

MACHINE LEARNING OPERATIONS



Basic_mlflow



Presented by Asst. Prof. Dr. Tuchsana Ploysuwan



Model Registry

บททวน MLflow Tracking

MLflow Tracking		
Experiment: "spam-detection"		
Run: run_001		
Parameters	Metrics	Artifacts
<ul style="list-style-type: none">lr=0.001epochs=50batch=32	<ul style="list-style-type: none">acc=0.92loss=0.23f1=0.89	<ul style="list-style-type: none">model.pklconfusion.pngrequirements.txt
Run: run_002		
Parameters	Metrics	Artifacts
<ul style="list-style-type: none">lr=0.01epochs=30batch=64	<ul style="list-style-type: none">acc=0.88loss=0.31f1=0.85	<ul style="list-style-type: none">model.pklconfusion.pngrequirements.txt
Run: run_003		
Parameters	Metrics	Artifacts
<ul style="list-style-type: none">lr=0.005epochs=100batch=16	<ul style="list-style-type: none">acc=0.95loss=0.15f1=0.94	<ul style="list-style-type: none">model.pklconfusion.pngrequirements.txt

Experiment: "spam-detection" - Summary Table				
PARAMETERS				
Run	Learning Rate	Epochs	Batch Size	Status
run_001	0.001	50	32	Baseline
run_002	0.01	30	64	Underfitting
run_003	0.005	100	16	★ Best Model
METRICS				
Run	Accuracy	Loss	F1-Score	Improvement
run_001	0.92	0.23	0.89	-
run_002	0.88	0.31	0.85	-4.35%
run_003	0.95	0.15	0.94	+3.26%
ARTIFACTS				
Run	Files Logged			
run_001	model.pkl, confusion.png, requirements.txt			
run_002	model.pkl, confusion.png, requirements.txt			
run_003	model.pkl, confusion.png, requirements.txt			
KEY INSIGHTS				
<div><div> Best Accuracy:</div><div>run_003 (0.95) ← +3.26% improvement over baseline</div></div> <div><div> Lowest Loss:</div><div>run_003 (0.15) ← 34.8% reduction from baseline</div></div> <div><div> Best F1-Score:</div><div>run_003 (0.94) ← +5.6% improvement over baseline</div></div> <div><div> Insight:</div><div>Smaller batch size (16) + more epochs (100) + moderate lr (0.005) yields the best performance for spam detection task</div></div> <div><div> Warning:</div><div>run_002 shows underfitting due to insufficient epochs (30) and high learning rate (0.01)</div></div>				

๓. ปัญหาที่เกิดขึ้นเมื่อไม่มี Model Registry

3.1 สถานการณ์จำลอง: โฟลเดอร์ที่วุ่นวาย

ลองนึกภาพโปรเจกต์ ML ที่ไม่มีระบบจัดการ:

```

📁 โปรเจกต์ ML (ไม่มี Model Registry)
├── model_v1.pkl
├── model_v2.pkl
├── model_v2_fixed.pkl
├── model_v2_fixed_final.pkl
├── model_v2_fixed_final_REAL.pkl    ← 🏆 อันไหนดีที่สุด?
├── model_best.pkl
├── model_best_new.pkl
├── model_production_maybe.pkl      ← 😬 งง!
├── model_john_test.pkl
├── model_mary_experiment.pkl
└── model_DONT_DELETE.pkl          ← 🤖 ใครจะกล้าลบ?
```

3.2 ปัญหาที่พบบ่อย

ปัญหา	คำอธิบาย	ผลกระทบ
🔍 หาโมเดลไม่เจอ	ไม่รู้ว่าโมเดลไหนดีที่สุด	ต้องฝึกใหม่ เสียเวลาและทรัพยากร
🌈 ไม่รู้ Hyperparameters	จำไม่ได้ว่าใช้ค่าอะไรฝึก	ไม่สามารถ Reproduce ได้
🔄 ไม่มี Version Control	ไม่สามารถย้อนกลับไปใช้โมเดลเก่าได้	เสี่ยงต่อการสูญเสียงานที่ทำไว้
👥 ทำงานเป็นทีมยาก	ไม่รู้ว่าใครแก้ไขอะไร เมื่อไหร่	เกิดความขัดแย้งและความสับสน
🚀 Deploy ยาก	ไม่มั่นใจว่าโมเดลไหนพร้อม Production	เสี่ยงต่อการ Deploy Model ที่ไม่ถูกต้อง
📄 ไม่มี Audit Trail	ไม่สามารถตรวจสอบย้อนหลังได้	ปัญหาด้าน Compliance และ Governance
🔗 ไม่มี Lineage	ไม่รู้ว่า Model มาจากข้อมูลและการทดลองใด	ยากต่อการ Debug และปรับปรุง

3.3 ผลกระทบทางธุรกิจ

- **เสียเวลา:** ต้องหา Model ที่ถูกต้องนานหลายชั่วโมง
- **เสียเงิน:** ต้องฝึก Model ใหม่เพราะหาของเดิมไม่เจอ
- **เสี่ยง:** Deploy Model ผิดตัวทำให้ผลลัพธ์ไม่ดี
- **Compliance:** ไม่สามารถตอบคำถาม Auditor ได้

4. Model Registry คืออะไร?

4.1 คำนิยาม

Model Registry คือระบบจัดการ Machine Learning Model แบบรวมศูนย์ (Centralized Model Management System) ที่ทำหน้าที่:

- 1. จัดเก็บ Model อย่างเป็นระบบ
- 2. ติดตาม Version ทั้งหมด
- 3. จัดการ Lifecycle ของ Model
- 4. บันทึก Metadata ที่เกี่ยวข้อง
- 5. ควบคุมการเข้าถึง และการ Deploy

4.2 โครงสร้างของ Model Registry



องค์ประกอบหลักของ Model Registry

Registered Model คือชื่อที่ใช้เรียก Model ในระบบ เปรียบเสมือน "โปรเจกต์" หนึ่งโปรเจกต์

คุณสมบัติ:

- มีชื่อเฉพาะ (Unique Name)
- มี Description อธิบายจุดประสงค์
- มี Tags สำหรับการจัดหมวดหมู่
- มีหลาย Versions ได้

ตัวอย่าง:

```
Registered Model: "fraud-detection-xgboost"  
├─ Description: "ตรวจจับการฉ้อโกงบัตรเครดิต"  
├─ Tags: [classification, xgboost, fraud, production]  
└─ Versions: [1, 2, 3, 4, 5]
```

Model Version คือ Instance หนึ่งของ Registered Model ที่สร้างจากการฝึกครั้งหนึ่ง

คุณสมบัติ:

- Version Number (1, 2, 3, ...)
- เชื่อมโยงกับ Run ID
- มี Stage หรือ Alias
- มี Metadata และ Artifacts

ตัวอย่าง:

```
Version 3:  
├─ Run ID: abc123xyz  
├─ Stage: Production  
├─ Alias: champion  
├─ Created: 2024-01-25 09:00:00  
├─ Metrics: {accuracy: 0.97, f1: 0.96}  
└─ Parameters: {n_estimators: 100, max_depth: 10}
```

