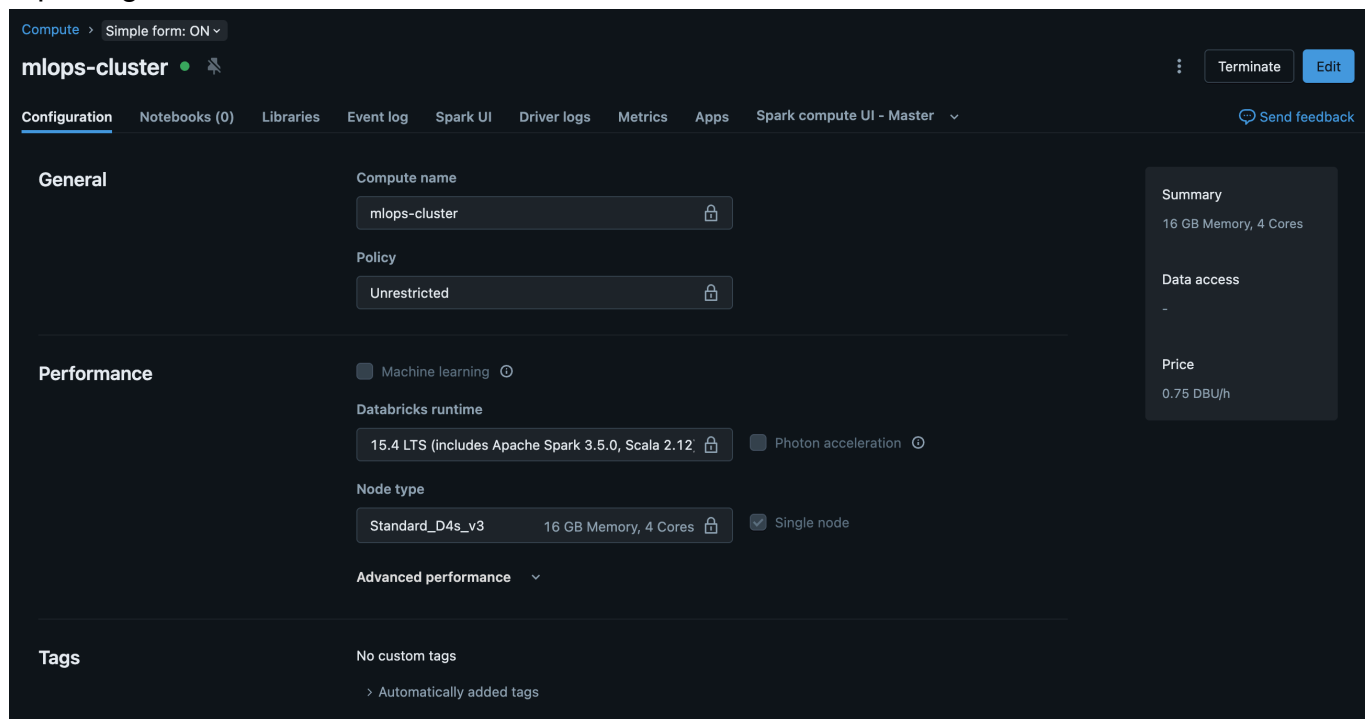


Databricks. Integration and Workflow.

This report documents the execution of the P300 classification pipeline on the Databricks platform. The workflow demonstrates data ingestion, preprocessing, model training, a champion-challenger deployment strategy, and automated model evaluation.

Initial setup

The foundation of the workflow is a Databricks All-Purpose Compute cluster. This managed environment provides the necessary computational resources for all notebook-based tasks, replacing the local CPU and Docker environment.

