

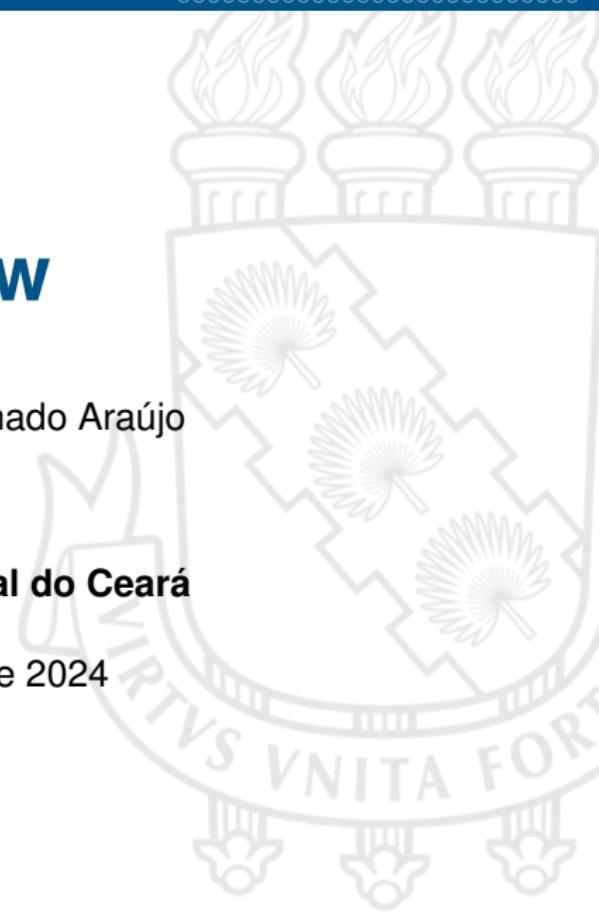
MLflow

Pedro Hericson Machado Araújo

MDCC

Universidade Federal do Ceará

5 de setembro de 2024



Summary

1 Motivation

2 MLflow



Motivation



Machine Learning Workflow

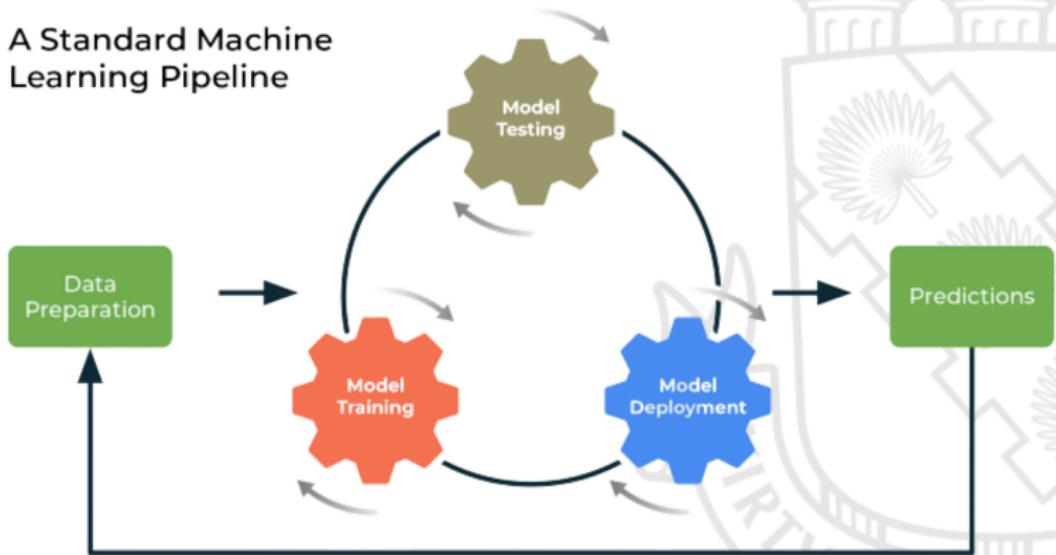


Figura: A Standard Machine Learning Workflow



accuracy mse
f1-score svm

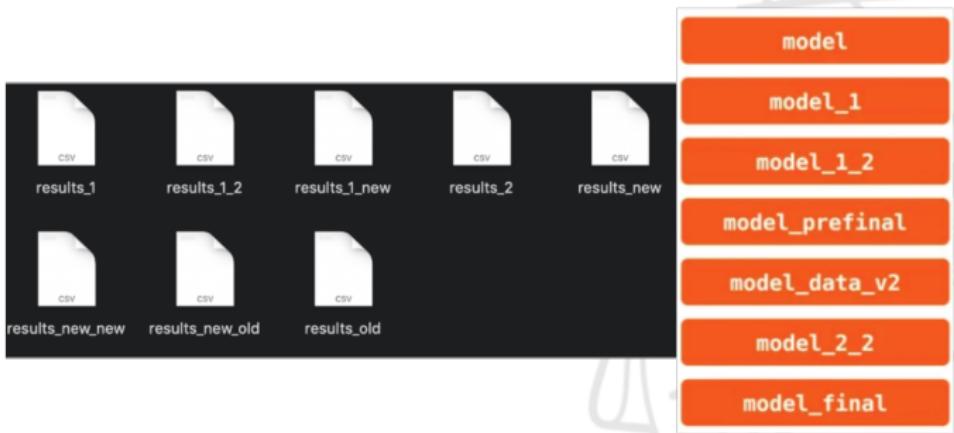
test random forest

r2 roc curve precision model
decision tree learning rate data auc mae
rmse ols train lssvm
neurons logistic regression hiperparameters
recall mlp validation

accuracy mse
f1-score svm

test random forest
r2 roc curve precision model
decision tree learning rate data auc mae
rmse logistic regression hiperparameters
neurons recall mlp validation





