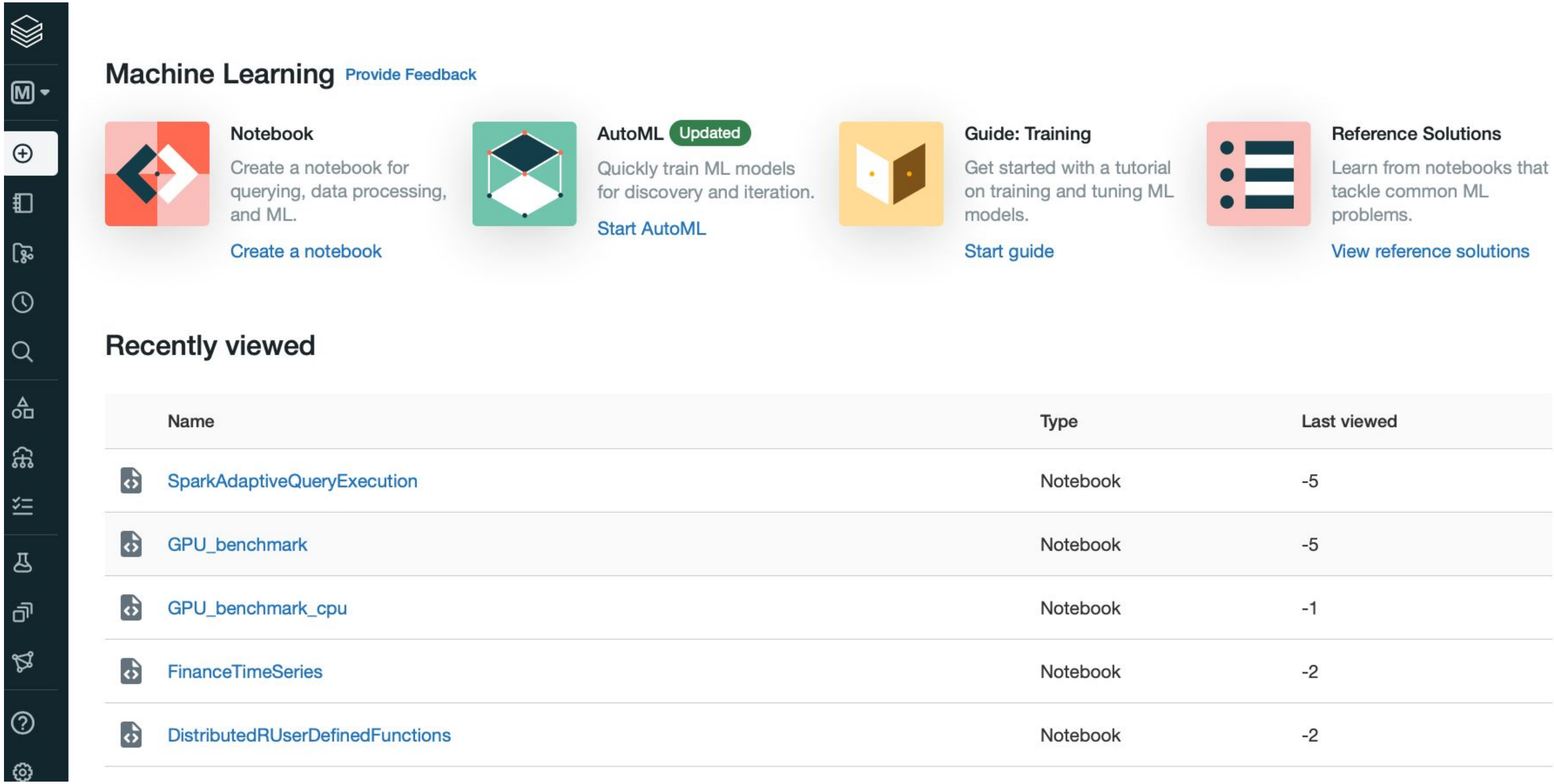
Build -> Test -> Deploy -> Monitor -> Feedback -> Build