Metadata คือข้อมูลที่อธิบาย Model:

```
# ตัวอย่าง Metadata
model_name: iris-classifier
version: 3
created_by: john.doe@company.com
created_at: 2024-01-25T09:00:00Z

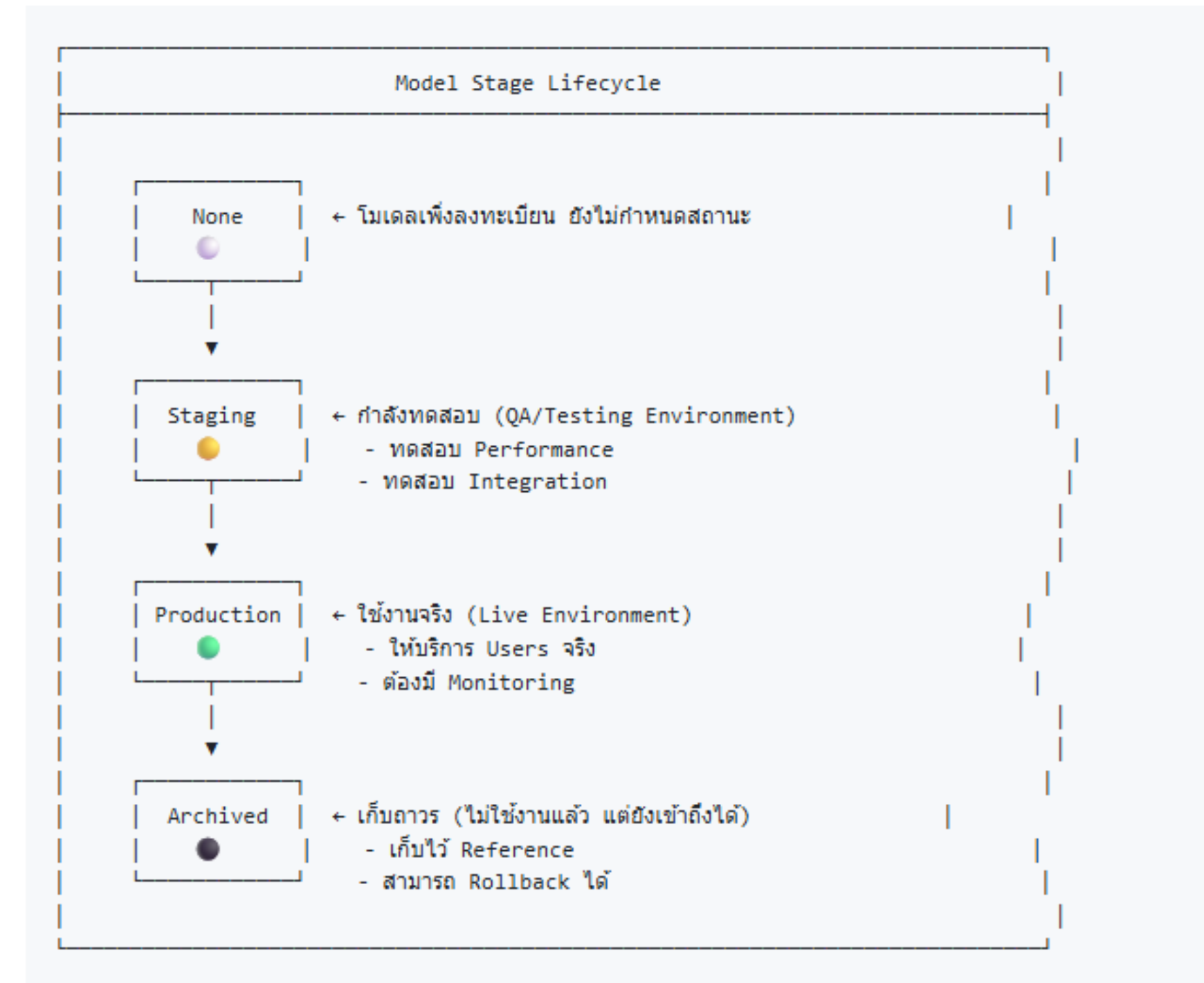
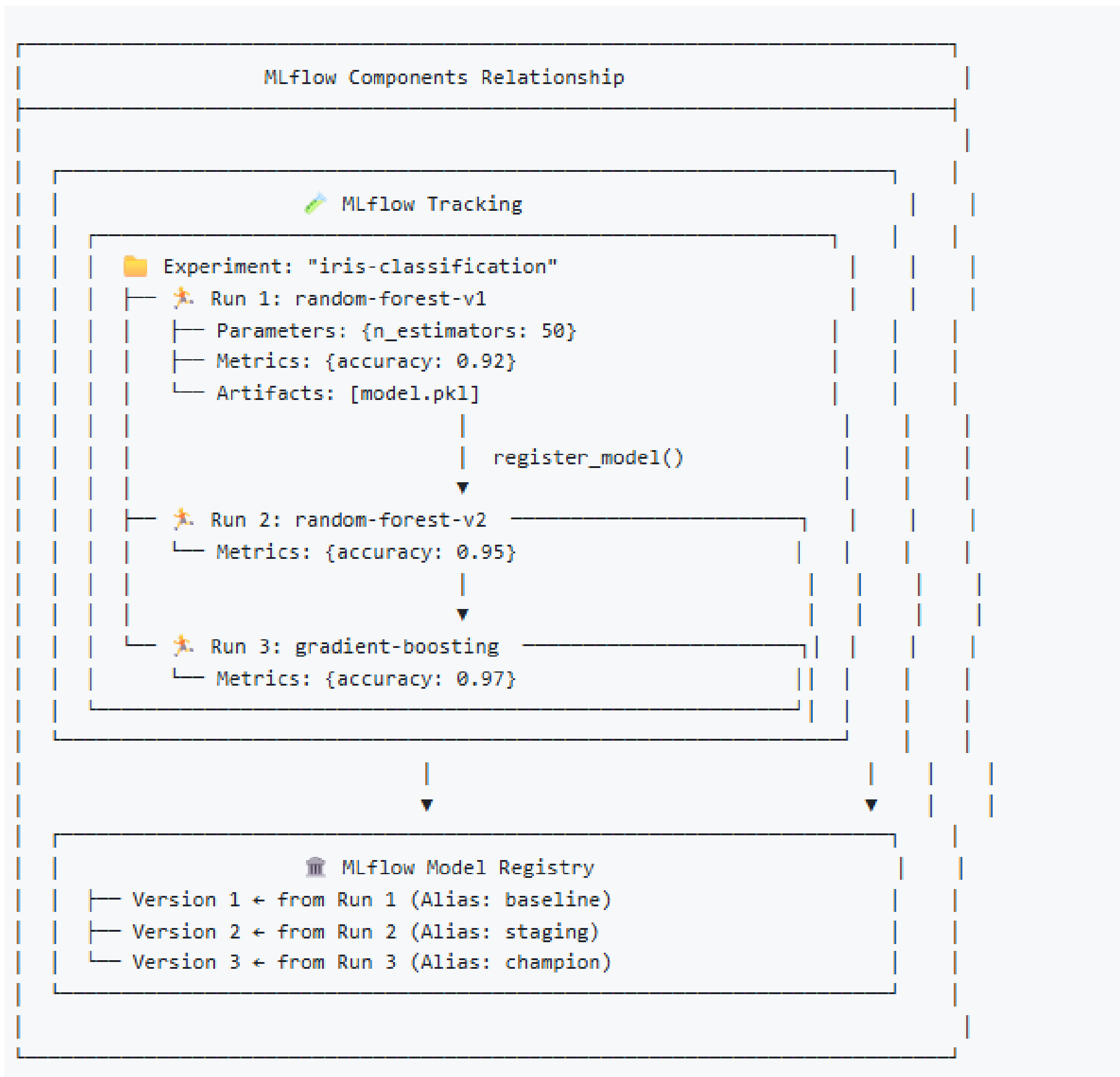
parameters:
  algorithm: GradientBoostingClassifier
  n_estimators: 100
  learning_rate: 0.1
  max_depth: 5

metrics:
  accuracy: 0.97
  f1_score: 0.96
  precision: 0.95
  recall: 0.97

tags:
  task: classification
  dataset: iris
  environment: production
  approved_by: ML-Team-Lead
```

Artifacts คือไฟล์ที่เกี่ยวข้องกับ Model:

Artifact	คำอธิบาย
MLmodel	ไฟล์ YAML ที่อธิบายโครงสร้าง Model
model.pkl	ไฟล์ Model (สำหรับ sklearn)
model.pth	ไฟล์ Model (สำหรับ PyTorch)
conda.yaml	Dependencies สำหรับรัน Model
requirements.txt	Python packages ที่ต้องการ
input_example.json	ตัวอย่าง Input data



Model Registry : Version 1

2.1 วิธีที่ 1: ลงทะเบียนพร้อม Train (Scikit-learn)


```
: from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score
from mlflow.models import infer_signature
import numpy as np

# โหลดข้อมูล
iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(
    iris.data, iris.target, test_size=0.2, random_state=42
)

# ตั้งชื่อ Registered Model
SKLEARN_MODEL_NAME = "iris-classifier-sklearn"

# สร้างหรือเลือก Experiment
mlflow.set_experiment("model-registry-lab")

2026/01/26 15:00:54 INFO mlflow.tracking.fluent: Experiment with name 'model-registry-lab' does not exist. Creating a new experiment.
: <Experiment: artifact_location='mlflow-artifacts:/1', creation_time=1769439654326, experiment_id='1', last_update_time=1769439654326, lifecycle_stage='active', name='model-registry-lab', tags={}>
```