Machine Learning Workflow: Difficulties

- Experiments
 - Data
 - Code
 - Parameters
 - Models
- Absence of a central place to collaborate and manage model lifecycle

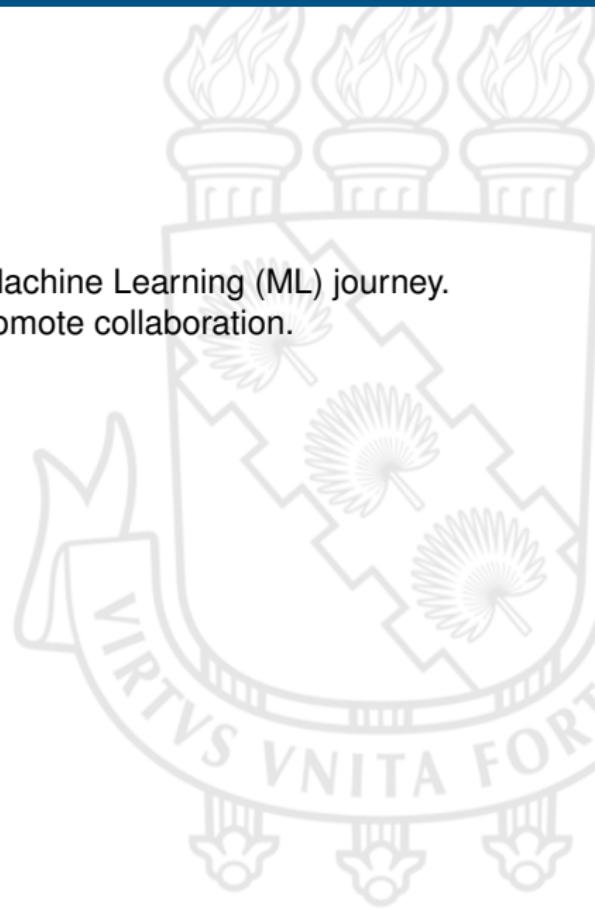
MLflow



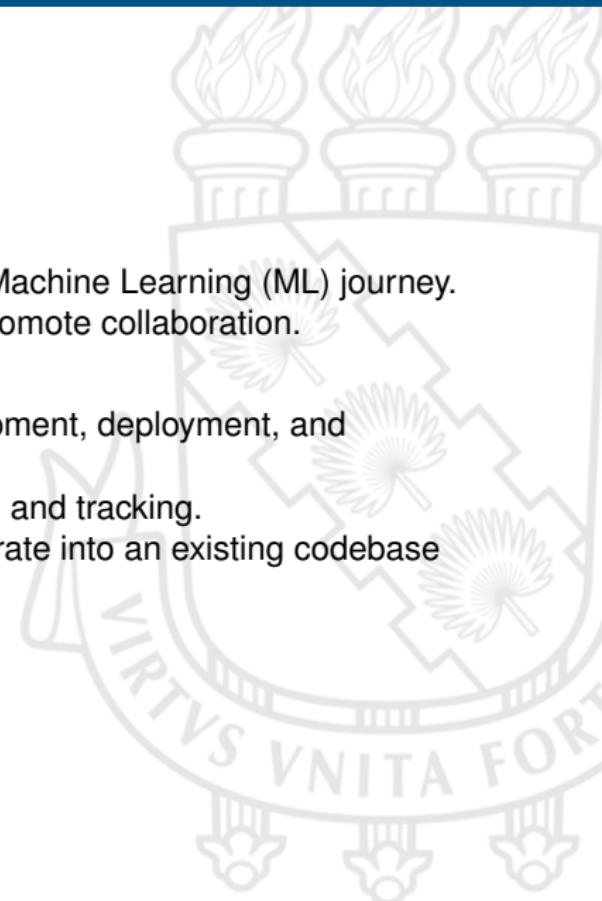
What is MLflow?

■ Introduction to MLflow:

- A solution for complexities in the Machine Learning (ML) journey.
 - Tools to simplify processes and promote collaboration.



What is MLflow?



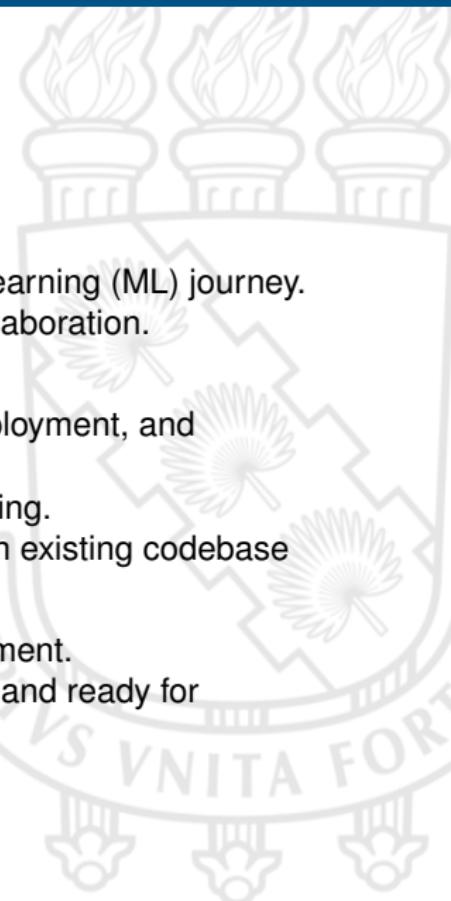
■ Introduction to MLflow:

- A solution for complexities in the Machine Learning (ML) journey.
 - Tools to simplify processes and promote collaboration.