Model management

Databricks Ecosystem for ML/DL

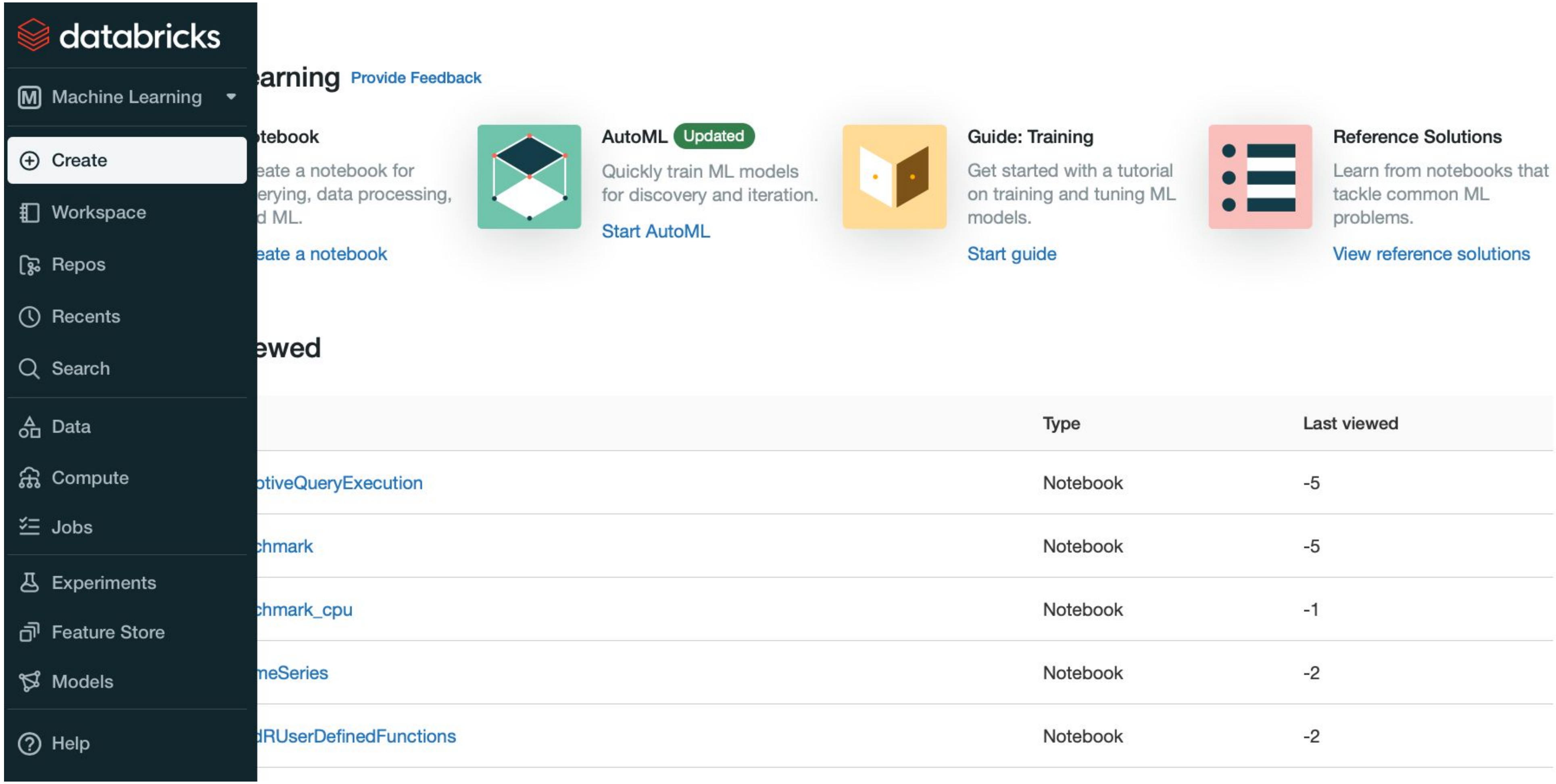
- Integrated Environment
    - Use compute instances from AWS, Azure or GCP
        - Centered around a notebook environment
            - Version control them with GitHub
        - Integrated 'DBFS' filesystem that can mount cloud filesystems like S3
        - Mix SQL, Python, R and Bash in the same notebook
        - Schedule jobs to run anytime
- Databricks Runtimes (DBRs)
    - Preinstalled with packages for ML/DL
    - Additional packages can be installed per cluster or per notebook
- MLflow integrated into the Databricks platform
    - Model tracking for experiment management/reproducibility
    - MLflow projects for packaging an experiment
    - Model serving with MLflow

Workspace



Machine Learning
Provide Feedback
Notebook
Create a notebook for querying, data processing, and ML.
Create a notebook
AutoML
Updated
Quickly train ML models for discovery and iteration.
Start AutoML
Guide: Training
Get started with a tutorial on training and tuning ML models.
Start guide
Reference Solutions
Learn from notebooks that tackle common ML problems.
View reference solutions
Recently viewed
Name
Type
Last viewed
SparkAdaptiveQueryExecution
Notebook
-5
GPU_benchmark
Notebook
-5
GPU_benchmark_cpu
Notebook
-1
FinanceTimeSeries
Notebook
-2
DistributedRUserDefinedFunctions
Notebook
-2

Workspace


databricks
Machine Learning
Create
Workspace
Repos
Recents
Search
Data
Compute
Jobs
Experiments
Feature Store
Models
Help
earning Provide Feedback
tebook
eate a notebook for
erying, data processing,
d ML.
eate a notebook
AutoML Updated
Quickly train ML models
for discovery and iteration.
Start AutoML
Guide: Training
Get started with a tutorial
on training and tuning ML
models.
Start guide
Reference Solutions
Learn from notebooks that
tackle common ML
problems.
View reference solutions
ewed
Type
Last viewed
otiveQueryExecution
Notebook
-5
chmark
Notebook
-5
chmark_cpu
Notebook
-1
meSeries
Notebook
-2
dRUserDefinedFunctions
Notebook
-2