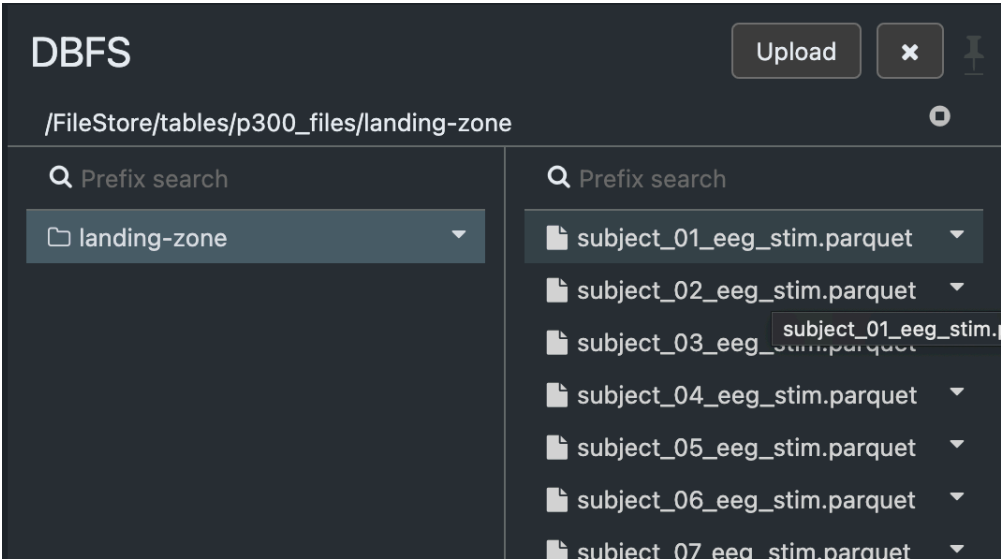


The screenshot displays the Databricks console interface for configuring a cluster named 'mlops-cluster'. The interface is divided into several sections:

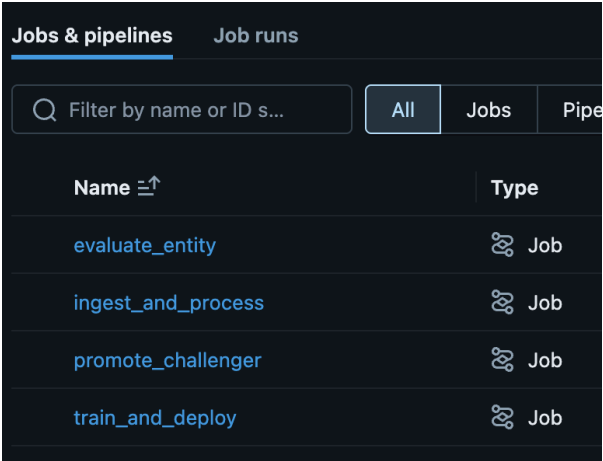
- General:** Includes fields for 'Compute name' (mlops-cluster) and 'Policy' (Unrestricted).
- Performance:** Includes a 'Machine learning' toggle, 'Databricks runtime' (15.4 LTS), 'Node type' (Standard_D4s_v3, 16 GB Memory, 4 Cores), and a 'Single node' checkbox.
- Tags:** Shows 'No custom tags' and a link to 'Automatically added tags'.
- Summary:** A sidebar on the right provides a summary of the cluster configuration, including '16 GB Memory, 4 Cores', 'Data access', and 'Price' (0.75 DBU/h).

Data is managed using the Databricks File System (DBFS). The workflow begins with raw subject data uploaded to a landing-zone directory within DBFS. It is worth noting that due to permissions on the university-provided Azure account, I was unable to use the modern Unity Catalog Volumes. As a result, this project uses the legacy DBFS for file management, which

functions similarly to the MinIO object storage used in the local Docker setup.



The entire pipeline is orchestrated using Databricks Workflows (Jobs). Separate jobs are configured for data preparation, model training/deployment, evaluation, and promotion. These jobs execute a series of parameterized notebooks, passing configuration like data paths and model names, which makes the pipeline modular and configurable without code changes.



Example of job configuration with attached compute and job parameters. They are like environmental variables for notebook-based tasks.

Compute

mlops-cluster

Single node: Standard_D4s_v3 · Release: 15.4.18

View details Swap Spark UI Logs

Metrics

< > **Job parameters** ⓘ

batch_size	5
landing_zone_path	/dbfs/FileStore/tables/p300_files/landing-zone/
processed_features_path	/dbfs/FileStore/tables/p300_files/processed-features/
raw_data_path	/dbfs/FileStore/tables/p300_files/raw-data/

Edit parameters

We can also schedule job run with the corresponding section allowing continuous pipeline execution:

Schedule

None

Add trigger

1. Data ingestion and preprocessing

The `ingest_and_process` job is executed first. This job consists of two sequential tasks:

1. `ingest_batch_from_landing_zone` moves a batch of raw data files from the landing zone to the raw-data directory.
2. `preprocess_batch_from_raw_data` scans the raw-data directory, identifies new subjects not present in processed-features, and triggers a separate notebook run for each new subject to perform filtering, epoching, and feature extraction.