■ Benefits of MLflow:

- Unified platform for model development, deployment, and management.
 - Streamlines logging, organization, and tracking.
 - Require minimal changes to integrate into an existing codebase

What is MLflow?



■ Introduction to MLflow:

- A solution for complexities in the Machine Learning (ML) journey.
 - Tools to simplify processes and promote collaboration.

■ Benefits of MLflow:

- Unified platform for model development, deployment, and management.
 - Streamlines logging, organization, and tracking.
 - Require minimal changes to integrate into an existing codebase

■ Goals of MLflow:

- Empower innovation in ML solution development.
 - Ensure ML projects are robust, transparent, and ready for real-world challenges.

What makes MLflow different?



- **Open Source**
 - Integrate with any ML library and platform
 - **Comprehensive**
 - Manage end-to-end ML and GenAI workflows, from development to production
 - **Unified**
 - Unified platform for both traditional ML and GenAI applications



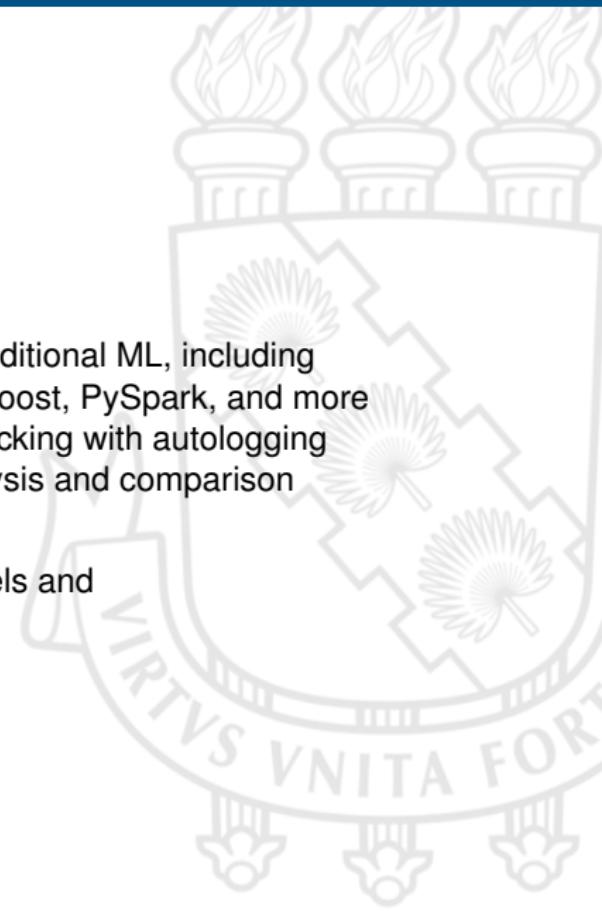
MLflow

■ Traditional ML

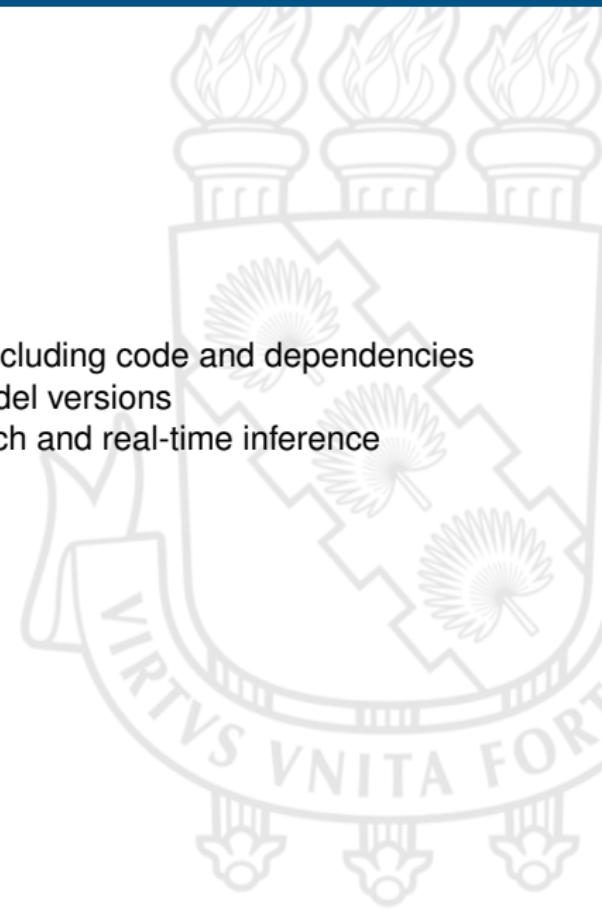
- End-to-end MLOps solution for traditional ML, including integrations with scikit-learn, XGBoost, PySpark, and more
 - Simple, low-code performance tracking with autologging
 - State-of-the-art UI for model analysis and comparison