Notebooks

GPU_benchmark Python

Schedule

test_g4dn

Cmd 12

Using the Huggingface Transformer in Pytorch

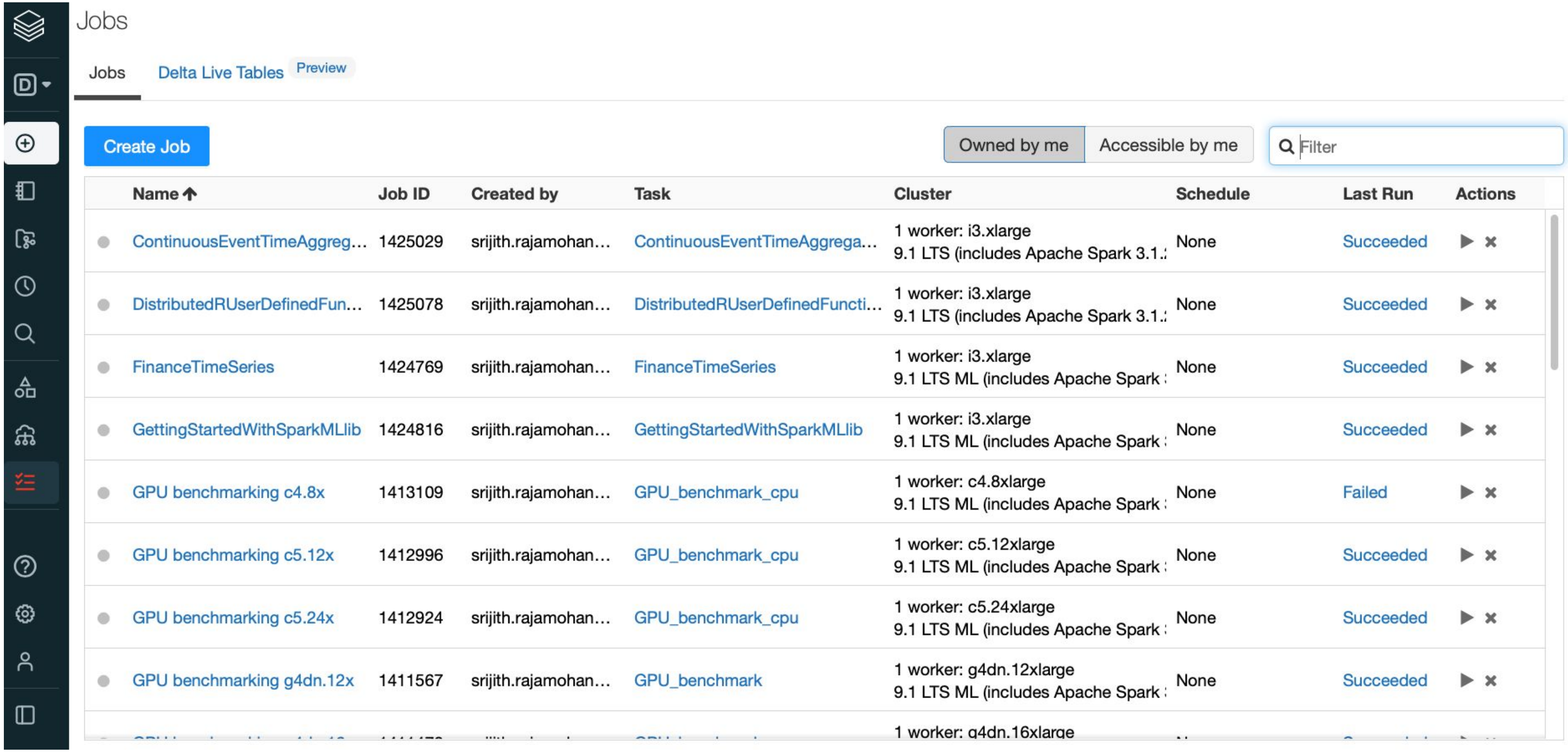
Cmd 13

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
from torch import nn

model_name = "distilbert-base-uncased-finetuned-sst-2-english"

pt_model = AutoModelForSequenceClassification.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)
pt_batch = tokenizer(
["We are very happy to show you the 🤗 Transformers library.", "We hope you don't hate it."],
    padding=True,
    truncation=True,
    max_length=512,
    return_tensors="pt")

pt_outputs = pt_model(**pt_batch)
pt_predictions = nn.functional.softmax(pt_outputs.logits, dim=1)
pt_predictions
```