 3.8.1

+ New

Home

Experiments

Models

Prompts

Experiments

Create Compare Delete

Filter experiments by name

Q

Tag filter

<input type="checkbox"/>	Name	Time created	Last modified	Description	Tags
<input type="checkbox"/>	model-registry-lab	01/26/2026, 10:00:54 PM	01/26/2026, 10:00:54 PM	-	
<input type="checkbox"/>	Default	01/26/2026, 09:58:39 PM	01/26/2026, 09:58:39 PM	-	

```

# Train และลงทะเบียน Model Version 1 (RandomForest n=50)
with mlflow.start_run(run_name="sklearn-rf-v1"):
    # Hyperparameters
    n_estimators = 50
    max_depth = 5

    # บันทึก Parameters
    mlflow.log_params({
        "n_estimators": n_estimators,
        "max_depth": max_depth,
        "model_type": "RandomForestClassifier",
        "version_note": "Baseline model"
    })

    # Train Model
    model_v1 = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        random_state=42
    )
    model_v1.fit(X_train, y_train)

    # Evaluate
    y_pred = model_v1.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    # บันทึก Metrics
    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    # สร้าง Signature
    signature = infer_signature(X_train, model_v1.predict(X_train))

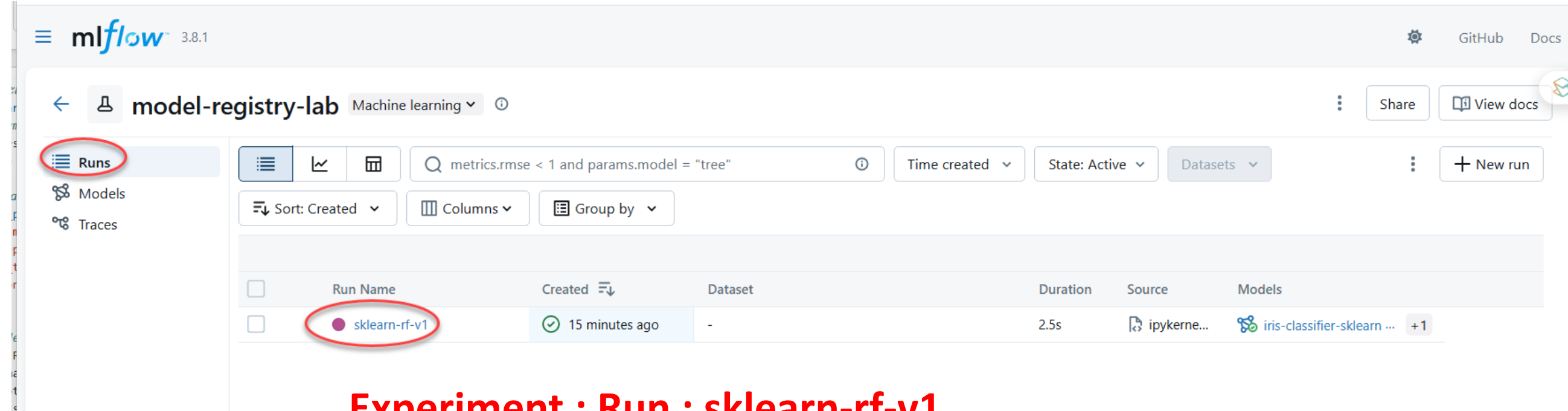
    # บันทึกและลงทะเบียน Model พร้อมกัน
    mlflow.sklearn.log_model(
        sk_model=model_v1,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3],
        registered_model_name=SKLEARN_MODEL_NAME # ← ลงทะเบียนทันที!
    )

sklearn_run_id_v1 = mlflow.active_run().info.run_id

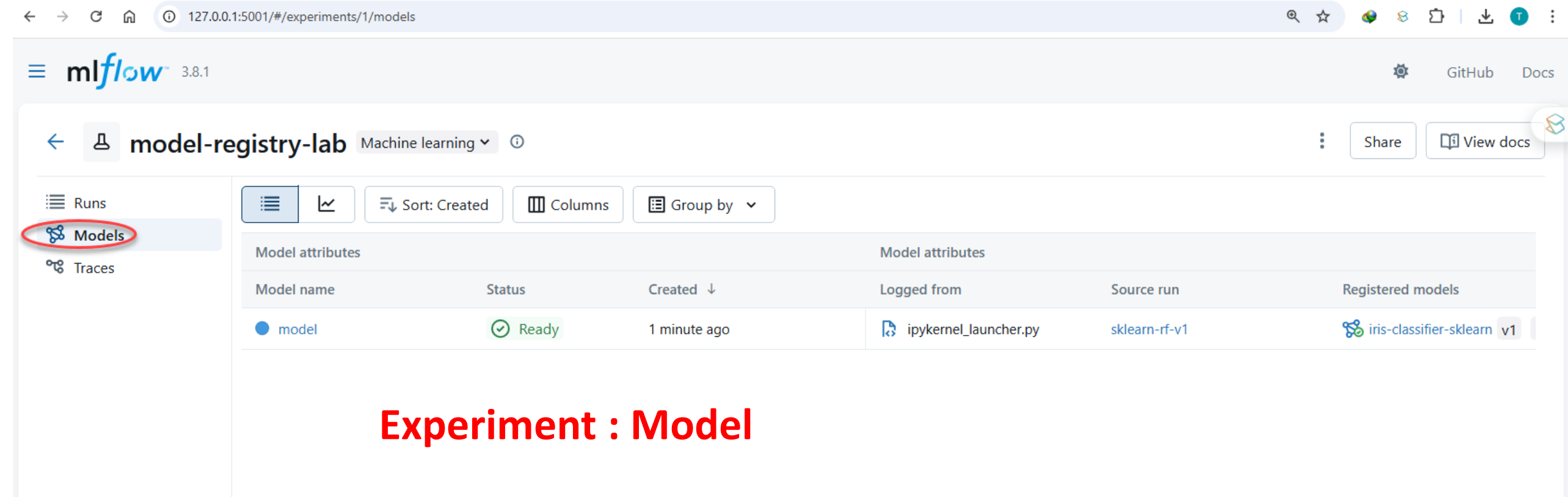
print(f"✅ ลงทะเบียน Model Version 1 สำเร็จ!")
print(f"📊 Accuracy: {accuracy:.4f}")
print(f"📊 F1 Score: {f1:.4f}")
print(f"📄 Run ID: {sklearn_run_id_v1}")

```

Model Version 1 (Random Forest) (50,5)



Experiment : Run : sklearn-rf-v1



Experiment : Model

```

# Train และลงทะเบียน Model Version 1 (RandomForest n=50)
with mlflow.start_run(run_name="sklearn-rf-v1"):
    # Hyperparameters
    n_estimators = 50
    max_depth = 5

    # บันทึก Parameters
    mlflow.log_params({
        "n_estimators": n_estimators,
        "max_depth": max_depth,
        "model_type": "RandomForestClassifier",
        "version_note": "Baseline model"
    })

    # Train Model
    model_v1 = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        random_state=42
    )
    model_v1.fit(X_train, y_train)

    # Evaluate
    y_pred = model_v1.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    # บันทึก Metrics
    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    # สร้าง Signature
    signature = infer_signature(X_train, model_v1.predict(X_train))

    # บันทึกและลงทะเบียน Model พร้อมกัน
    mlflow.sklearn.log_model(
        sk_model=model_v1,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3],
        registered_model_name=SKLEARN_MODEL_NAME # <- ลงทะเบียนพื้นที่!
    )

sklearn_run_id_v1 = mlflow.active_run().info.run_id

print(f"✅ ลงทะเบียน Model Version 1 สำเร็จ!")
print(f"📊 Accuracy: {accuracy:.4f}")
print(f"📊 F1 Score: {f1:.4f}")
print(f"📄 Run ID: {sklearn_run_id_v1}")

```

Experiment :Run tab

model-registry-lab > **Runs**

sklearn-rf-v1

Overview Model metrics System metrics Traces Artifacts

Metrics (2)

Search metrics

Metric	Value	Models
accuracy	1	model
f1_score	1	model

Parameters (4)

Search parameters

Parameter	Value
n_estimators	50
max_depth	5
model_type	RandomForestClassifier
version_note	Baseline model

Logged models (1)

Model attributes

Type	Step	Model name	Status	Created
Output	0	model	Ready	17 minutes ago

Model : name = iris-classifier-sklearn

```
# Train และลงทะเบียน Model Version 1 (RandomForest n=50)
with mlflow.start_run(run_name="sklearn-rf-v1"):
    # Hyperparameters
    n_estimators = 50
    max_depth = 5

    # บันทึก Parameters
    mlflow.log_params({
        "n_estimators": n_estimators,
        "max_depth": max_depth,
        "model_type": "RandomForestClassifier",
        "version_note": "Baseline model"
    })

    # Train Model
    model_v1 = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        random_state=42
    )
    model_v1.fit(X_train, y_train)

    # Evaluate
    y_pred = model_v1.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    # บันทึก Metrics
    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    # สร้าง Signature
    signature = infer_signature(X_train, model_v1.predict(X_train))

    # บันทึกและลงทะเบียน Model พร้อมกัน
    mlflow.sklearn.log_model(
        sk_model=model_v1,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3],
        registered_model_name=SKLEARN_MODEL_NAME # ← ลงทะเบียนทันที!
    )

sklearn_run_id_v1 = mlflow.active_run().info.run_id

print(f"✅ ลงทะเบียน Model Version 1 สำเร็จ!")
print(f"📊 Accuracy: {accuracy:.4f}")
print(f"📊 F1 Score: {f1:.4f}")
print(f"🆔 Run ID: {sklearn_run_id_v1}")
```

mlflow 3.8.1

+ New

Home Experiments **Models** Prompts

Registered Models

Share and manage machine learning models. [Learn more](#)

Filter registered models ...