■ Evaluation

- Easily compare different ML models and
 - Evaluate different experiments



MLflow



■ Model Management

- Package models for production, including code and dependencies
 - Catalog, govern, and manage model versions
 - Deploy models for large scale batch and real-time inference

■ Run MLflow anywhere

- Databricks
 - Your cloud Provider
 - Your Computer

MLflow

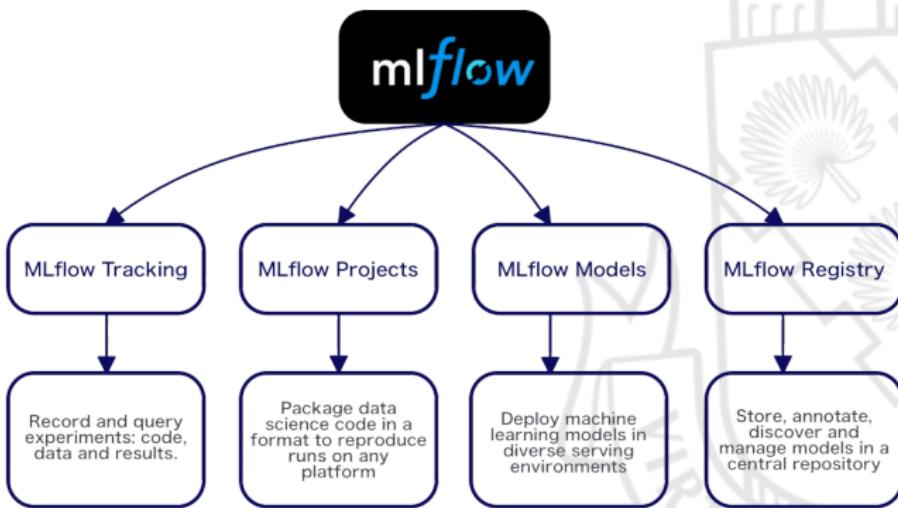
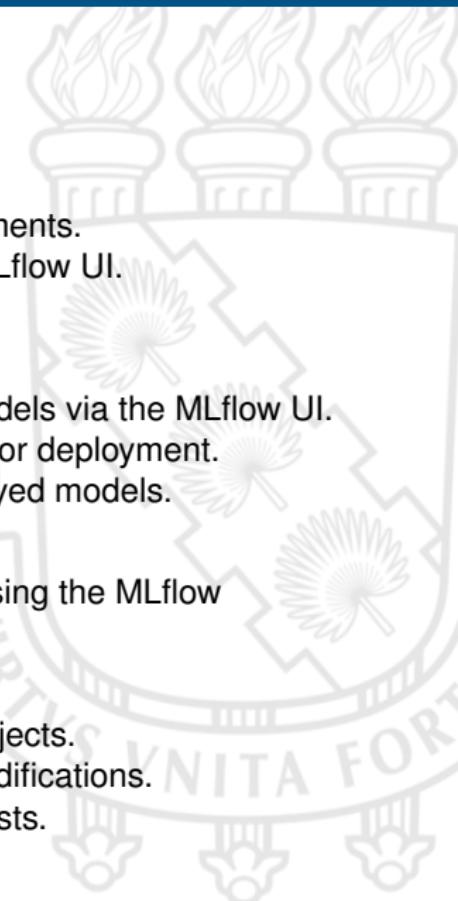


Figura: MLFlow Components



MLflow Use Cases



■ Experiment Tracking:

- Logging parameters and metrics for experiments.
- Comparing results approaches using the MLflow UI.
- Preserving outcomes as MLflow models.

■ Model Selection and Deployment:

- Assessing and selecting top-performing models via the MLflow UI.
- Registering models in the MLflow Registry for deployment.
- Monitoring real-world performance of deployed models.

■ Model Performance Monitoring:

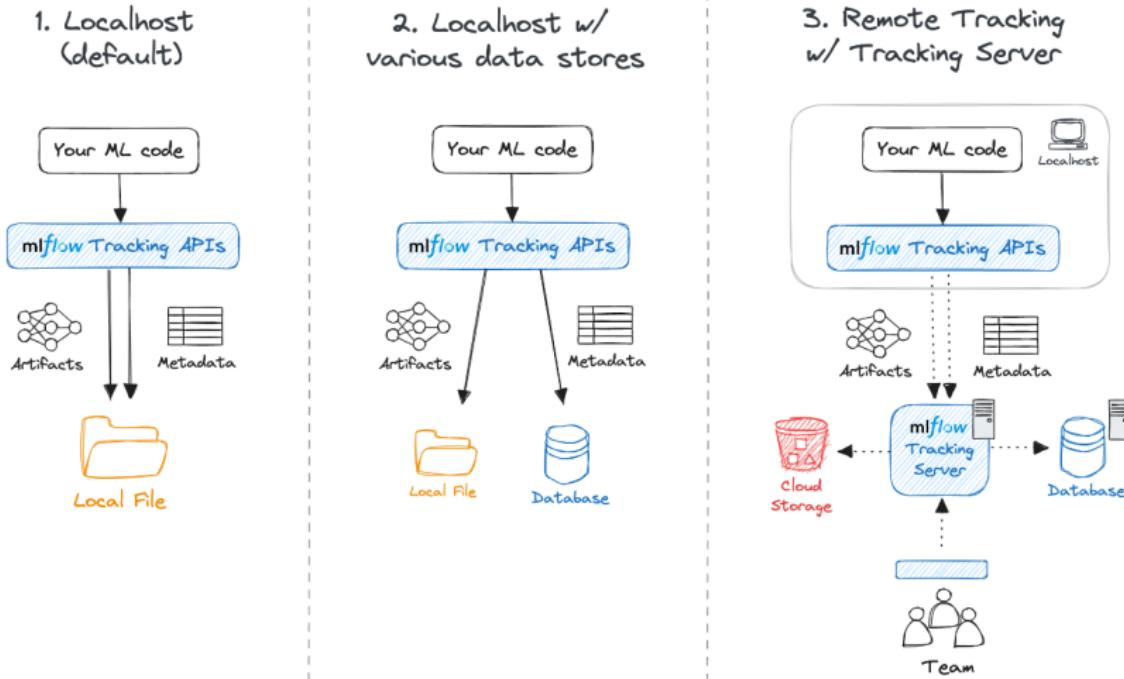
- Gauging model efficacy post-deployment using the MLflow Registry.