```
Downloading:   0%|          | 0.00/629 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/255M [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/48.0 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/226k [00:00<?, ?B/s]
Out[8]: tensor([[2.2043e-04, 9.9978e-01],
        [5.3086e-01, 4.6914e-01]], grad_fn=<SoftmaxBackward>)
```

Command took 18.57 seconds -- by srijith.rajamohan@databricks.com at 10/21/2021, 3:31:37 PM on test_g4dn

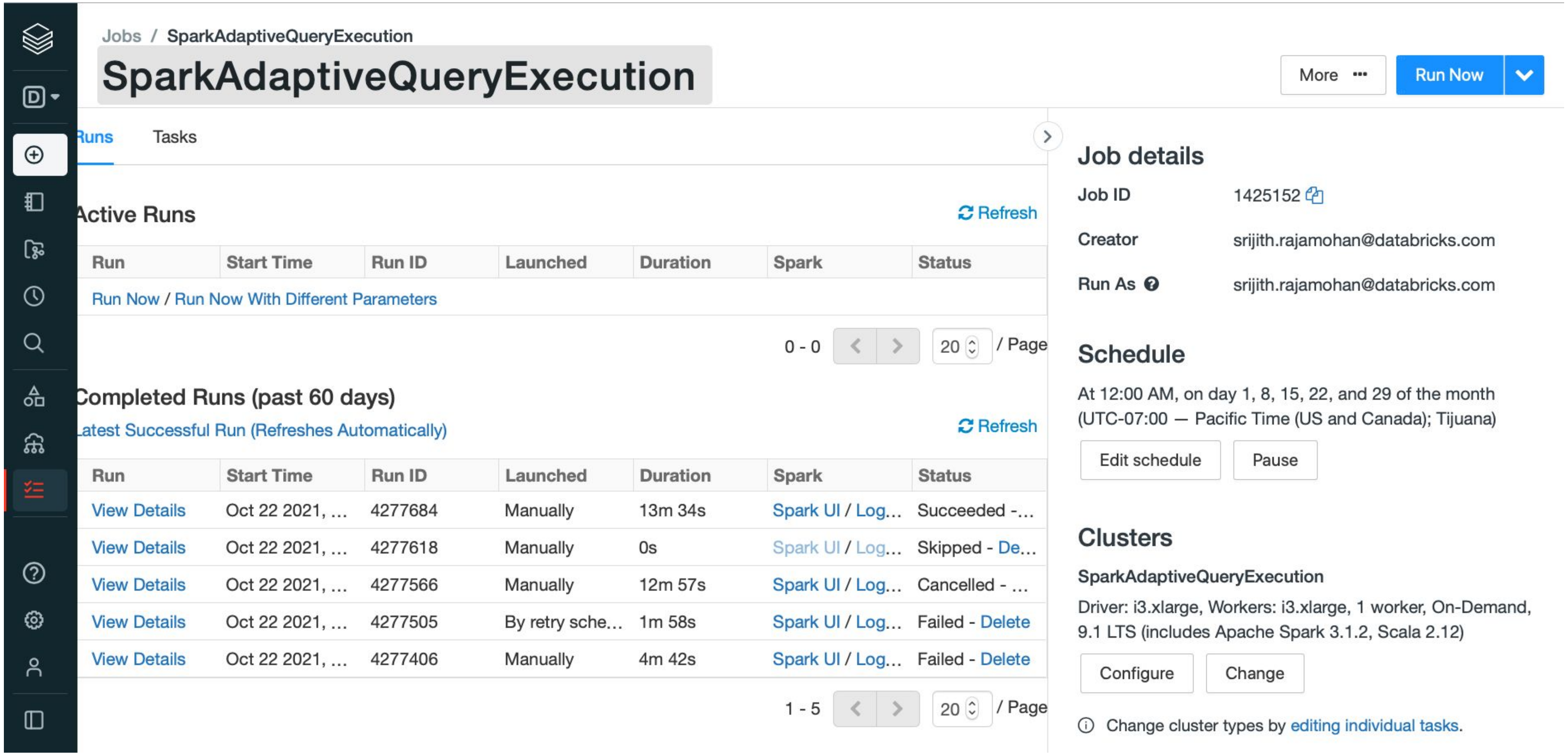
Job scheduling



	Name	Job ID	Created by	Task	Cluster	Schedule	Last Run	Actions
	ContinuousEventTimeAggreg…	1425029	srijith.rajamohan…	ContinuousEventTimeAggrega…	1 worker: i3.xlarge 9.1 LTS (includes Apache Spark 3.1.:	None	Succeeded	
	DistributedRUserDefinedFun…	1425078	srijith.rajamohan…	DistributedRUserDefinedFuncti…	1 worker: i3.xlarge 9.1 LTS (includes Apache Spark 3.1.:	None	Succeeded	
	FinanceTimeSeries	1424769	srijith.rajamohan…	FinanceTimeSeries	1 worker: i3.xlarge 9.1 LTS ML (includes Apache Spark :	None	Succeeded	
	GettingStartedWithSparkMLlib	1424816	srijith.rajamohan…	GettingStartedWithSparkMLlib	1 worker: i3.xlarge 9.1 LTS ML (includes Apache Spark :	None	Succeeded	
	GPU benchmarking c4.8x	1413109	srijith.rajamohan…	GPU_benchmark_cpu	1 worker: c4.8xlarge 9.1 LTS ML (includes Apache Spark :	None	Failed	
	GPU benchmarking c5.12x	1412996	srijith.rajamohan…	GPU_benchmark_cpu	1 worker: c5.12xlarge 9.1 LTS ML (includes Apache Spark :	None	Succeeded	
	GPU benchmarking c5.24x	1412924	srijith.rajamohan…	GPU_benchmark_cpu	1 worker: c5.24xlarge 9.1 LTS ML (includes Apache Spark :	None	Succeeded	
	GPU benchmarking g4dn.12x	1411567	srijith.rajamohan…	GPU_benchmark	1 worker: g4dn.12xlarge 9.1 LTS ML (includes Apache Spark :	None	Succeeded	
					1 worker: g4dn.16xlarge			

databricks

Job page


Jobs / SparkAdaptiveQueryExecution
SparkAdaptiveQueryExecution
More
Run Now
Runs
Tasks
Active Runs
Refresh
Run
Start Time
Run ID
Launched
Duration
Spark
Status
Run Now / Run Now With Different Parameters
0 - 0
20
/ Page
Completed Runs (past 60 days)
Latest Successful Run (Refreshes Automatically)
Refresh
Run
Start Time
Run ID
Launched
Duration
Spark
Status
View Details
Oct 22 2021, ...
4277684
Manually
13m 34s
Spark UI / Log...
Succeeded -...
View Details
Oct 22 2021, ...
4277618
Manually
0s
Spark UI / Log...
Skipped - De...
