Basic_mlflow

MACHINE LEARNING OPERATIONS

UNLIMITED

Presented by Asst. Prof. Dr. Tuchsanai Ploysuwan

How to install and set up MLflow

3. 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

```
┌─────────────────────────────────────────────────────────────────┐
│                        🏛️ MLflow Model Registry                 │
├─────────────────────────────────────────────────────────────────┤
│                                                                 │
│  ┌──────────────────────────────────────────────────────────┐   │
│  │  📦 Registered Model: "iris-classifier"                  │   │
│  │  ├── 📝 Description: โมเดลจำแนกดอกไอริส                  │   │
│  │  ├── 🏷️ Tags: [classification, sklearn, production-ready]  │
│  │  │                                                       │   │
│  │  ├── 📌 Version 1 ──────────────────────────────────────  │   │
│  │  │   ├── Stage: ⚫ Archived                              │   │
│  │  │   ├── Created: 2024-01-15 10:30:00                    │   │
│  │  │   ├── Accuracy: 0.92                                  │   │
│  │  │   ├── Algorithm: RandomForest(n_estimators=50)        │   │
│  │  │   └── Run ID: abc123...                               │   │
│  │  │                                                       │   │
│  │  ├── 📌 Version 2 ──────────────────────────────────────  │   │
│  │  │   ├── Stage: 🟡 Staging                               │   │
│  │  │   ├── Created: 2024-01-20 14:45:00                    │   │
│  │  │   ├── Accuracy: 0.95                                  │   │
│  │  │   ├── Algorithm: RandomForest(n_estimators=100)       │   │
│  │  │   └── Run ID: def456...                               │   │
│  │  │                                                       │   │
│  │  └── 📌 Version 3 ──────────────────────────────────────  │   │
│  │      ├── Stage: 🟢 Production  ← กำลังใช้งานจริง         │   │
│  │      ├── Created: 2024-01-25 09:00:00                    │   │
│  │      ├── Accuracy: 0.97                                  │   │
│  │      ├── Algorithm: GradientBoosting(n_estimators=100)   │   │
│  │      └── Run ID: ghi789...                               │   │
│  └──────────────────────────────────────────────────────────┘   │
│                                                                 │
└─────────────────────────────────────────────────────────────────┘
```

5. องค์ประกอบหลักของ Model Registry

5.1 Registered Model

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]
```

5.2 Model Version

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}
```

5.3 Model Artifacts

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

5.4 Metadata

Metadata คือข้อมูลที่อธิบาย Model:

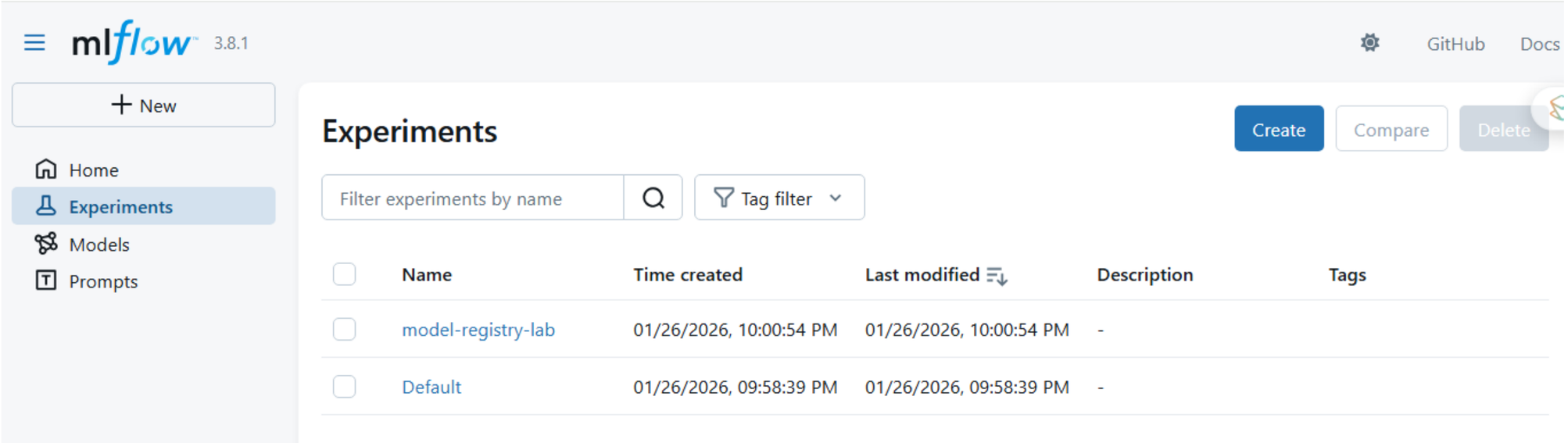
```yaml
# ตัวอย่าง Metadata
model_name: iris-classifier
version: 3
created_by: john.doe@company.com
created_at: 2024-01-25T09:00:00Z

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
```