■ Collaborative Projects:

- Organizing new experiments as MLflow Projects.
- Facilitating easy sharing and parameter modifications.
- Promoting collaboration among data scientists.



Common Setup



UNIVERSIDADE
FEDERAL DO CEARÁ

Tracking Server

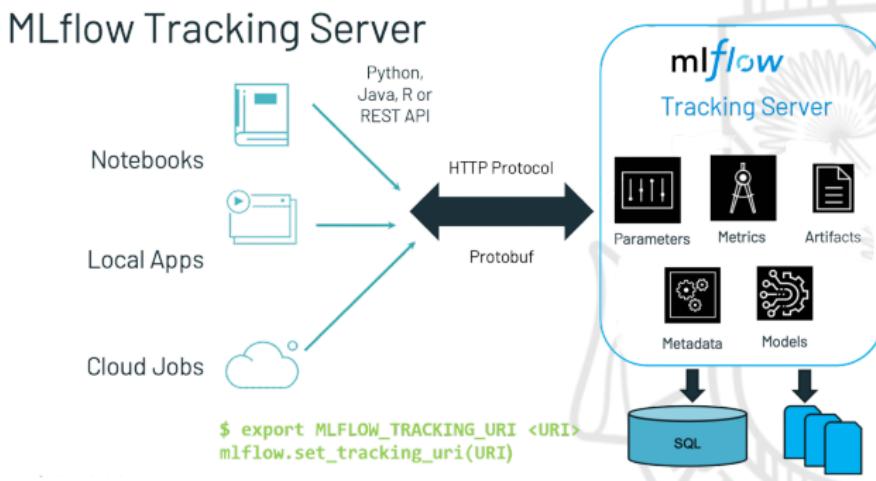


Figura: MLFlow Tracking



Tracking



UNIVERSIDADE
FEDERAL DO CEARÁ

Tracking

The experiment that we created

That contains our run

Run Name	Created	Dataset	Duration	Source	Models
kindly-store-713	4 minutes ago	-	3.7s	ipykernel	tracking-q-11



UNIVERSIDADE
FEDERAL DO CEARÁ



The screenshot shows the MLflow UI for a completed experiment run. The experiment name is 'kindly-stork-713'. The run ID is '4eaf57b14954d889dc72d8374c4ef08'. The status is 'FINISHED' and the date is '2023-11-07 11:28:03'. The source is 'databricks_launcher.py' and the user is 'Benjamin Wilson'. The experiment has 4 parameters: max_iter (1000), multi_class (auto), random_state (6666), and solver (lbfgs). It has 1 metric: accuracy (1). A tag 'tracking-pipeline.v1' was set. An artifact named 'iris_model' was registered, with a full path of 'FullPipelineArtifact@/d4057841509515044834eaf57b14954d889dc72d8374c4ef08/iris_model'. The MLflow Model section shows code snippets for making predictions using the logged model. Handwritten annotations include green arrows pointing to the parameters, metrics, and tag sections, and a red arrow pointing to the 'iris_model' artifact. A large green arrow points to the 'MLflow Model' section. A red box highlights the 'Our model signature' text.

MLflow Dokelist >
kindly-stork-713

Run ID: 4eaf57b14954d889dc72d8374c4ef08 Date: 2023-11-07 11:28:03 Status: FINISHED Source: databricks_launcher.py User: Benjamin Wilson

> Description Edit

> Datasets

> Parameters (4)

Name	Value
max_iter	1000
multi_class	auto
random_state	6666
solver	lbfgs

> Metrics (1)

Name	Value
accuracy	1

> Tags (1)

Name	Value
tracking-pipeline.v1	

> Training info

Name	Value
Basic LR model for Iris data	

> Actions

Add

> Artifacts

Name	Value
iris_model	FullPipelineArtifact@/d4057841509515044834eaf57b14954d889dc72d8374c4ef08/iris_model
constan	param
input_params.json	
model.pkl	
pyfunc.pkl	
requirements.txt	

Our model signature →

The parameters that we logged

The loss metric (accuracy) that we logged

Our tag that we set for future reference

Our model and its metadata

Our registration link

Our model signature

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. This model is also registered to the `model` registry.