View Details
Oct 22 2021, ...
4277566
Manually
12m 57s
Spark UI / Log...
Cancelled - ...
View Details
Oct 22 2021, ...
4277505
By retry sche...
1m 58s
Spark UI / Log...
Failed - Delete
View Details
Oct 22 2021, ...
4277406
Manually
4m 42s
Spark UI / Log...
Failed - Delete
1 - 5
20
/ Page
Job details
Job ID
1425152
Creator
srijith.rajamohan@databricks.com
Run As
srijith.rajamohan@databricks.com
Schedule
At 12:00 AM, on day 1, 8, 15, 22, and 29 of the month
(UTC-07:00 — Pacific Time (US and Canada); Tijuana)
Edit schedule
Pause
Clusters
SparkAdaptiveQueryExecution
Driver: i3.xlarge, Workers: i3.xlarge, 1 worker, On-Demand,
9.1 LTS (includes Apache Spark 3.1.2, Scala 2.12)
Configure
Change
Change cluster types by editing individual tasks.

Experiments

Experiments

Preview Provide Feedback

Create AutoML Experiment | Owned by me | Accessible by me | Search experiments | Search

Name	Location	Last Modified	Created by	Notes
Test2	/Users/srijith.rajamohan@databricks.com/Test2	2021-10-22 12:07:08 EDT	srijith.rajamohan@data…	
flight_school_assignment_… /	/Shared/flight-school-apj_group2/flight-school/flight…	2021-10-21 03:03:19 EDT	junichi.maruyama@dat…	
flight_school_assignment_… /	/Shared/flight-school-team-4/flight-school-day-1/fli…	2021-10-20 18:00:10 EDT	alex.lopes@databricks.…	
ProdModeling /	/Repos/sean.owen@databricks.com/dais-2021-chur…	2021-10-20 17:21:54 EDT	sean.owen@databricks.…	
existing_run /	/Users/andre.mesarovic@databricks.com/work/exist…	2021-10-20 15:59:11 EDT	andre.mesarovic@data…	
_model_wrapper /	/Users/marijse.vandenberg@databricks.com/Experi…	2021-10-20 06:03:46 EDT	marijse.vandenberg@d…	
Churn_demographic_service-20…	/Users/sean.owen@databricks.com/databricks_auto…	2021-10-19 23:56:49 EDT	sean.owen@databricks.…	
Churn_demographic-2021_10_1…	/Users/sean.owen@databricks.com/databricks_auto…	2021-10-19 23:25:35 EDT	sean.owen@databricks.…	
Setup /	/Repos/sean.owen@databricks.com/dais-2021-chur…	2021-10-19 22:06:01 EDT	sean.owen@databricks.…	
AutomatedTest /	/Repos/sean.owen@databricks.com/dais-2021-chur…	2021-10-19 22:05:37 EDT	sean.owen@databricks.…	

Registered models

Registered Models

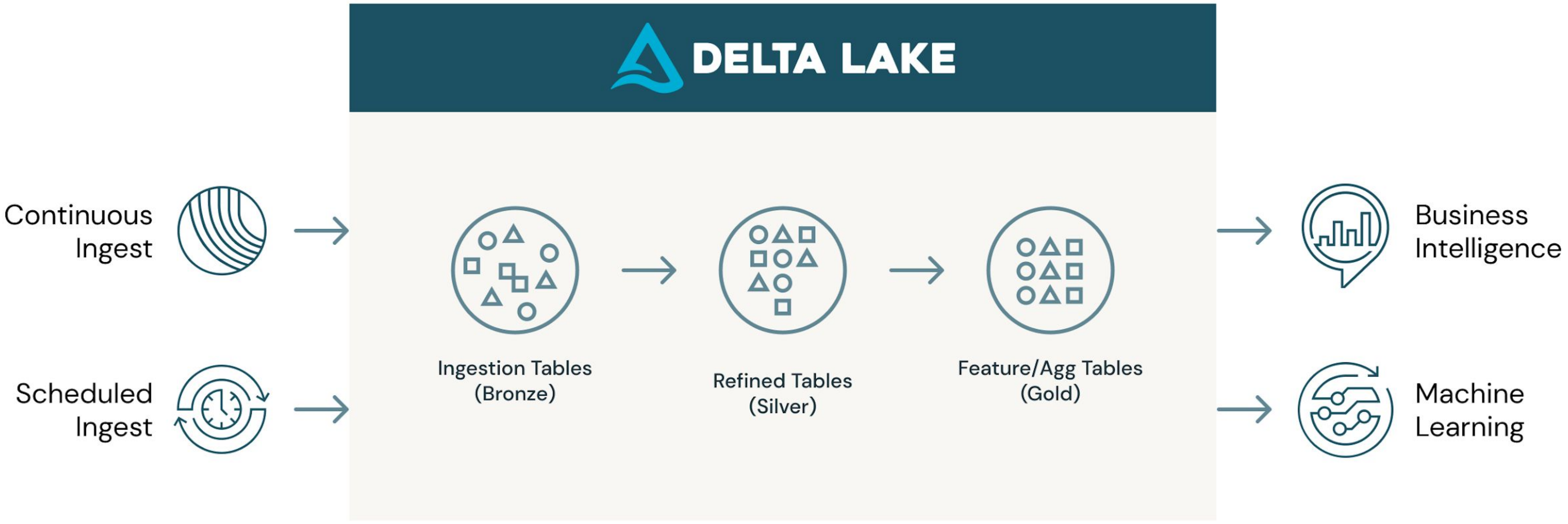
Share and serve machine learning models. Learn more

Create Model | Search by model name | Search | Filter | Clear

Name	Latest Version	Staging	Production	Last Modified	Tags	Serving