Model Registry from Train

2.1 วิธีที่ 1: ลงทะเบียนพร้อม Train (Scikit-learn)

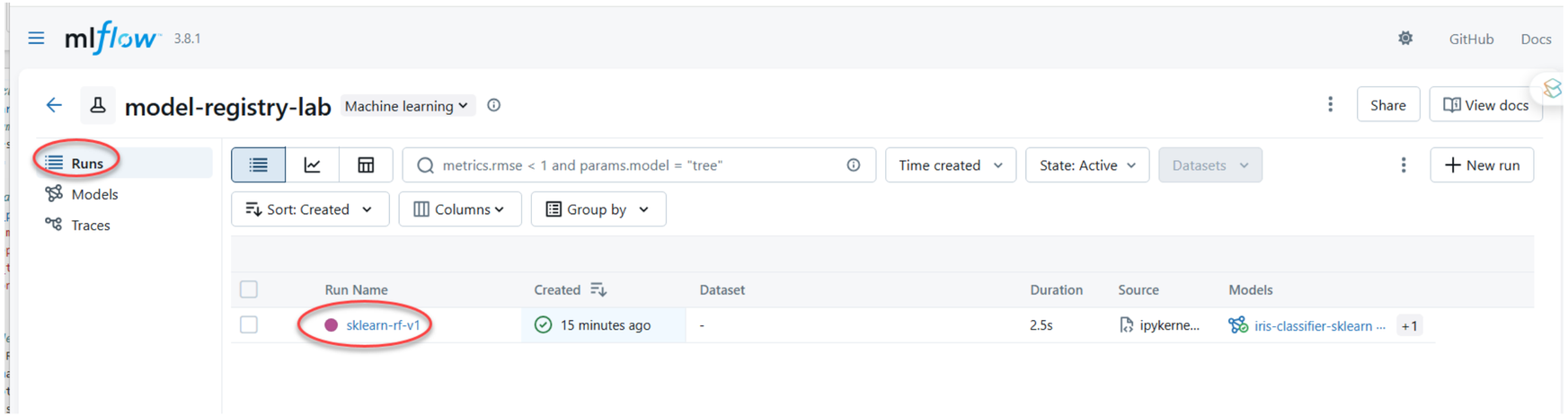
```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score
from mlflow.models import infer_signature
import numpy as np

# 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")
```

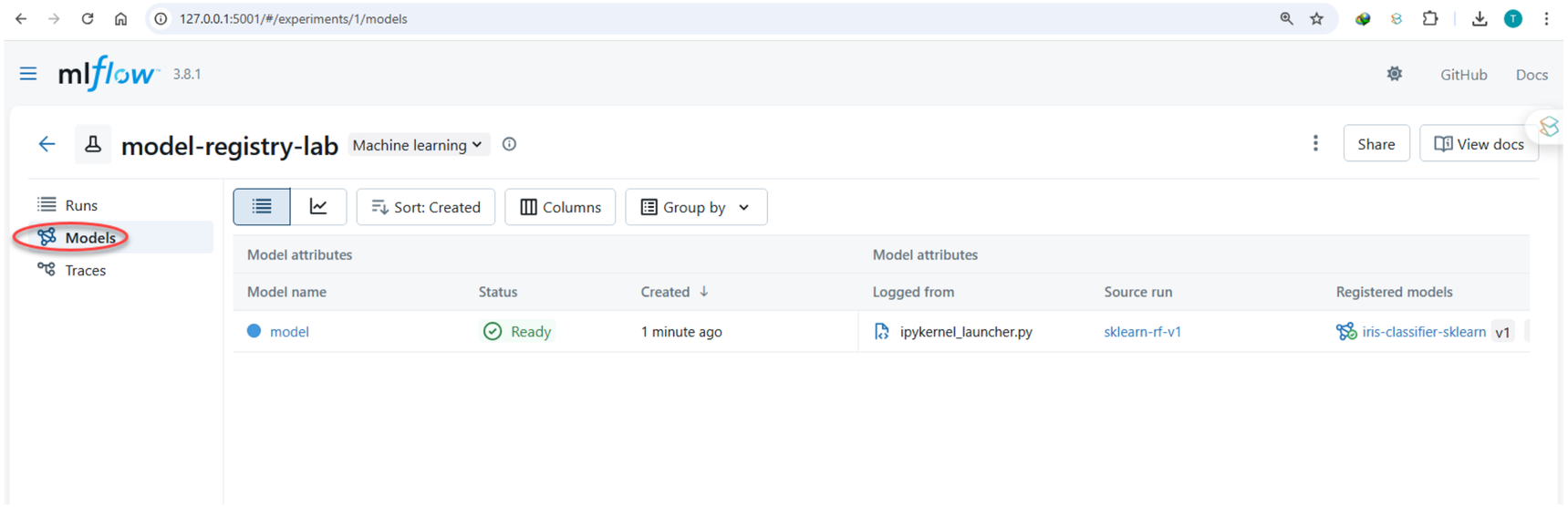
```
2026/01/26 15:00:54 INFO mlflow.tracking.fluent: Experiment with name 'model-registry-lab' does not exist. Creating a new experiment.
<Experiment: artifact_location='mlflow-artifacts:/1', creation_time=1769439654326, experiment_id='1', last_update_time=1769439654326, lifecycle_stage='active', name='model-registry-lab',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



Model Version 1