Model schema

Input and output schema for your model. [Learn more](#)

Name	Type
Inputs (1)	Tensor (shape: float64, shape: [-1,4])
Outputs (1)	Tensor (shape: int64, shape: [-1])

Make Predictions

Predict on a Pandas DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/e4fa7b14954d889dc72d8374c4ef08/iris_model'

# Load model as a Spark UDF. Override result_type if the model does not return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Pandas DataFrame.
df = df[columns].predict(loaded_model).to_pandas()
```

Predict on a Pandas DataFrame:

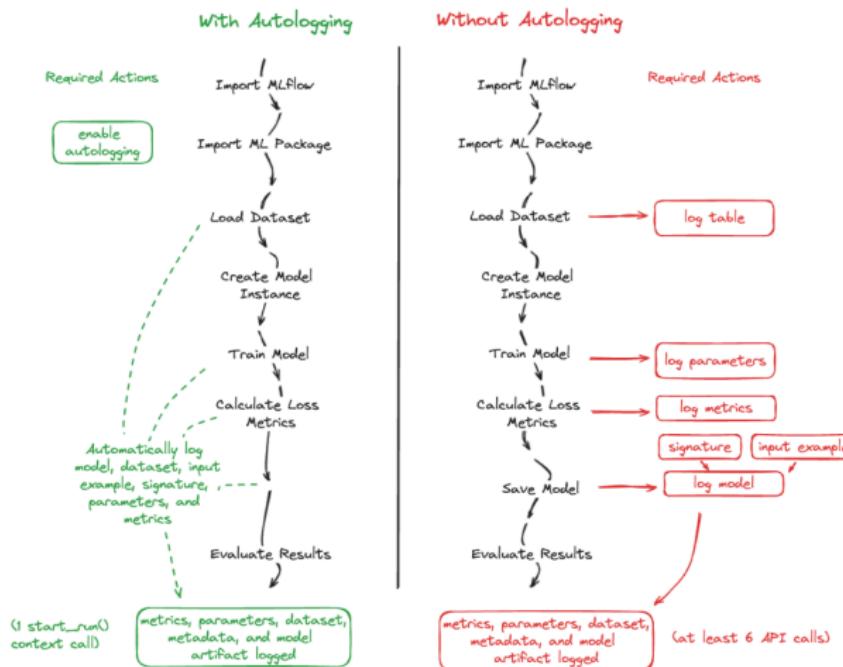
```
import mlflow
logged_model = 'runs:/e4fa7b14954d889dc72d8374c4ef08/iris_model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)
```



UNIVERSIDADE
FEDERAL DO CEARÁ

Autologging





The screenshot shows the mlflow UI interface. At the top, there are tabs for 'Experiments' and 'Models'. The 'Experiments' tab is selected, showing a sidebar with 'Search Experiments' and a list of runs: 'Default', 'Apples Demand', and 'Span Classifier Training'. Below this is a search bar with the query 'metrics.rossme > 1 and params.model = "tree"'. The main area has tabs for 'Table', 'Chart', 'Evaluation', and 'Experimental'. The 'Experimental' tab is active, displaying several charts:

- Parallel Coordinates:** A chart comparing 95 runs across various parameters like 'gamma', 'lambda', 'max_depth', 'min_samples_leaf', and 'eta'. A red arrow points from the text 'parameters from runs' to this chart.
- Time vs. runs:** A scatter plot showing the time taken for each run. A red box highlights the word 'run'.
- Comparing first 20 runs:** A scatter plot showing the first 20 runs. A red box highlights the word 'run'.
- eta vs. rmse:** A scatter plot comparing eta and rmse for the first 101 runs. A red box highlights the word 'run'.
- lambda vs. rmse:** A scatter plot comparing lambda and rmse for the first 101 runs. A blue box highlights the word 'lambda'.

A green box highlights the text 'Runs that contain parameters and metrics' with a green arrow pointing to the sidebar.

Annotations in the bottom right corner include:

- A red arrow pointing to the 'eta vs. rmse' chart with the text 'parameters from runs'.
- A blue arrow pointing to the 'lambda vs. rmse' chart with the text 'parameters from runs'.



MLflow Project

An **MLflow Project** is simply a directory that contains all the files and information needed to run a machine learning experiment. Inside this directory, you can have:

- The model code or data pipeline.
- Configuration files (such as a `conda.yaml` or `requirements.txt`) that specify the project dependencies.
- A file called `MLproject`, which contains instructions on how the project should be executed.

Advantages

- Reproducibility



UNIVERSIDADE
FEDERAL DO CEARÁ

Models

■ Standard Format for ML Models:

- Packaging machine learning models for diverse downstream tools.

■ Applications:

- Real-time serving through a REST API.
- Batch inference on Apache Spark.

■ Model Flavors:

- Convention that enables saving models in different "flavors."
- Ensures compatibility with various downstream tools.

- CatBoost
- PyTorch
- Scikit-learn
- Fastai



Model Signature

Introduction

- **Definition:** Describes the expected input and output schema for a machine learning model.
- **Importance:** Ensures correct data format during model inference and reuse.

Key Components of a Model Signature

- **Input Schema:** Specifies the data types and shapes for the input features (e.g., int, float, string).
- **Output Schema:** Defines the expected format of the model's output (e.g., predicted values or probabilities).

Model Registry - Core Components

■ Centralized Model Store:

- A single location for MLflow models.
- Facilitates model versioning, sharing, and deployment.

■ Set of APIs:

- Programmatically create, read, update, and delete models.

■ GUI:

- Manually view and manage models in the centralized model store.



UNIVERSIDADE
FEDERAL DO CEARÁ

Model Registry - Additional Functionality:

■ Model Versioning:

- Log different iterations of a model for comparison and serving.
- Models are versioned with a monotonically increasing ID by default.
- Alias model versions for easy reference.

■ Model Aliasing:

- Assign mutable, named references to specific model versions.
- Simplifies model deployment.

■ Model Tagging:

- Label models with custom key-value pairs.
- Facilitates documentation and categorization.

■ Model Annotations:

- Add descriptive notes to a model.



UNIVERSIDADE
FEDERAL DO CEARÁ

Model Registry

■ Model

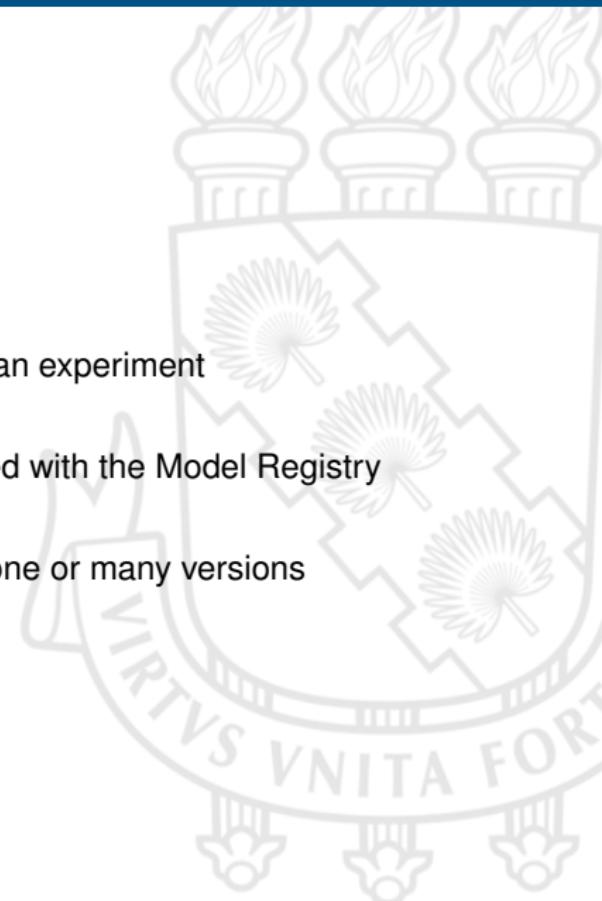
- An MLflow Model is created from an experiment

■ Registered Model

- An MLflow Model can be registered with the Model Registry

■ Model Version

- Each registered model can have one or many versions
 - Version: 1,2...



MLflow - Registered Models

mlflow 2.8.0 Experiments Models GitHub Docs

Default >

magnificent-seal-662

Run ID: f34c3372e3bf4f278cb3fd73cd0815ab Date: 2023-09-25 14:05:17 Source: python

User: jerry.liang Duration: 2.5s Status: FINISHED

Lifecycle Stage: active

> Description Edit

> Parameters (2)

> Metrics

> Tags

< Artifacts

sk_models

- MLmodel
- conda.yaml
- input_example.json
- model.pkl
- python_env.yaml
- requirements.txt

Full Path: file:///Users/jerry.liang/mlflow/mlruns/0/f34c3372e3bf4f278cb3fd73cd0815ab/artifacts/sk... Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also [register](#) it to the model registry to version control

Model schema

Input and output schema for your model. [Learn more](#)

Name	Type
------	------

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/f34c3372e3bf4f278cb3fd73cd0815ab/sk_models'
```