1_sri_bad_loan_model	Version 13	Version 11	Version 8	2021-10-19 16:32:11	–	–
andre_02_Sklearn_Train_Predict_Imported_05	Version 1	–	–	2021-08-10 14:38:07	–	–
gartner_2020	Version 33	Version 31	Version 30	2021-01-12 03:41:24	–	–
LynchWine	Version 37	–	Version 33	2021-09-14 10:40:44	–	–
nih_xray	Version 5	–	Version 5	2021-08-16 15:20:37	domain:hls	–
power-forecasting-model	Version 26	–	Version 25	2021-07-29 06:22:11	–	–
RP DistilBERT Classifier	Version 1	–	Version 1	2021-07-28 15:58:46	–	–
spark-model	Version 15	Version 15	Version 11	2021-07-08 16:55:54	–	–
turbine_failure_model	Version 10	Version 8	Version 10	2021-10-14 02:47:14	–	–

The Data Preparation

The Delta Lake Architecture



Data Store and Versioning

Delta Lake

- Scalable metadata
- Time travel
- Open format
- Unified Batch and Streaming
- Schema enforcement

Feature Store

- Data stored needs to be transformed into features to be useful
- Feature tables are Delta tables
- Feature Stores can save these features
  - Discoverable and reusable across an organization
  - Ensures consistency for Data Engineers, Data Scientists and ML Engineers
- Track feature lineage in a model

ETL and EDA

- Delta lake
    - Save data in scalable file formats like Parquet
    - Delta file formats can let you version control your data
- ETL
    - Read data
        - PySpark - Ideal for large data
        - Tensorflow (tf.data) and Pytorch (DataLoader)
    - Clean and process data
        - PySpark/Pandas API on Spark can work with large datasets across clusters
        - Clean and prepare the data
        - Extract features and save them using Feature Stores
- EDA
    - Preliminary data analysis such as inspecting records, summary statistics
    - Visualize the data and its distribution

The Model Build

Model training

- DBRs provide your favorite DL frameworks such as Tensorflow, Pytorch, Keras etc.
- Integration with MLflow for model tracking
- Hyperparameter tuning with Hyperopt/Optuna
- Seamlessly run single node but multi-CPU/multi-GPU jobs
- Distributed training on multiple nodes with Horovod
  - NVlink/NCCL enabled instances available for accelerating DL workloads
  - Tightly coupled - Train directly on Spark Dataframes with Horovod Estimator
  - Train on distributed Spark clusters with Horovod Runner

Distributed Training with Spark/Horovod