Jobs & Pipelines > ingest_and_process >

ingest_and_process run Lakeflow UI: OFF

Graph Timeline **List** Filter task names starting with... Status Type

Status	Name	Type	Duration	Depends on
✓ Succeeded	ingest_batch_from_landing_zone	Notebook	13s	
✓ Succeeded	preprocess_batch_from_raw_data	Notebook	2m 18s	ingest_batch_from_landir

Upon completion, the DBFS directories are populated, with the final features ready for model training in processed-features.

DBFS Upload ×

/FileStore/tables/p300_files/raw-data

Prefix search

- landing-zone
- processed-features
- raw-data**

Prefix search

- subject_01_eeg_stim.parquet
- subject_02_eeg_stim.parquet
- subject_03_eeg_stim.parquet
- subject_04_eeg_stim.parquet
- subject_05_eeg_stim.parquet

DBFS Upload ×

/FileStore/tables/p300_files/processed-features/01

Prefix search

- 01**
- 02
- 03
- 04
- 05

Prefix search

- features.parquet
- labels.parquet

2. Training and deploying the first model

The `train_and_deploy` job orchestrates model creation and initial deployment. It consists of two tasks: `train_classifier` and `deploy_as_challenger`.

The training notebook retrains the model from scratch using all available processed data. For the current dataset size and the choice of LinearDiscriminantAnalysis (which does not support

incremental learning), this approach is computationally feasible.

Jobs & Pipelines > train_and_deploy >

train_and_deploy run Lakeflow UI: OFF

GraphTimelineList

Filter task names starting with...

StatusType

Status	Name	Type	Duration	Depends on
Succeeded	train_classifier	Notebook	41s	
Succeeded	deploy_as_challenger	Notebook	20s	train_classifier

The run logs parameters, metrics, and registers the packaged model and scaler to the MLflow Model Registry. A key feature is that the training run also logs a list of subject names it trained on as an artifact (training_subjects.json). This ensures that evaluation further on can be performed on a completely unseen holdout set, preventing data leakage.

Experiments > /Users/kalapun.pn@ucu.edu.ua/P300_BCI_Training > Runs >

LDA_Training_Run Send feedback

Reproduce RunRegister model

OverviewModel metricsSystem metricsTracesEvaluation resultsArtifacts

Metrics (3)

Search metrics

Metric	Latest	Min	Max	Models
validation_accuracy	0.867098865...	0.867098865...	0.867098865...	p300-pyfunc...
validation_kappa	0.44299365...	0.44299365...	0.44299365...	p300-pyfunc...
validation_auc	0.827073827...	0.827073827...	0.827073827...	p300-pyfunc...

Parameters (8)

Search parameters

Parameter	Value
internal_val_split_ratio	0.8
lda_shrinkage	auto
lda_solver	lsqr
model_type	LDA
num_subjects_trained_on	5
training_set_size	2464
training_subject_names	01, 02, 03, 04, 05
validation_set_size	617

Logged models (1)

Model attributes

No dataset

Type	Step	Model name	Status	Created	Dataset	validation_accuracy	validation_auc
Output	0	P300-Classifer	Ready	3 minutes ago	-	0.86709886547812	0.8270738270738272

The deploy_as_challenger task then programmatically creates the p300 Model Serving Endpoint if it does not exist, or updates it if it does. On the first run, it seeds the endpoint by assigning the newly trained model (Version 1) to both the champion and challenger slots. Traffic is configured to send 100% of requests to the champion entity by default.