UNIVERSIDADE
FEDERAL DO CEARÁ

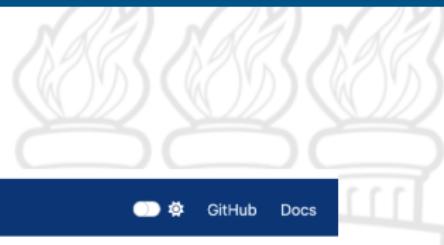
MLflow - Registered Models

The screenshot shows the mlflow UI interface. At the top, there's a navigation bar with 'mlflow 2.8.0' and links for 'Experiments' and 'Models'. On the right, there are 'GitHub' and 'Docs' links. Below the navigation, a breadcrumb path 'Default > magnificent-seal-662' is shown. The main content area displays a run summary for 'Run ID: f34c3372e3bf4f278cb3fd73cd0815ab' (Date: 2023-09-25 14:05:17, User: jerry.liang, Duration: 2.5s, Status: FINISHED). To the left, a sidebar lists lifecycle stages: 'active' (Description, Edit), 'Parameters (2)', Metrics, Tags, and Artifacts (sk_models, MLmodel, conda.yaml, input_example.json, model.pkl, python_env.yaml, requirements.txt). A modal window titled 'Register Model' is open, showing a dropdown for 'Model' ('Create New Model') and a text input for 'Model Name' ('iris_model_testing'). At the bottom of the modal are 'Cancel', 'Register' (disabled), and 'Register Model' buttons. Below the modal, a section titled 'MLflow Model' contains code snippets for making predictions using the registered model. A table at the bottom shows the 'Model schema' (Input and output schema) and 'Make Predictions' (Predict on a Spark DataFrame).