```python
# 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-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

mlflow 3.8.1 GitHub

model-registry-lab › Models ›

model

Register mode

Overview | Traces | Artifacts

Description

No description

Metrics (2)

Search metrics

Metric	Dataset	Source run	Value
accuracy	-	sklearn-rf-v1	1
f1_score	-	sklearn-rf-v1	1

Parameters (4)

Search parameters

Parameter	Value
max_depth	5
model_type	RandomForestClassifier
n_estimators	50
version_note	Baseline model

Runs

Search runs

Run	Input
sklearn-rf-v1	-

About this logged model

Created at	2 minutes ago
Created by	root
Status	Ready
Model ID	m-832336883e82413da0bb...
Source run	sklearn-rf-v1
Source run ID	f74a11d547014a3e9a81b75...
Logged from	ipykernel_launcher.py

Datasets used

None

Model versions

iris-classifier-sklearn v1 +1

```python
# 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!")
```

mlflow 3.8.1

GitHub Do

+ 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	

Name Value Add

Versions

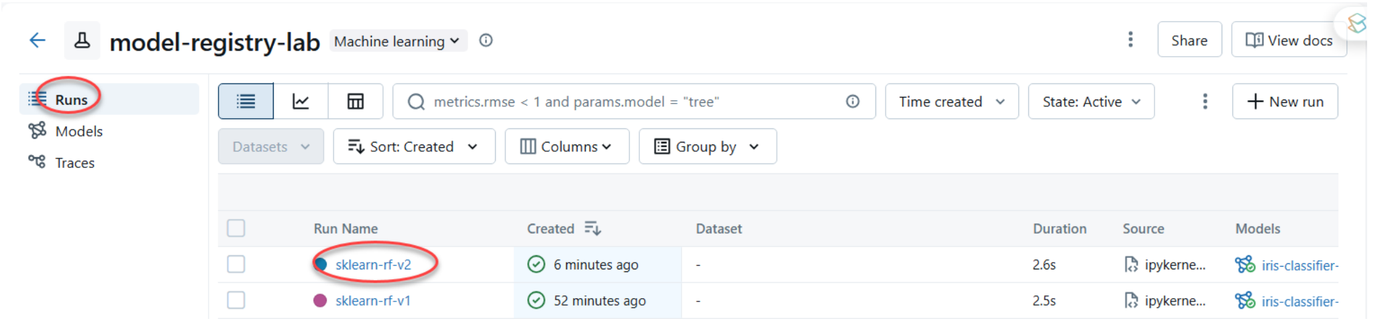
Compare

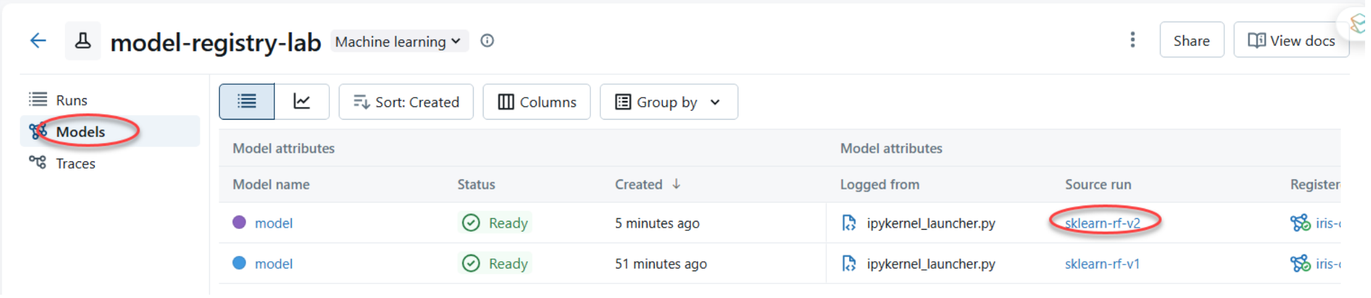
New model registry UI

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

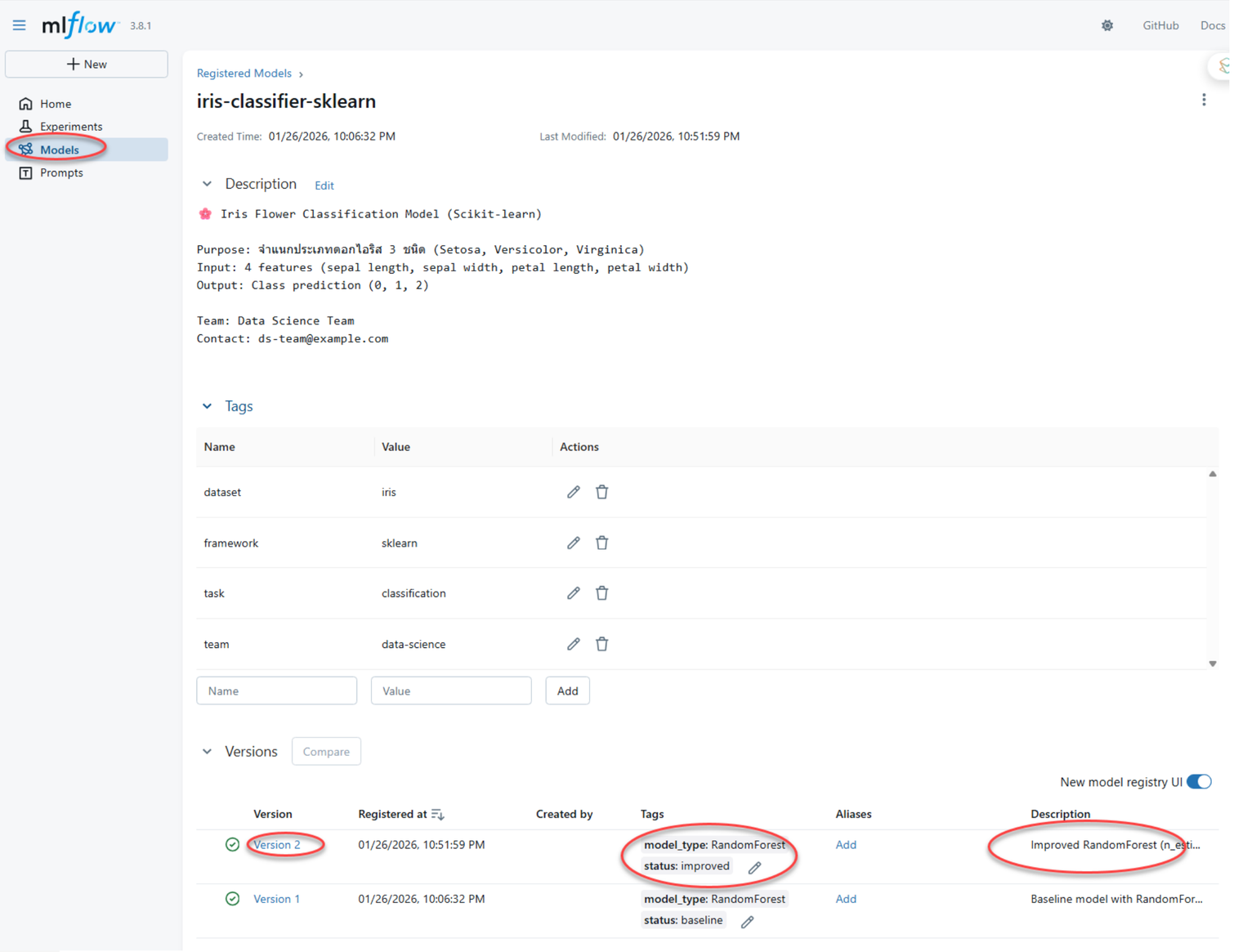
```python
# Train และลงทะเบียน Model Version 2 (RandomForest n=100)
with mlflow.start_run(run_name="sklearn-rf-v2"):

    # 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 Version 2