Serving endpoints >

p300

Ready <https://adb-4428371568956498.18.azuredatabricks.net/serving-endpoints/p300/invocations>

Gateway Configure AI Gateway

Rate limits: Not configured

Active configuration

Entity	Version	Name	State	Compute	Traffic (%)
P300-Classifier	Version 1	champion	Ready	CPU, Small 0-4 concurrency (0-4 DBU)	100
P300-Classifier	Version 1	challenger	Ready	CPU, Small 0-4 concurrency (0-4 DBU)	0

3. Evaluating the champion model

After ingesting more data (+5 subjects), the `evaluate_entity` job is run. By default, its `target_entity_name` parameter is set to "champion".

le tasks using the hover task controls

make_inference_with_servin...

....pn@ucu.edu.ua/05_run_simulation

mlops-cluster

Override job parameters for this run. Any parameters that are not overridden here will have their default values. [Learn More](#)

Key	Value
num_subjects_to_test	5 (default) ✎
processed_data_path	/dbfs/FileStore/tables/p300_files/proc... ✎
registered_model_name	P300-Classifier (default) ✎
serving_endpoint_name	p300 (default) ✎
target_entity_name	champion (default) ✎

The simulation notebook first queries the serving endpoint's API to confirm which model version the champion is serving. It then fetches that version's `training_subjects.json` artifact to create a holdout set of unseen subjects.

The final metrics are logged to a new MLflow run.

Experiments > /Users/kalapun.pn@ucu.edu.ua/P300_BCI_Inference > Runs >

evaluation_champion_v1 [Send feedback](#)

Overview Model metrics System metrics Traces Evaluation results [?](#) Artifacts

Experiment ID	3056886130461916 🔗
Status	🟢 Finished
Run ID	ae0d3479897f4793b27222b1aad1f603 🔗
Duration	1.0s
Datasets used	—
Tags	Add tags
Source	🔗 run 461102575480478 of job 546861661145923
Logged models	—
Registered models	—

Metrics (2)

Metric	Latest	Min	Max
evaluation_accuracy	0.834645669291...	0.834645669291...	0.834645669291...
evaluation_kappa	0	0	0

Parameters (4)

Parameter	Value
evaluated_entity_alias	champion
evaluated_model_name	P300-Classifer
evaluated_model_version	1
num_test_subjects	5

Notes on evaluation

1. A key part of this evaluation is ensuring requests are sent to the correct model. While aliases are a Unity Catalog feature and not available in my workspace, I discovered through documentation that it is still possible to query a specific served entity directly.

The simulation notebook constructs a direct invocation URL like `/serving-endpoints/{endpoint-name}/served-models/{entity-name}/invocations`. By setting entity-name to "champion", it bypasses the main traffic split and guarantees that only the champion model is evaluated.

The notebook first queries the endpoint's management API to confirm which model version the champion entity is serving, then fetches that version's `training_subjects.json` artifact to create a holdout set of unseen subjects.

2. During the simulation, I encountered `SSLError` issues, which indicated the serving endpoint was crashing. This was resolved by implementing mini-batching in the request loop, sending a smaller number of records per API call to fit within the resource limits of the "Small" compute size of the serving endpoint.

4. Training and deploying a new challenger

The `train_and_deploy` job is run again.

Jobs & Pipelines > train_and_deploy >

train_and_deploy run Lakeflow UI: OFF

Graph	Timeline	List	Filter task names starting with...	Status	Type
Status	Name	Type	Duration	Depends on	
Succeeded	train_classifier	Notebook	34s		
Succeeded	deploy_as_challenger	Notebook	13s	train_classifier	

This creates a new model (Version 2), transitions it to the Staging stage in the legacy Model Registry, and the `deploy_as_challenger` task updates the serving endpoint to point its challenger entity to this new Version 2.

Versions All Active 1 Compare

Version	Registered at	Created by	Stage	Pending Requests	Description	Endpoints
Version 2	Jul 09, 2025, 12:30 AM	kalapun.pn@ucu.e...	Staging	-		p300
Version 1	Jul 08, 2025, 11:15 PM	kalapun.pn@ucu.e...	Archived	-		p300

Active configuration

Entity	Version	Name	State	Compute	Traffic (%)
P300-Classifer	Version 1	champion	Ready (Scaled to zero)	CPU, Small 0-4 concurrency (0-4 DBU)	100
P300-Classifer	Version 2	challenger	Ready	CPU, Small 0-4 concurrency (0-4 DBU)	0