```python
def train_hvd(checkpoint_path, learning_rate=1.0):
  # Initialize Horovod
  hvd.init()
  # Call the get_dataset function you created, this time with the Horovod rank and size
  (x_train, y_train), (x_test, y_test) = get_dataset(num_classes, hvd.rank(), hvd.size())
  model = get_model(num_classes)
  # Adjust learning rate based on number of GPUs
  optimizer = keras.optimizers.Adadelta(lr=learning_rate * hvd.size())
  # Use the Horovod Distributed Optimizer
  optimizer = hvd.DistributedOptimizer(optimizer)
  model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
  model.fit(x_train, y_train,
            batch_size=batch_size,
            callbacks=callbacks,
            epochs=epochs,
            verbose=2,
            validation_data=(x_test, y_test))
```

Distributed Training with Spark/Horovod contd...

Invoke training across multiple nodes

```
checkpoint_path = checkpoint_dir + '/checkpoint-{epoch}.ckpt'
# Distribute training across 2 nodes
hr = HorovodRunner(np=2)
hr.run(train_hvd, checkpoint_path=checkpoint_path, learning_rate=0.1)
```

Inference using Horovod

```
hvd_model = get_model(num_classes)
hvd_model.compile(optimizer=tf.keras.optimizers.Adadelta(lr=0.1),
                loss='categorical_crossentropy',
                metrics=['accuracy'])
loss, accuracy = hvd_model.evaluate(x_test, y_test, batch_size=128)
```

Distributed Training

Data parallelism

- Data is divided among the different nodes
    - Entire model is copied to all the nodes
- Gradients are communicated back to all other nodes to update the model
    - Synchronous or asynchronous updates
- Model size is a concern

Model parallelism

- Model is divided among all the nodes
- Only works if you can take advantage of task parallelism in the model
- Model size is less of a concern

Deep Learning Synchronization

Model parameter server

- Central servers hold all shared parameters
- Workers receive updates from the central server
- Harder to scale
  - Speedup now depends on the overhead of communication with the central server

All-reduce

- All the machines store the shared parameters
- No central server
- Several architectures for this
  - Ring All-reduce
  - Tree All-reduce

Other Topics in Training

- Quantization-aware training
  - Lower-precision training to minimize memory/compute requirements
- Federated learning
  - Decentralized learning with the Federated Averaging algorithm (Google)
  - Keep data on device
  - Model is updated with data on device and updates sent back to central server
  - Updates from all devices are averaged
- Privacy-preserving learning
  - Learn from data that is encrypted or with minimal exposure to the data

Model tracking with MLflow

- The MLflow Tracking API
  - Integrations with common ML/DL tools such as Scikit-learn, Pytorch, Tensorflow, Spark etc.
- Logs metrics and artifacts (output files)
  - Can log this locally or a remote tracking server
- Tracking UI to query runs and visualize the results of a run
- Save and load models from a run

Model tracking with MLflow - Keras

```
with mlflow.start_run():
            # log parameters
            mlflow.log_param("hidden_layers", args.hidden_layers)
            mlflow.log_param("output", args.output)
            mlflow.log_param("epochs", args.epochs)
            mlflow.log_param("loss_function", args.loss)
            # log metrics
            mlflow.log_metric("binary_loss", ktrain_cls.get_binary_loss(history))
            mlflow.log_metric("binary_acc",  ktrain_cls.get_binary_acc(history))
            mlflow.log_metric("validation_loss", ktrain_cls.get_binary_loss(history))
            mlflow.log_metric("validation_acc", ktrain_cls.get_validation_acc(history))
            mlflow.log_metric("average_loss", results[0])
            mlflow.log_metric("average_acc", results[1])

            # log artifacts (matplotlib images for loss/accuracy)
            mlflow.log_artifacts(image_dir)
          #log model
            mlflow.keras.log_model(keras_model, model_dir)
```

Model tracking with MLflow - Autolog

With many of the popular libraries, you can use the autologging feature

```python
# enable autologging
mlflow.sklearn.autolog()

# train a model
model = LinearRegression()
with mlflow.start_run() as run:
    model.fit(X, y)
```

AutoML

- Only ML algorithms for now
- Works with 9.1 LTS ML DBRs and above
- Classification and Regression
  - Decision trees, Random Forests, Logistic Regression, XGBoost, LightGBM
- Forecasting with Prophet
- Run from the UI or use the command line API

AutoML

```python
import databricks.automl
data_dir = "dbfs:/tmp/ensemble_automl/"
dbutils.fs.rm(data_dir, True)
automl_models = databricks.automl.classify(train_df,
                                           target_col = "churn",
                                           data_dir= data_dir,
                                           timeout_minutes=60,
                                           max_trials=1000)
```

Metrics for the best trial:

	Validation	Train
f1_score	0.799	0.809
score	0.805	0.818
recall_score	0.805	0.818
accuracy_score	0.805	0.818
log_loss	0.407	0.392
precision_score	0.797	0.809
roc_auc_score	0.852	0.865

AutoML contd...