```python
# 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}")
```




```python
# 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!
```

2.2 วิธีที่ 2: ลงทะเบียนภายหลังจาก Run ที่มีอยู่แล้ว

ใช้เมื่อต้องการ:

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

```python
from sklearn.ensemble import GradientBoostingClassifier

# 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}")
```

```
2026/01/26 16:11:17 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: 3c697061521342e993e7ad99bbe7e25f
🏃 View run sklearn-gb-candidate at: http://127.0.0.1:5000/#/experiments/1/runs/3c697061521342e993e7ad99bbe7e25f
🧪 View experiment at: http://127.0.0.1:5000/#/experiments/1
```

```python
# ตัดสินใจลงทะเบียน Model ภายหลัง
# สมมติว่าตรวจสอบแล้วพบว่า Model นี้ดีที่สุด

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!")
```

```
Registered model 'iris-classifier-sklearn' already exists. Creating a new version of this model...
2026/01/26 16:12:25 WARNING mlflow.tracking._model_registry.fluent: Run with id 3c697061521342e993e7ad99bbe7e25f has no artifacts at artifact path 'model', registering mo
del based on models:/m-b7025e9dfd3649f6a0bb1abddfa6c102 instead
2026/01/26 16:12:25 INFO mlflow.store.model_registry.abstract_store: Waiting up to 300 seconds for model version to finish creation. Model name: iris-classifier-sklearn,
version 3
Created version '3' of model 'iris-classifier-sklearn'.
✅ ลงทะเบียน Model ภายหลังสำเร็จ!
📦 Model Name: iris-classifier-sklearn
📌 Version: 3
✅ เพิ่ม Description และ Tags สำหรับ Version 3 สำเร็จ!
```

mlflow 3.8.1

GitHub Doc

+ New

- Home
- Experiments
- Models
- Prompts

Settings

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	

Name 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	Add	GradientBoosting model. Best ...
	Version 2	01/26/2026, 10:51:59 PM		model_type: RandomForest status: improved	Add	Improved RandomForest (n_esti...
	Version 1	01/26/2026, 10:06:32 PM		model_type: RandomForest status: baseline	Add	Baseline model with RandomFor...

📚 ส่วนที่ 3: การดูข้อมูล Registered Models

แนวคิด

หลังจากลงทะเบียน Model แล้ว สามารถดูข้อมูลได้หลายวิธี

ฟังก์ชันสำคัญ

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

[9]:
```
# 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 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-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

   📅 Created: 1769439992111
   📅 Last Updated: 1769443945598
```

```python
# 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
```

```python
# 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]:
```python
# 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
```

🏷️ ส่วนที่ 4: การจัดการ Model Stages (Aliases)

แนวคิด (MLflow 2.x)

ใน MLflow 2.x แนะนำให้ใช้ **Model Aliases** แทน Stages เดิม

Concept เก่า (Stages)	Concept ใหม่ (Aliases)
`None` , `Staging` , `Production` , `Archived`	กำหนดเองได้ เช่น `champion` , `challenger`
เปลี่ยน Stage ด้วย `transition_model_version_stage()`	กำหนด Alias ด้วย `set_registered_model_alias()`

ฟังก์ชันสำคัญ (MLflow 2.x - Aliases)

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

4.1 การใช้ Model Aliases (แนะนำสำหรับ MLflow 2.x)

[13]:
```
# 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
```

mlflow 3.8.1

GitHub Docs

+ 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	

Name 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...

```python
# 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: 3c697061521342e993e7ad99bbe7e25f
```

```python
# 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'}
```

📦 ส่วนที่ 5: การโหลด Model จาก Registry

แนวคิด

การโหลด Model จาก Registry สามารถทำได้โดยการโหลดจาก **ARTIFACTS_BASE** โดยตรง ซึ่งเป็นวิธีที่เร็วที่สุดสำหรับ Production บน Server เดียวกัน