5. Evaluating the new challenger and promotion

The `evaluate_entity` job is run again, but this time the `target_entity_name` parameter is explicitly set to `"challenger"`.

ers

ect multiple tasks using the hover task controls

make_inference_with_servin...
...pn@ucu.edu.ua/05_run_simulation
mlops-cluster

Override job parameters for this run. Any parameters that are not overridden here will have their default values. [Learn More](#)

Key	Value
num_subjects_to_test	5 (default) ✎
processed_data_path	/dbfs/FileStore/tables/p300_files/proc... ✎
registered_model_name	P300-Classifer (default) ✎
serving_endpoint_name	p300 (default) ✎
target_entity_name	<input type="text" value="challenger"/> {} ✕

The evaluation metrics show that Version 2 performs slightly better than the current champion.

Experiments > /Users/kalapun.pn@ucu.edu.ua/P300_BCI_Inference > Runs >

evaluation_challenger_v2 [Send feedback](#)

Overview

Model metrics

System metrics

Traces

Evaluation results ⓘ

Artifacts

Duration

0.7s

Datasets used

—

Tags

[Add tags](#)

Source

[05_run_simulation](#)

Logged models

—

Registered models

—

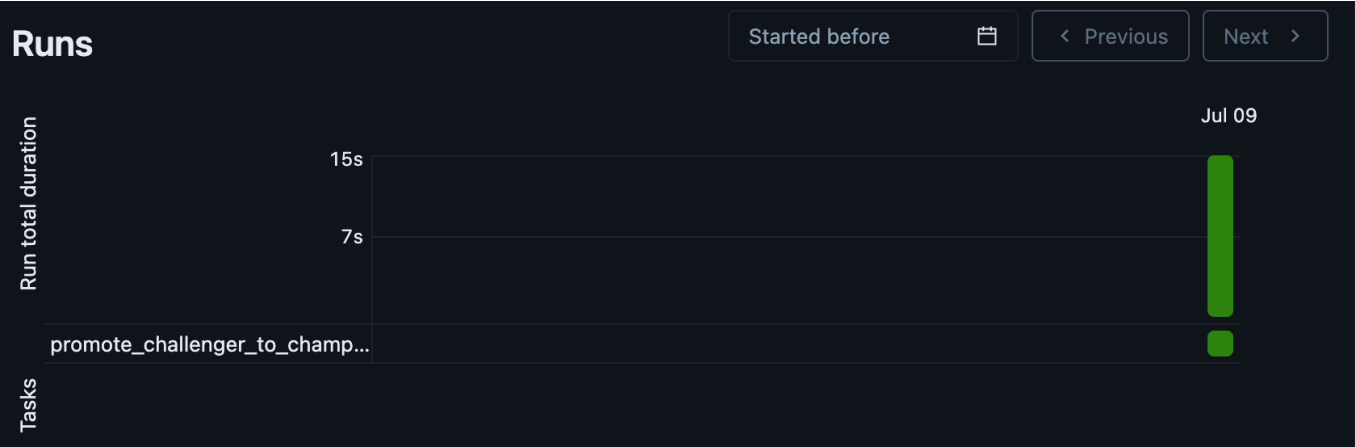
Metrics (2)

Metric	Latest	Min	Max
evaluation_kappa	0	0	0
evaluation_accuracy	0.84995425434...	0.84995425434...	0.84995425434...

Parameters (4)

Parameter	Value
evaluated_entity_alias	challenger
evaluated_model_name	P300_Classifier
evaluated_model_version	2
num_test_subjects	5

Based on these results, we run the `promote_challenger` job.



This completes the automated promotion cycle, demonstrating a full champion-challenger workflow on the Databricks platform

Registered Models > P300-Classifier >

Version 2

Registered At: Jul 09, 2025, 12:30 AM

Stage: Production ▾

Entity	Version	Name	State	Compute	Traffic (%)
■ P300-Classifier	Version 2	champion	✓ Ready	CPU, Small 0-4 concurrency (0-4 DBU)	100
■ P300-Classifier	Version 2	challenger	✓ Ready (Scaled to zero)	CPU, Small 0-4 concurrency (0-4 DBU)	0