Name	Latest ver...	Aliased versions	Created by	Last mod...	Tags
iris-classifier-sklearn	Version 1			01/27/20...	—

Create Model

mlflow 3.8.1

+ New

Home Experiments **Models** Prompts

Registered Models > iris-classifier-sklearn

Created Time: 01/27/2026, 10:53:47 AM Last Modified: 01/27/2026, 10:53:47 AM

> Description [Edit](#)

> Tags

▼ Versions [Compare](#)

Version	Registered at	Created by	Tags	Aliases	Description
✅ Version 1	01/27/2026, 10:53:47 AM		Add	Add	

< Previous Next > 25 / page

New model registry UI ☒


```
# เพิ่ม Description และ Tags ให้ Registered Model (ทำครั้งเดียวหลังสร้าง Model แรก)
client.update_registered_model(
    name=SKLEARN_MODEL_NAME,
    description="""
    🌸 Iris Flower Classification Model (Scikit-learn)

    Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)
    Input: 4 features (sepal length, sepal width, petal length, petal width)
    Output: Class prediction (0, 1, 2)

    Team: Data Science Team
    Contact: ds-team@example.com
    """
)

# เพิ่ม Tags ให้ Registered Model
client.set_registered_model_tag(SKLEARN_MODEL_NAME, "task", "classification")
client.set_registered_model_tag(SKLEARN_MODEL_NAME, "dataset", "iris")
client.set_registered_model_tag(SKLEARN_MODEL_NAME, "team", "data-science")
client.set_registered_model_tag(SKLEARN_MODEL_NAME, "framework", "sklearn")

# เพิ่ม Description และ Tags ให้ Version 1
client.update_model_version(
    name=SKLEARN_MODEL_NAME,
    version="1",
    description="Baseline model with RandomForest (n_estimators=50). Initial version for comparison."
)
client.set_model_version_tag(SKLEARN_MODEL_NAME, "1", "model_type", "RandomForest")
client.set_model_version_tag(SKLEARN_MODEL_NAME, "1", "status", "baseline")

print(f"✅ เพิ่ม Description และ Tags สำหรับ Model และ Version 1 สำเร็จ!")
```

+ New

- Home
- Experiments
- Models
- Prompts

Created Time: 01/26/2026, 10:06:32 PM Last Modified: 01/26/2026, 10:09:52 PM









Description Edit

Iris Flower Classification Model (Scikit-learn)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)
Input: 4 features (sepal length, sepal width, petal length, petal width)
Output: Class prediction (0, 1, 2)

Team: Data Science Team
Contact: ds-team@example.com



Tags

Name	Value	Actions
dataset	iris	 
framework	sklearn	 
task	classification	 
team	data-science	 

Add

Versions Compare

New model registry UI ☒

Version	Registered at	Created by	Tags	Aliases	Description
 Version 1	01/26/2026, 10:06:32 PM		<div>model_type: RandomForest</div> <div>status: baseline </div>	Add	Baseline model with RandomFor...

Model Registry : Version 2

Model Version 2 (Random Forest) (100,10)

```
# Train และลงทะเบียน Model Version 2 (RandomForest n=100)
with mlflow.start_run(run_name="sklearn-rf-v2"):

    # Hyperparameters ที่ปรับปรุง
    n_estimators = 100
    max_depth = 10

    mlflow.log_params({
        "n_estimators": n_estimators,
        "max_depth": max_depth,
        "model_type": "RandomForestClassifier",
        "version_note": "Improved model with more trees"
    })

    # Train Model
    model_v2 = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        random_state=42
    )
    model_v2.fit(X_train, y_train)

    # Evaluate
    y_pred = model_v2.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    signature = infer_signature(X_train, model_v2.predict(X_train))

    # ลงทะเบียนเป็น Version ใหม่ของ Model เดิม
    mlflow.sklearn.log_model(
        sk_model=model_v2,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3],
        registered_model_name=SKLEARN_MODEL_NAME
    )

sklearn_run_id_v2 = mlflow.active_run().info.run_id

print(f"✅ ลงทะเบียน Model Version 2 สำเร็จ!")
print(f"📊 Accuracy: {accuracy:.4f}")
print(f"📊 F1 Score: {f1:.4f}")
print(f"📄 Run ID: {sklearn_run_id_v2}")
```

model-registry-lab Machine learning

Runs Models Traces

metrics.rmse < 1 and params.model = "tree" Time created State: Active + New run

Sort: Created Columns Group by

	Run Name	Created	Dataset	Duration	Source	Models
	sklearn-rf-v2	6 minutes ago	-	2.6s	ipykerne...	iris-classifier-
	sklearn-rf-v1	52 minutes ago	-	2.5s	ipykerne...	iris-classifier-

Experiment : Run : sklearn-rf-v2

mlflow 3.8.1

+ New Home Experiments Models Prompts

Registered Models

Share and manage machine learning models. Learn more

Filter registered models by name...

Name	Latest version	Aliased versions	Created by	Last modified	Tags
iris-classifier-sklearn	Version 2			01/27/2026, 11:13:39 ...	task: classificationdataset: iriste

Experiment : Model

```
# Train และลงทะเบียน Model Version 2 (RandomForest n=100)
with mlflow.start_run(run_name="sklearn-rf-v2"):

    # Hyperparameters ที่ปรับปรุง
    n_estimators = 100
    max_depth = 10

    mlflow.log_params({
        "n_estimators": n_estimators,
        "max_depth": max_depth,
        "model_type": "RandomForestClassifier",
        "version_note": "Improved model with more trees"
    })

    # Train Model
    model_v2 = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        random_state=42
    )
    model_v2.fit(X_train, y_train)

    # Evaluate
    y_pred = model_v2.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    signature = infer_signature(X_train, model_v2.predict(X_train))

    # ลงทะเบียนเป็น Version ใหม่ของ Model เดิม
    mlflow.sklearn.log_model(
        sk_model=model_v2,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3],
        registered_model_name=SKLEARN_MODEL_NAME
    )

    sklearn_run_id_v2 = mlflow.active_run().info.run_id

    print(f"✅ ลงทะเบียน Model Version 2 สำเร็จ!")
    print(f"📊 Accuracy: {accuracy:.4f}")
    print(f"📊 F1 Score: {f1:.4f}")
    print(f"📄 Run ID: {sklearn_run_id_v2}")
```

mlflow 3.8.1

+ New

Home

Experiments

Models

Prompts

Registered Models >

iris-classifier-sklearn

Created Time: 01/27/2026, 10:53:47 AMLast Modified: 01/27/2026, 11:13:39 AM

Description

Edit

🌸 Iris Flower Classification Model (Scikit-learn)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)
Input: 4 features (sepal length, sepal width, petal length, petal width)
Output: Class prediction (0, 1, 2)

Team: Data Science Team
Contact: ds-team@example.com

Tags

Name	Value	Actions
dataset	iris	
framework	sklearn	
task	classification	
team	data-science	

Name

Value

Add

Versions

Compare

Version

Registered at

Created by

Tags

Aliases

Description

✅ Version 2

01/27/2026, 11:13:39 AM

Add

Add

Baseline model with Rando...

✅ Version 1

01/27/2026, 10:53:47 AM

model_type: RandomForest

status: baseline

Add

< Previous

Next >

25 / page

New model registry UI

☐


```
# Train และลงทะเบียน Model Version 2 (RandomForest n=100)
with mlflow.start_run(run_name="sklearn-rf-v2"):

    # Hyperparameters ที่ปรับปรุง
    n_estimators = 100
    max_depth = 10

    mlflow.log_params({
        "n_estimators": n_estimators,
        "max_depth": max_depth,
        "model_type": "RandomForestClassifier",
        "version_note": "Improved model with more trees"
    })

    # Train Model
    model_v2 = RandomForestClassifier(
        n_estimators=n_estimators,
        max_depth=max_depth,
        random_state=42
    )
    model_v2.fit(X_train, y_train)

    # Evaluate
    y_pred = model_v2.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    signature = infer_signature(X_train, model_v2.predict(X_train))

    # ลงทะเบียนเป็น Version ใหม่ของ Model เดิม
    mlflow.sklearn.log_model(
        sk_model=model_v2,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3],
        registered_model_name=SKLEARN_MODEL_NAME
    )

    sklearn_run_id_v2 = mlflow.active_run().info.run_id

    print(f"✅ ลงทะเบียน Model Version 2 สำเร็จ!")
    print(f"📊 Accuracy: {accuracy:.4f}")
    print(f"📊 F1 Score: {f1:.4f}")
    print(f"📄 Run ID: {sklearn_run_id_v2}")
```

```
# เพิ่ม Description และ Tags ให้ Version 2
client.update_model_version(
    name=SKLEARN_MODEL_NAME,
    version="2",
    description="Improved RandomForest (n_estimators=100). Better accuracy than baseline."
)

client.set_model_version_tag(SKLEARN_MODEL_NAME, "2", "model_type", "RandomForest")
client.set_model_version_tag(SKLEARN_MODEL_NAME, "2", "status", "improved")

print(f"✅ เพิ่ม Description และ Tags สำหรับ Version 2 สำเร็จ!")

✅ เพิ่ม Description และ Tags สำหรับ Version 2 สำเร็จ!
```

mlflow 3.8.1

+ New

Home

Experiments

Models

Prompts

Registered Models >

iris-classifier-sklearn

Created Time: 01/26/2026, 10:06:32 PM Last Modified: 01/26/2026, 10:51:59 PM

Description Edit

🌸 Iris Flower Classification Model (Scikit-learn)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)
Input: 4 features (sepal length, sepal width, petal length, petal width)
Output: Class prediction (0, 1, 2)