```
1   automl_models.experiment
```

```
Out[46]: <Experiment: artifact_location='dbfs:/databricks/mlflow-tracking/1467187977257554', experiment_id='1467187977257554', lifecycle_stage='activ
e', name='/Users/srijith.rajamohan@databricks.com/databricks_automl/21-10-14-16:29-automl_ensemble-13783057/automl_ensemble-Experiment-13783057', tag
s={'_databricks_automl': 'True',
 '_databricks_automl.best_trial_notebook_id': '1467187977257758',
 '_databricks_automl.data_dir': 'dbfs:/tmp/ensemble_automl/',
 '_databricks_automl.evaluation_metric': 'val_f1_score',
 '_databricks_automl.evaluation_metric_order_by_asc': 'False',
 '_databricks_automl.exploration_notebook_id': '1467187977257555',
 '_databricks_automl.max_trials': '1000',
 '_databricks_automl.problem_type': 'classification',
 '_databricks_automl.run_id': 'b9993667-17b2-49f5-8230-b61a095c0313',
 '_databricks_automl.start_time': '1634228966',
 '_databricks_automl.state': 'RUNNING',
 '_databricks_automl.target_col': 'churn',
 '_databricks_automl.timeout_minutes': '60',
 'mlflow.experimentType': 'MLFLOW_EXPERIMENT',
 'mlflow.ownerEmail': 'srijith.rajamohan@databricks.com',
 'mlflow.ownerId': '3655034657934253'}>

Command took 0.02 seconds -- by srijith.rajamohan@databricks.com at 10/14/2021, 1:47:05 PM on Test
```

AutoML - Load the best model

```python
print(automl_models.best_trial.model_description)
best_model_uri = automl_models.best_trial.model_path
metrics = automl_models.best_trial.metrics
print('accuracy=', metrics['val_accuracy_score'], ' f1 score=', metrics['val_f1_score'], ' precision=', metrics['val_precision_score'],  \
                ' recall=',metrics['val_recall_score'],  ' roc_auc_score=',metrics['val_roc_auc_score'])
predict_udf = mlflow.pyfunc.spark_udf(spark, model_uri=best_model_uri, result_type="integer")
test_df = test_df.withColumn("bestModel", predict_udf())
display(test_df)
```

▸ (1) Spark Jobs

```
XGBClassifier(base_score=None, booster=None, colsample_bylevel=None,
              colsample_bynode=None, colsample_bytree=None, gamma=None,
              gpu_id=None, importance_type='gain', interaction_constraints=None,
              learning_ra...
accuracy= 0.805170239596469  f1 score= 0.799325453841428  precision= 0.7968969091728362  recall= 0.805170239596469  roc_auc_score= 0.8524996656137788
```

databricks

AutoML - Experiments

Showing 100 matching runs

Refresh | Compare | Delete | Download CSV | ↓ val_f1_score | All

Columns | metrics.rmse < 1 and params.model = "tree" | Search | Filter | Clear

							Metrics <			
	Start Time	Run Name	User	Source	Version	Models	training_accura	training_f1_scc	training_log_lo	tra
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.818	0.809	0.392	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.828	0.82	0.372	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.829	0.821	0.387	0.8
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.996	0.996	0.074	0.9
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.808	0.794	0.428	0.7
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.829	0.82	0.376	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.819	0.81	0.394	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.824	0.816	0.384	0.8
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.82	0.812	0.39	0.8
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.819	0.808	0.395	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.811	0.802	0.402	0.8
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.793	0.774	0.434	0.7
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.846	0.84	0.352	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.869	0.864	0.309	0.8
	12 days ago	xgboost	srijith.rajam...	Notebook:	-	sklearn	0.819	0.809	0.547	0.8
	12 days ago	lightgbm	srijith.rajam...	Notebook:	-	sklearn	0.99	0.99	0.024	0.9

The Model Inference and Deployment

Model Inference - Pandas UDF

- Use a compiled DL model with Pandas UDF for distributed inference
- Scalar pandas UDF (batch of data) vs. Iterator pandas UDF (iterator of batches) here so model is no initialized for every batch

```
@pandas_udf(ArrayType(FloatType()), PandasUDFType.SCALAR_ITER)
def predict_batch_udf(image_batch_iter):
  batch_size = 64
  model = ResNet50(weights=None)
  model.set_weights(bc_model_weights.value)
  for image_batch in image_batch_iter:
    images = np.vstack(image_batch)
    dataset = tf.data.Dataset.from_tensor_slices(images)
    dataset = dataset.map(parse_image, num_parallel_calls=8).prefetch(5000).batch(batch_size)
    preds = model.predict(dataset)
    yield pd.Series(list(preds))

predictions_df = df.select(predict_batch_udf(col("data")).alias("prediction"))
```

Model Packaging with MLflow Projects

MLProject file for reproducible executions

File under folder sklearn_elasticnet_wine

Execute this project using the command below

```
name: tutorial

conda_env: conda.yaml

entry_points:
  main:
    parameters:
      alpha: {type: float, default: 0.5}
      l1_ratio: {type: float, default: 0.1}
    command: "python train.py {alpha} {l1_ratio}"
```

```
mlflow run sklearn_elasticnet_wine -P alpha=0.42
```

Model Serve with MLflow

Serve the model

```
mlflow models serve -m
/Users/mlflow/mlflow-prototype/mlruns/0/7c1a0d5c42844dcdb8f5191146925
174/artifacts/model -p 1234
```

Send a request

```
curl -X POST -H "Content-Type:application/json; format=pandas-split"
--data '{"columns":["alcohol", "chlorides", "citric acid",
],"data":[[12.8, 0.029, 0.48]]}' http://127.0.0.1:1234/invocations
```

Thank you!