โครงสร้างไฟล์ใน ARTIFACTS_BASE

เมื่อใช้ `--serve-artifacts` กับ MLflow Server, **Models จะถูกเก็บในโฟลเดอร์ `models/` แยกต่างหาก**:

```
mlartifacts/
└── <experiment_id>/
    ├── <run_id>/
    │   └── artifacts/          # Artifacts ทั่วไป (plots, data, config)
    └── models/                 # ⚠️ Models ถูกเก็บที่นี่!
        └── m-<model_id>/
            └── artifacts/
                ├── MLmodel
                ├── model.pkl
                └── ...
```

```python
import yaml

# กำหนด ARTIFACTS_BASE path ตาม MLflow Server configuration
ARTIFACTS_BASE = "/home/student/workspace/mlflowserver-lab/mlartifacts"

# ดึงข้อมูล Experiment ID
experiment = mlflow.get_experiment_by_name("model-registry-lab")
experiment_id = experiment.experiment_id

print(f"📁 ARTIFACTS_BASE: {ARTIFACTS_BASE}")
print(f"🆔 Experiment ID: {experiment_id}")
```

```
📁 ARTIFACTS_BASE: /home/student/workspace/mlflowserver-lab/mlartifacts
🆔 Experiment ID: 1
```

```python
# Helper Function: ค้นหา Model ตาม flavor (sklearn, pytorch, etc.)

def find_model_path_by_flavor(artifacts_base: str, experiment_id: str, flavor: str = "sklearn") -> str:
    """
    ค้นหา model path จาก models/ folder ตาม flavor ที่ต้องการ
    (MLflow เก็บ models แยกใน models/ folder เมื่อใช้ --serve-artifacts)

    Args:
        artifacts_base: Base path ของ artifacts
        experiment_id: Experiment ID
        flavor: MLflow flavor ที่ต้องการ ("sklearn", "pytorch", "tensorflow", etc.)

    Returns:
        Full path ไปยัง model หรือ None ถ้าไม่พบ
    """
    models_folder = f"{artifacts_base}/{experiment_id}/models"

    if not os.path.exists(models_folder):
        print(f"⚠️ ไม่พบโฟลเดอร์: {models_folder}")
        return None

    # ค้นหา model folder ที่มี flavor ตรงกับที่ต้องการ
    for model_dir in os.listdir(models_folder):
        model_path = f"{models_folder}/{model_dir}/artifacts"
        mlmodel_file = f"{model_path}/MLmodel"

        if os.path.exists(mlmodel_file):
            # อ่าน MLmodel file เพื่อตรวจสอบ flavor
            with open(mlmodel_file, 'r') as f:
                mlmodel = yaml.safe_load(f)

            # ตรวจสอบว่ามี flavor ที่ต้องการหรือไม่
            if 'flavors' in mlmodel and flavor in mlmodel['flavors']:
                return model_path

    return None
```

```python
# ค้นหาและโหลด sklearn model จาก ARTIFACTS_BASE
print("📥 โหลด Scikit-learn Model จาก ARTIFACTS_BASE:")
print("-" * 50)

sklearn_model_path = find_model_path_by_flavor(ARTIFACTS_BASE, experiment_id, flavor="sklearn")

if sklearn_model_path:
    print(f"📦 Model Path: {sklearn_model_path}")

    # โหลด model
    loaded_sklearn_model = mlflow.sklearn.load_model(sklearn_model_path)
    print(f"✅ โหลด Model สำเร็จ: {type(loaded_sklearn_model)}")

    # ทดสอบทำนาย
    predictions = loaded_sklearn_model.predict(X_test[:5])
    print(f"\n🔮 ทดสอบทำนาย 5 ตัวอย่างแรก:")
    print(f"   Predictions: {predictions}")
    print(f"   Actual:      {y_test[:5]}")
else:
    print("⚠️ ไม่พบ sklearn model ใน ARTIFACTS_BASE")
    print("💡 ตรวจสอบว่า ARTIFACTS_BASE path ถูกต้อง")
```

```
📥 โหลด Scikit-learn Model จาก ARTIFACTS_BASE:
--------------------------------------------------
📦 Model Path: /home/student/workspace/mlflowserver-lab/mlartifacts/1/models/m-832336883e82413da0bbb006a0868e80/artifacts
✅ โหลด Model สำเร็จ: <class 'sklearn.ensemble._forest.RandomForestClassifier'>

🔮 ทดสอบทำนาย 5 ตัวอย่างแรก:
   Predictions: [1 0 2 1 1]
   Actual:      [1 0 2 1 1]
```