Team: Data Science Team
Contact: ds-team@example.com

Tags

Name	Value	Actions
dataset	iris	
framework	sklearn	
task	classification	
team	data-science	

Add

Versions Compare

Version	Registered at	Created by	Tags	Aliases	Description
✅ Version 2	01/26/2026, 10:51:59 PM		<div>model_type: RandomForest</div> <div>status: improved </div>	Add	Improved RandomForest (n_esti...
✅ Version 1	01/26/2026, 10:06:32 PM		<div>model_type: RandomForest</div> <div>status: baseline </div>	Add	Baseline model with RandomFor...

New model registry UI ☒

Model Registry : Register Version 3 Later

```
# Train Model ใหม่โดยยังไม่ลงทะเบียน
with mlflow.start_run(run_name="sklearn-gb-candidate") as run:
```

```
    mlflow.log_params({
        "n_estimators": 100,
        "learning_rate": 0.1,
        "max_depth": 5,
        "model_type": "GradientBoostingClassifier",
        "version_note": "Gradient Boosting candidate"
    })

    model_gb = GradientBoostingClassifier(
        n_estimators=100,
        learning_rate=0.1,
        max_depth=5,
        random_state=42
    )
    model_gb.fit(X_train, y_train)

    y_pred = model_gb.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred, average='weighted')

    mlflow.log_metrics({
        "accuracy": accuracy,
        "f1_score": f1
    })

    signature = infer_signature(X_train, model_gb.predict(X_train))

    # บันทึก Model แต่ยังไม่ลงทะเบียน
    mlflow.sklearn.log_model(
        sk_model=model_gb,
        artifact_path="model",
        signature=signature,
        input_example=X_train[:3]
    )

    candidate_run_id = run.info.run_id

    print(f"✅ บันทึก Model สำเร็จ (ยังไม่ลงทะเบียน)")
    print(f"📊 Accuracy: {accuracy:.4f}")
    print(f"📊 F1 Score: {f1:.4f}")
    print(f"🆔 Run ID: {candidate_run_id}")
```

ไม่มี parameter
registered_model_name=SKLEARN_MODEL_NAME

```
2026/01/27 04:49:49 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
✅ บันทึก Model สำเร็จ (ยังไม่ลงทะเบียน)
📊 Accuracy: 1.0000
📊 F1 Score: 1.0000
🆔 Run ID: d0a73c4f10bf4b56bb2af493fe0ca248
🔗 View run sklearn-gb-candidate at: http://127.0.0.1:5000/#/experiments/1/runs/d0a73c4f10bf4b56bb2af493fe0ca248
🌱 View experiment at: http://127.0.0.1:5000/#/experiments/1
```

Registered Models >

iris-classifier-sklearn

Created Time: 01/27/2026, 10:53:47 AM Last Modified: 01/27/2026, 11:13:39 AM









Description Edit

🌸 Iris Flower Classification Model (Scikit-learn)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)
Input: 4 features (sepal length, sepal width, petal length, petal width)
Output: Class prediction (0, 1, 2)

Team: Data Science Team
Contact: ds-team@example.com

Tags

Name	Value	Actions
dataset	iris	 
framework	sklearn	 
task	classification	 
team	data-science	 

Versions Compare

Version	Registered at	Created by	Tags	Aliases	Description
🟢 Version 2	01/27/2026, 11:13:39 AM		model_type: RandomForest status: improved 	Add	Improved RandomForest (n_esti...
🟢 Version 1	01/27/2026, 10:53:47 AM		model_type: RandomForest status: baseline 	Add	Baseline model with RandomFor...

New model registry UI ☒

ลงทะเบียน Version 3 ไม่ได้

Train Model ใหม่โดยยังไม่ลงทะเบียน

with mlflow.start_run(run_name="sklearn-gb-candidate") as run:

```
mlflow.log_params({
    "n_estimators": 100,
    "learning_rate": 0.1,
    "max_depth": 5,
    "model_type": "GradientBoostingClassifier",
    "version_note": "Gradient Boosting candidate"
})

model_gb = GradientBoostingClassifier(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=5,
    random_state=42
)
model_gb.fit(X_train, y_train)

y_pred = model_gb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average='weighted')

mlflow.log_metrics({
    "accuracy": accuracy,
    "f1_score": f1
})
```

signature = infer_signature(X_train, model_gb.predict(X_train))

บันทึก Model แต่ยังไม่ลงทะเบียน

```
mlflow.sklearn.log_model(
    sk_model=model_gb,
    artifact_path="model",
    signature=signature,
    input_example=X_train[:3]
)
```

ไม่มี parameter
registered_model_name=SKLEARN_MODEL_NAME

candidate_run_id = run.info.run_id

```
print(f"✅ บันทึก Model สำเร็จ (ยังไม่ลงทะเบียน)")
print(f"📊 Accuracy: {accuracy:.4f}")
print(f"📊 F1 Score: {f1:.4f}")
print(f"📄 Run ID: {candidate_run_id}")
```

2026/01/27 04:49:49 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.

✅ บันทึก Model สำเร็จ (ยังไม่ลงทะเบียน)

📊 Accuracy: 1.0000

📊 F1 Score: 1.0000

📄 Run ID: d0a73c4f10bf4b56bb2af493fe0ca248

🔗 View run sklearn-gb-candidate at: <http://127.0.0.1:5000/#/experiments/1/runs/d0a73c4f10bf4b56bb2af493fe0ca248>

🌿 View experiment at: <http://127.0.0.1:5000/#/experiments/1>

ลงทะเบียนภายหลังจาก Run ที่มีอยู่แล้ว

ใช้เมื่อต้องการ:

- เลือก Model ที่ดีที่สุดจากหลายๆ Run ก่อนลงทะเบียน
- ลงทะเบียน Model ที่ Train ไว้ก่อนหน้านี้

```
# ดึงชื่อไฟล์ลงทะเบียน Model ภายหลัง
model_uri = f"runs:/candidate_run_id/model"

# ลงทะเบียนเป็น Version ใหม่
result = mlflow.register_model(
    model_uri=model_uri,
    name=SKLEARN_MODEL_NAME
)

print(f"✅ ลงทะเบียน Model ภายหลังสำเร็จ!")
print(f"📄 Model Name: {result.name}")
print(f"🌟 Version: {result.version}")

# เพิ่ม Description และ Tags ให้ Version 3 (GradientBoosting)
client.update_model_version(
    name=SKLEARN_MODEL_NAME,
    version="3",
    description="GradientBoosting model. Best performance, deployed as champion."
)
client.set_model_version_tag(SKLEARN_MODEL_NAME, "3", "model_type", "GradientBoosting")
client.set_model_version_tag(SKLEARN_MODEL_NAME, "3", "deployment_status", "production")
client.set_model_version_tag(SKLEARN_MODEL_NAME, "3", "approved_by", "ML-Team-Lead")
client.set_model_version_tag(SKLEARN_MODEL_NAME, "3", "approval_date", "2024-01-25")

print(f"✅ เพิ่ม Description และ Tags สำหรับ Version 3 สำเร็จ!")
```

[+ New](#)[Home](#)[Experiments](#)[Models](#)[Prompts](#)[Registered Models](#) >

iris-classifier-sklearn

Created Time: 01/27/2026, 10:53:47 AM

Last Modified: 01/27/2026, 12:06:35 PM

[Description](#) [Edit](#)

🌸 Iris Flower Classification Model (Scikit-learn)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)

Input: 4 features (sepal length, sepal width, petal length, petal width)

Output: Class prediction (0, 1, 2)

Team: Data Science Team

Contact: ds-team@example.com

[Tags](#)

Name	Value	Actions
dataset	iris	✎ 🗑
framework	sklearn	✎ 🗑
task	classification	✎ 🗑
team	data-science	✎ 🗑

 [Versions](#) [Compare](#)New model registry UI ☒

Version	Registered at ↕	Created by	Tags	Aliases	Description
✔ Version 3	01/27/2026, 12:06:35 PM		<div>model_type: GradientBoosting</div> <div>deployment_status: production</div> <div>approved_by: ML-Team-Lead</div> <div>approval_date: 2024-01-25 ✎</div>	Add	GradientBoosting model. Best ...
✔ Version 2	01/27/2026, 11:13:39 AM		<div>model_type: RandomForest</div> <div>status: improved ✎</div>	Add	Improved RandomForest (n_esti...
✔ Version 1	01/27/2026, 10:53:47 AM		<div>model_type: RandomForest</div> <div>status: baseline ✎</div>	Add	Baseline model with RandomFor...