UNIVERSIDADE
FEDERAL DO CEARÁ

MLflow - Registered Models



mlflow 2.10.0 Experiments Models

GitHub Docs

Registered Models

Create Model

Filter registered models by name o...

Name	Latest version	Aliased versions	Created by	Last modified	Tags
iris_model_dev	Version 17			2023-09-25 12:50:...	-
iris_model_prod	Version 11	@ champion: Version 11 +3		2023-10-26 17:10:...	-
iris_model_staging	Version 11			2023-09-25 12:46:...	-
iris_model_testing	Version 1			2023-09-27 13:17:...	-
mnist_model_dev	Version 12			2023-09-25 12:39:...	-
mnist_model_prod	Version 8	@ challenger: Version 8 +1		2024-01-19 10:35:...	-
mnist_model_staging	Version 8			2023-09-25 12:51:...	-

New model registry UI

< Previous Next >

25 / page

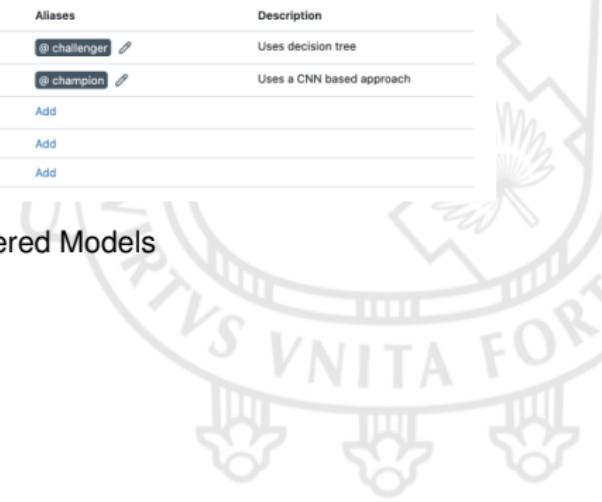
Figura: MLFLow Registered Models

MLflow - Registered Models



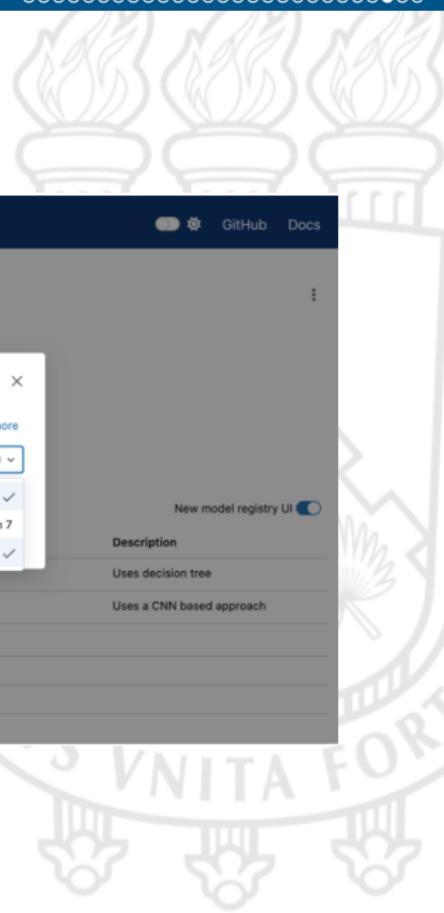
Description		Edit			
Tags					
Versions	Compare				
New model registry UI <input checked="" type="checkbox"/>					
Version	Registered at	Created by	Tags	Aliases	Description
Version 8	2023-09-25 12:38:44		validation_status: pending <input type="text"/>	@challenger <input type="text"/>	Uses decision tree
Version 7	2023-09-25 12:38:41		validation_status: approved <input type="text"/>	@champion <input type="text"/>	Uses a CNN based approach
Version 6	2023-09-25 12:38:38		validation_status: approved <input type="text"/>	Add	
Version 5	2023-09-25 12:38:35		Add	Add	
Version 4	2023-09-25 12:38:32		Add	Add	

Figura: MLFlow Registered Models



UNIVERSIDADE
FEDERAL DO CEARÁ

MLflow - Registered Models



mlflow 2.10.0 Experiments Models

Registered Models >
mnist_model_prod

Created Time: 2023-09-25 12:10:09 Last Modified: 2024-01-19 10:35:06

> Description Edit

> Tags

Versions Compare

Version	Registered at	Description
Version 8	2023-09-25 12:38:44	validation_status: pending challenger
Version 7	2023-09-25 12:38:41	validation_status: approved champion
Version 6	2023-09-25 12:38:38	validation_status: approved Add
Version 5	2023-09-25 12:38:35	Add
Version 4	2023-09-25 12:38:32	Add
Version 3	2023-09-25 12:38:30	Add

Add/Edit alias for model version 8

Aliases allow you to assign a mutable, named reference to a particular model version. [Learn more](#)

@ challenger ✎ @ new_alias ✎

challenger This version ✓

champion Version 7

new_alias ✓

New model registry UI

Example

Obrigado pela Atenção!