Model Registry : **Viewing Registered Models** **Information**

การดูข้อมูล Registered Models

แนวคิด

หลังจากลงทะเบียน Model แล้ว สามารถดูข้อมูลได้หลายวิธี

ฟังก์ชันสำคัญ

ฟังก์ชัน	คำอธิบาย
<code>client.search_registered_models()</code>	ค้นหา Registered Models ทั้งหมด
<code>client.get_registered_model(name)</code>	ดูข้อมูล Model ตามชื่อ
<code>client.search_model_versions(filter_string)</code>	ค้นหา Versions ของ Model
<code>client.get_model_version(name, version)</code>	ดูข้อมูล Version เฉพาะ

```
# ดู Registered Models ทั้งหมด
print("📦 Registered Models ทั้งหมด:")
print("=" * 60)

for rm in client.search_registered_models():
    print(f"\n📁 Model Name: {rm.name}")
    print(f"📄 Description: {rm.description[:50] + '...' if rm.description and len(rm.description) > 50 else rm.description or 'ไม่มี'}")
    print(f"📅 Created: {rm.creation_timestamp}")
    print(f"📅 Last Updated: {rm.last_updated_timestamp}")
```

📦 Registered Models ทั้งหมด:

=====

📁 Model Name: iris-classifier-sklearn
📄 Description:
🌸 Iris Flower Classification Model (Scikit-le...
📅 Created: 1769486027965
📅 Last Updated: 1769490395451

```
# ดูข้อมูล Versions ของ Model เฉพาะ
print(f"\n🚀 Versions ของ '{SKLEARN_MODEL_NAME}':")
print("=" * 60)

versions = client.search_model_versions(f"name='{SKLEARN_MODEL_NAME}'")

for v in versions:
    print(f"\n Version {v.version}:")
    print(f"    - Status: {v.status}")
    print(f"    - Stage: {v.current_stage}")
    print(f"    - Run ID: {v.run_id}")
    print(f"    - Created: {v.creation_timestamp}")
```

🚀 Versions ของ 'iris-classifier-sklearn':

=====

Version 3:

- Status: READY
- Stage: None
- Run ID: 3c697061521342e993e7ad99bbe7e25f
- Created: 1769443945598

Version 2:

- Status: READY
- Stage: None
- Run ID: 5fd7e2bfa8ce4c3b8673d7bbe395979b
- Created: 1769442719628

Version 1:

- Status: READY
- Stage: None
- Run ID: f74a11d547014a3e9a81b757c942e74f
- Created: 1769439992169

```
# ดูข้อมูล Version เฉพาะพร้อม Metrics จาก Run
print(f"\n📊 รายละเอียด Version พร้อม Metrics:")
print("=" * 60)

for v in versions:
    print(f"\n🚀 Version {v.version}:")

    # ดึงข้อมูล Run ที่เกี่ยวข้อง
    run = client.get_run(v.run_id)

    # แสดง Parameters
    print(f"    Parameters:")
    for key, value in run.data.params.items():
        print(f"        - {key}: {value}")

    # แสดง Metrics
    print(f"    Metrics:")
    for key, value in run.data.metrics.items():
        print(f"        - {key}: {value:.4f}")
```

📊 รายละเอียด Version พร้อม Metrics:

=====

🚀 Version 3:

Parameters:

- n_estimators: 100
- learning_rate: 0.1
- max_depth: 5
- model_type: GradientBoostingClassifier
- version_note: Gradient Boosting candidate

Metrics:

- accuracy: 1.0000
- f1_score: 1.0000

🚀 Version 2:

Parameters:

- n_estimators: 100
- max_depth: 10
- model_type: RandomForestClassifier
- version_note: Improved model with more trees

Metrics:

- accuracy: 1.0000
- f1_score: 1.0000

🚀 Version 1:

Parameters:

- n_estimators: 50
- max_depth: 5
- model_type: RandomForestClassifier
- version_note: Baseline model

Metrics:

- accuracy: 1.0000
- f1_score: 1.0000

```
[12]: # ดู Tags ทั้งหมดของ Model และ Versions
model_info = client.get_registered_model(SKLEARN_MODEL_NAME)
print(f"\n 📄 Tags ของ '{SKLEARN_MODEL_NAME}':")
for tag in model_info.tags:
    print(f"    - {tag}: {model_info.tags[tag]}")

version_info = client.get_model_version(SKLEARN_MODEL_NAME, "3")
print(f"\n 📄 Tags ของ Version 3:")
for tag in version_info.tags:
    print(f"    - {tag}: {version_info.tags[tag]}")
```

```
📄 Tags ของ 'iris-classifier-sklearn':
- task: classification
- dataset: iris
- team: data-science
- framework: sklearn
```

```
📄 Tags ของ Version 3:
- model_type: GradientBoosting
- deployment_status: production
- approved_by: ML-Team-Lead
- approval_date: 2024-01-25
```

Model Registry : Managing Model Stages

Aliases (การตั้งชื่อแบบแฝงให้ **Model**)

การจัดการ Model Stages Aliases (การตั้งชื่อนามแฝงให้ Model)

ฟังก์ชันสำคัญ (MLflow 2.x - Aliases)

ฟังก์ชัน	คำอธิบาย
<code>client.set_registered_model_alias(name, alias, version)</code>	กำหนด Alias ให้ Version
<code>client.delete_registered_model_alias(name, alias)</code>	ลบ Alias
<code>client.get_model_version_by_alias(name, alias)</code>	ดึง Version ตาม Alias

4.1 การใช้ Model Aliases (แนะนำสำหรับ MLflow 2.x)

```
# กำหนด Aliases ให้แต่ละ Version

# Version 1: กำหนดเป็น "baseline"
client.set_registered_model_alias(
    name=SKLEARN_MODEL_NAME,
    alias="baseline",
    version="1"
)
print(f"✅ กำหนด Alias 'baseline' ให้ Version 1")

# Version 2: กำหนดเป็น "staging"
client.set_registered_model_alias(
    name=SKLEARN_MODEL_NAME,
    alias="staging",
    version="2"
)
print(f"✅ กำหนด Alias 'staging' ให้ Version 2")

# Version 3: กำหนดเป็น "champion" (Production)
client.set_registered_model_alias(
    name=SKLEARN_MODEL_NAME,
    alias="champion",
    version="3"
)
print(f"✅ กำหนด Alias 'champion' ให้ Version 3")
```

- ✅ กำหนด Alias 'baseline' ให้ Version 1
- ✅ กำหนด Alias 'staging' ให้ Version 2
- ✅ กำหนด Alias 'champion' ให้ Version 3

```
# ดู Model Version จาก Alias
champion_version = client.get_model_version_by_alias(
    name=SKLEARN_MODEL_NAME,
    alias="champion"
)

print(f"\n🏆 Champion Model:")
print(f"    Version: {champion_version.version}")
print(f"    Run ID: {champion_version.run_id}")
```

🏆 Champion Model:
Version: 3
Run ID: 466523af27cd4d05b7d4db7dc90750de

```
# แสดง Aliases ทั้งหมดของ Model
model_info = client.get_registered_model(SKLEARN_MODEL_NAME)
print(f"\n📁 Aliases ของ '{SKLEARN_MODEL_NAME}':")
print(f"    {model_info.aliases}")
```

📁 Aliases ของ 'iris-classifier-sklearn':
{'baseline': '1', 'champion': '3', 'staging': '2'}

+ New

- Home
- Experiments
- Models**
- Prompts

Registered Models >

iris-classifier-sklearn

Created Time: 01/26/2026, 10:06:32 PM

Last Modified: 01/26/2026, 11:12:25 PM

Description Edit

Iris Flower Classification Model (Scikit-learn)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด (Setosa, Versicolor, Virginica)









Input: 4 features (sepal length, sepal width, petal length, petal width)

Output: Class prediction (0, 1, 2)

Team: Data Science Team

Contact: ds-team@example.com










Tags

Name	Value	Actions
dataset	iris	 
framework	sklearn	 
task	classification	 
team	data-science	 

<input type="text" value="Name"/>	<input type="text" value="Value"/>	<input type="button" value="Add"/>
-----------------------------------	------------------------------------	------------------------------------

Versions Compare

New model registry UI ☒

Version	Registered at	Created by	Tags	Aliases	Description
 Version 3	01/26/2026, 11:12:25 PM		model_type: GradientBoosting deployment_status: production approved_by: ML-Team-Lead approval_date: 2024-01-25 	@ champion 	GradientBoosting model. Best ...
 Version 2	01/26/2026, 10:51:59 PM		model_type: RandomForest status: improved 	@ staging 	Improved RandomForest (n_esti...
 Version 1	01/26/2026, 10:06:32 PM		model_type: RandomForest status: baseline 	@ baseline 	Baseline model with RandomFor...

Model Registry : Loading Model from Registry

การโหลด Model จาก Registry

แนวคิด

การโหลด Model จาก Registry มีหลายวิธี ขึ้นอยู่กับ Use Case:

วิธี	URI Format	Use Case
โดย Model Name + Version	<code>models:/<model_name>/<version></code>	โหลด Version เฉพาะ
โดย Model Name + Alias	<code>models:/<model_name>@<alias></code>	โหลดตาม Alias (เช่น champion)
โดย Run ID	<code>runs:/<run_id>/<artifact_path></code>	โหลดจาก Run โดยตรง
โดย ARTIFACTS_BASE + Search	Local path	โหลดโดยตรงจาก Disk (เร็วที่สุด)

ฟังก์ชันสำคัญ

ฟังก์ชัน	คำอธิบาย
<code>mlflow.sklearn.load_model(model_uri)</code>	โหลด Scikit-learn Model
<code>mlflow.pytorch.load_model(model_uri)</code>	โหลด PyTorch Model
<code>mlflow.pyfunc.load_model(model_uri)</code>	โหลด Model แบบ Generic

5.1 โหลด Model โดยใช้ Model Name และ Version

```
# โหลด Model โดยระบุ Version
print("📁 วิธีที่ 1: โหลด Model โดยระบุ Version")
print("=" * 60)

# โหลด Version 1
model_uri_v1 = f"models://{SKLEARN_MODEL_NAME}/1"
print(f"\n🔗 Model URI: {model_uri_v1}")

loaded_model_v1 = mlflow.sklearn.load_model(model_uri_v1)
print(f"✅ โหลด Version 1 สำเร็จ: {type(loaded_model_v1)}")

# ทดสอบทำนาย
predictions_v1 = loaded_model_v1.predict(X_test[:5])
print(f"🔮 Predictions (v1): {predictions_v1}")
print(f"📋 Actual: {y_test[:5]}")

📁 วิธีที่ 1: โหลด Model โดยระบุ Version
=====

🔗 Model URI: models:/iris-classifier-sklearn/1
✅ โหลด Version 1 สำเร็จ: <class 'sklearn.ensemble._forest.RandomForestClassifier'>
🔮 Predictions (v1): [1 0 2 1 1]
📋 Actual: [1 0 2 1 1]

# โหลด Version 3
model_uri_v3 = f"models://{SKLEARN_MODEL_NAME}/3"
print(f"\n🔗 Model URI: {model_uri_v3}")

loaded_model_v3 = mlflow.sklearn.load_model(model_uri_v3)
print(f"✅ โหลด Version 3 สำเร็จ: {type(loaded_model_v3)}")

# ทดสอบทำนาย
predictions_v3 = loaded_model_v3.predict(X_test[:5])
print(f"🔮 Predictions (v3): {predictions_v3}")
print(f"📋 Actual: {y_test[:5]}")

🔗 Model URI: models:/iris-classifier-sklearn/3
✅ โหลด Version 3 สำเร็จ: <class 'sklearn.ensemble._gb.GradientBoostingClassifier'>
🔮 Predictions (v3): [1 0 2 1 1]
📋 Actual: [1 0 2 1 1]
```

5.2 โหลด Model โดยใช้ Alias (แนะนำสำหรับ Production)

```
# โหลด Model โดยใช้ Alias
print("\n 📁 วิธีที่ 2: โหลด Model โดยใช้ Alias")
print("=" * 60)

# โหลด Champion Model
model_uri_champion = f"models://{SKLEARN_MODEL_NAME}@champion"
print(f"\n 🔗 Model URI: {model_uri_champion}")

loaded_champion = mlflow.sklearn.load_model(model_uri_champion)
print(f"✅ โหลด Champion Model สำเร็จ: {type(loaded_champion)}")

# ทดสอบทำนาย
predictions_champion = loaded_champion.predict(X_test[:5])
print(f"🧠 Predictions (champion): {predictions_champion}")
print(f"📄 Actual: {y_test[:5]}")
```

📁 วิธีที่ 2: โหลด Model โดยใช้ Alias

=====

```
🔗 Model URI: models:/iris-classifier-sklearn@champion
✅ โหลด Champion Model สำเร็จ: <class 'sklearn.ensemble._gb.GradientBoostingClassifier'>
🧠 Predictions (champion): [1 0 2 1 1]
📄 Actual: [1 0 2 1 1]
```

```
# โหลด Baseline Model
model_uri_baseline = f"models://{SKLEARN_MODEL_NAME}@baseline"
print(f"\n 🔗 Model URI: {model_uri_baseline}")

loaded_baseline = mlflow.sklearn.load_model(model_uri_baseline)
print(f"✅ โหลด Baseline Model สำเร็จ: {type(loaded_baseline)}")

# เปรียบเทียบ Accuracy
accuracy_baseline = accuracy_score(y_test, loaded_baseline.predict(X_test))
accuracy_champion = accuracy_score(y_test, loaded_champion.predict(X_test))

print(f"\n 📊 เปรียบเทียบ Accuracy:")
print(f"   Baseline: {accuracy_baseline:.4f}")
print(f"   Champion: {accuracy_champion:.4f}")
print(f"   ปรับปรุง: +{(accuracy_champion - accuracy_baseline) * 100:.2f}%")
```

```
🔗 Model URI: models:/iris-classifier-sklearn@baseline
✅ โหลด Baseline Model สำเร็จ: <class 'sklearn.ensemble._forest.RandomForestClassifier'>
```

```
📊 เปรียบเทียบ Accuracy:
Baseline: 1.0000
Champion: 1.0000
ปรับปรุง: +0.00%
```


Model Registry with Pytorch

ส่วนที่ 6: การลงทะเบียน PyTorch Model

แนวคิด

PyTorch Model สามารถลงทะเบียนได้เหมือนกับ Scikit-learn โดยใช้ `mlflow.pytorch.log_model()` พร้อม `registered_model_name`

```
import torch
import torch.nn as nn

# สร้าง Simple Neural Network
class IrisClassifier(nn.Module):
    def __init__(self, input_size=4, hidden_size=16, num_classes=3):
        super(IrisClassifier, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size, num_classes)

    def forward(self, x):
        out = self.fc1(x)
        out = self.relu(out)
        out = self.fc2(out)
        return out

# ตั้งชื่อ Registered Model สำหรับ PyTorch
PYTORCH_MODEL_NAME = "iris-classifier-pytorch"
```

```
# Train และลงทะเบียน PyTorch Model Version 1
with mlflow.start_run(run_name="pytorch-nn-v1"):

    # Hyperparameters
    hidden_size = 16
    learning_rate = 0.01
    epochs = 100

    mlflow.log_params({
        "hidden_size": hidden_size,
        "learning_rate": learning_rate,
        "epochs": epochs,
        "model_type": "SimpleNN"
    })

    # สร้าง Model
    pytorch_model = IrisClassifier(hidden_size=hidden_size)

    # เตรียมข้อมูล
    X_train_float32 = X_train.astype('float32')
    X_test_float32 = X_test.astype('float32')
    X_tensor = torch.FloatTensor(X_train_float32)
    y_tensor = torch.LongTensor(y_train)

    # Training
    criterion = nn.CrossEntropyLoss()
    optimizer = torch.optim.Adam(pytorch_model.parameters(), lr=learning_rate)

    print("🚀 Training PyTorch Model...")
    for epoch in range(epochs):
        outputs = pytorch_model(X_tensor)
        loss = criterion(outputs, y_tensor)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if (epoch + 1) % 25 == 0:
            mlflow.log_metric("train_loss", loss.item(), step=epoch)
            print(f"    Epoch [{epoch+1}/{epochs}], Loss: {loss.item():.4f}")
```

```
# Evaluate
pytorch_model.eval()
with torch.no_grad():
    X_test_tensor = torch.FloatTensor(X_test_float32)
    outputs = pytorch_model(X_test_tensor)
    _, predicted = torch.max(outputs.data, 1)
    accuracy = (predicted.numpy() == y_test).sum() / len(y_test)

    mlflow.log_metric("accuracy", accuracy)
    print(f"📊 Test Accuracy: {accuracy:.4f}")

# สร้าง Signature
signature = infer_signature(
    X_train_float32,
    pytorch_model(torch.FloatTensor(X_train_float32)).detach().numpy()
)

# บันทึกและลงทะเบียน PyTorch Model
mlflow.pytorch.log_model(
    pytorch_model=pytorch_model,
    artifact_path="model",
    signature=signature,
    input_example=X_train_float32[:3],
    registered_model_name=PYTORCH_MODEL_NAME
)

pytorch_run_id_v1 = mlflow.active_run().info.run_id

print(f"✅ ลงทะเบียน PyTorch Model Version 1 สำเร็จ!")
print(f"📄 Run ID: {pytorch_run_id_v1}")
```

```
2026/01/27 06:21:17 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2026/01/27 06:21:17 WARNING mlflow.utils.requirements_utils: Found torch version (2.9.1+cpu) contains a local version label (+cpu). MLflow logged a pip requirement
🚀 Training PyTorch Model...
Epoch [25/100], Loss: 0.5775
Epoch [50/100], Loss: 0.2918
Epoch [75/100], Loss: 0.1665
Epoch [100/100], Loss: 0.1143

📊 Test Accuracy: 1.0000
2026/01/27 06:21:19 WARNING mlflow.utils.requirements_utils: Found torch version (2.9.1+cpu) contains a local version label (+cpu). MLflow logged a pip requirement
Successfully registered model 'iris-classifier-pytorch'.
2026/01/27 06:21:19 INFO mlflow.store.model_registry.abstract_store: Waiting up to 300 seconds for model version to finish creation. Model name: iris-classifier-py
✅ ลงทะเบียน PyTorch Model Version 1 สำเร็จ!
📄 Run ID: 1ff8550bb4d14a29b836fa7489e468e4
🔗 View run pytorch-nn-v1 at: http://127.0.0.1:5000/#/experiments/1/runs/1ff8550bb4d14a29b836fa7489e468e4
🌱 View experiment at: http://127.0.0.1:5000/#/experiments/1
Created version '1' of model 'iris-classifier-pytorch'.
```

≡ mlflow 3.8.1

←

🔖

model-registry-lab

Machine learning ▾

ⓘ

☰

Runs

🔗

Models

🔗

Traces

☰

📈

📊

🔍

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

⌵

Sort: Created ▾

☰

Columns ▾

☰

Group by ▾

☐

Run Name

Created ⌵

Dataset

☐

● pytorch-nn-v1

✅ 4 minutes ago

-

≡ mlflow 3.8.1

⚙

model-registry-lab > Runs >

🔗 pytorch-nn-v1

Overview

Model metrics

System metrics

Traces

Artifacts

Description 📝

No description

Metrics (2)

🔍 Search metrics

Metric	Value	Models
train_loss	0.08633295446634293	-
accuracy	1	model

Parameters (4)

🔍 Search parameters

Parameter	Value
hidden_size	16
learning_rate	0.01
epochs	100
model_type	SimpleNN

Logged models (1)

☰

About this run

Created at

01/28/2026, 09:12:15 AM

Created by

root

Experiment ID

1 📄

Status

✅ Finished

Run ID

f120e12573f44a249fa4fac92e715a77 📄

Duration

3.6s

Source

🔗 ipykernel_launcher.py

Registered prompts

—

Datasets

None

Tags

Add tags

Registered models

🔗 iris-classifier-pytorch

v1

≡ mlflow 3.8.1

model-registry-lab > Runs >

🔗 pytorch-nn-v1

Overview

Model metrics

System metrics

Traces

Artifacts

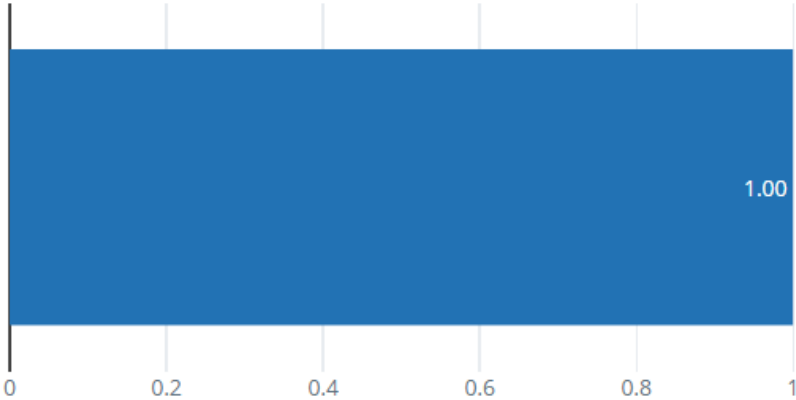
🔍 Search metric charts

Model metrics (2)

☰ accuracy

📄

⋮

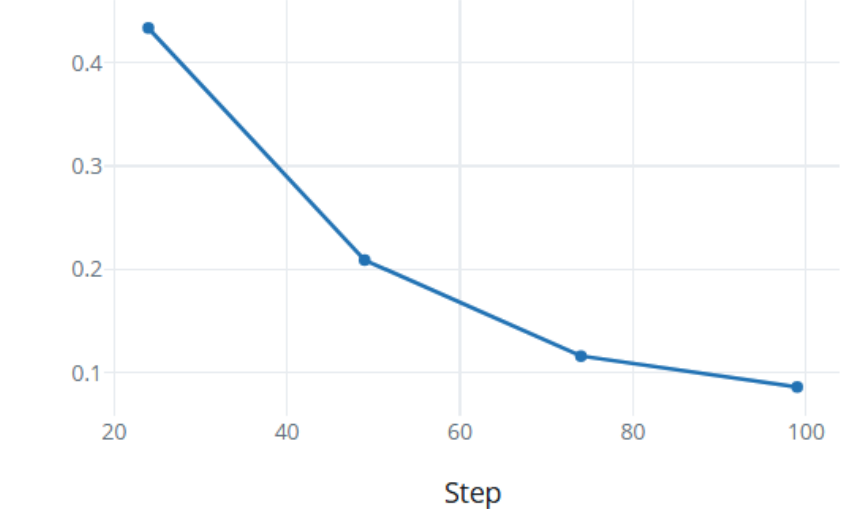


pytorch-nn-v1

☰ train_loss

📄

⋮



pytorch-nn-v1 (train_loss)

+ Add section

```
# เพิ่ม Description และ Tags ให้ PyTorch Model
client.update_registered_model(
    name=PYTORCH_MODEL_NAME,
    description="""
    🌸 Iris Flower Classification Model (PyTorch)

    Purpose: จำแนกประเภทดอกไอริส 3 ชนิด
    Architecture: Simple Neural Network (Input -> Hidden -> Output)
    Input: 4 features (float32)
    Output: 3 class logits
    """
)

client.set_registered_model_tag(PYTORCH_MODEL_NAME, "task", "classification")
client.set_registered_model_tag(PYTORCH_MODEL_NAME, "dataset", "iris")
client.set_registered_model_tag(PYTORCH_MODEL_NAME, "framework", "pytorch")

client.update_model_version(
    name=PYTORCH_MODEL_NAME,
    version="1",
    description="Baseline SimpleNN (hidden_size=16)."
)
client.set_model_version_tag(PYTORCH_MODEL_NAME, "1", "architecture", "SimpleNN")

print(f"✅ เพิ่ม Description และ Tags สำเร็จ!")
```

✅ เพิ่ม Description และ Tags สำเร็จ!

mlflow 3.8.1

+ New

Home

Experiments

Models

Prompts

Registered Models

Share and manage machine learning models. [Learn more](#)

Filter registered models by name or t...

🔍

Name ↕	Latest version	Aliased versions	Created by	Last modified	Tags
iris-classifier-pytorch	Version 1			01/28/2026, 09:13:02 ...	task: classificationdataset: irisfram

Create Model

mlflow 3.8.1

+ New

Home

Experiments

Models

Prompts

Registered Models >

iris-classifier-pytorch

Created Time: 01/28/2026, 09:12:19 AMLast Modified: 01/28/2026, 09:13:02 AM

Description

Edit

🌸 Iris Flower Classification Model (PyTorch)

Purpose: จำแนกประเภทดอกไอริส 3 ชนิด
Architecture: Simple Neural Network (Input -> Hidden -> Output)
Input: 4 features (float32)
Output: 3 class logits

Tags

Name	Value	Actions
dataset	iris	<div>✎🗑</div>
framework	pytorch	<div>✎🗑</div>
task	classification	<div>✎🗑</div>

Name

Value

Add

Versions

Compare

Version	Registered at ↕	Created by	Tags	Aliases	Description
📌 Version 1	01/28/2026, 09:12:19 AM		architecture: SimpleNN	<div>✎Add</div>	Baseline SimpleNN (hidden_siz...

New model registry UI

🔛

