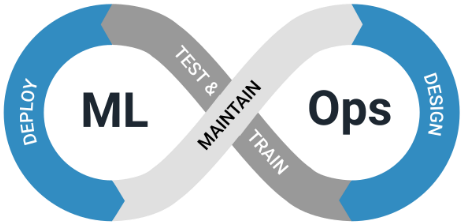
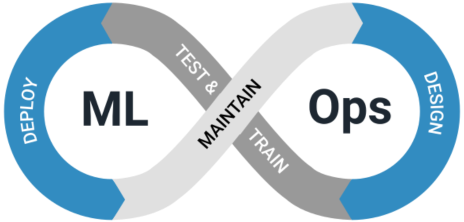


MLOPS

Machine Learning Operations

mlflow | Airflow | mlflow | Grafana

Video 1 Experiment/Tracking → Video 2 Orchestration → Video 3 Deployment → Video 4 Monitoring



MLOPS

Machine Learning Operations

mlflow | Airflow | mlflow | Grafana

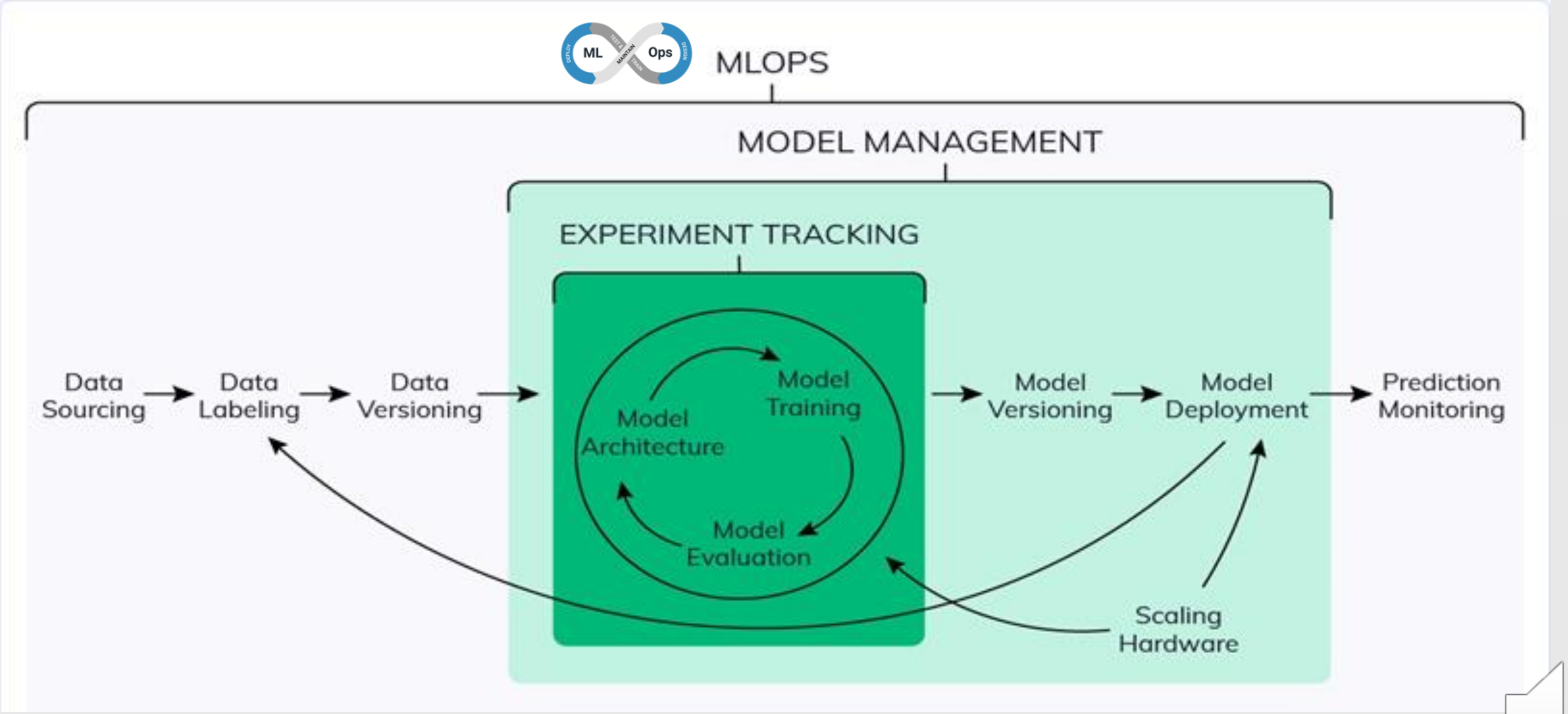
Video 1	Video 2	Video 3	Video 4
Experiment/Tracking	Orchestration	Deployment	Monitoring

MLOPS TARGET

1. Unifies ML development & operations - Ensures reliable, scalable, and reproducible model lifecycle

2. Key Focus:
    a. Automation of training, deployment, monitoring
    b. CI/CD for ML pipelines
    c. Governance & Compliance (versioning, auditing)

MACHINE LEARNING LIFECYCLE



https://neptune.ai/blog/mlops

MLOPS MATURITY MODEL FROM MICROSOFT

Level 0: No MLOPS	• Manual, ad-hoc training & deployment • Teams siloed, no versioning or monitoring
Level 1: DevOps—but no MLOPS	• Basic CI/CD for apps • Models still trained & deployed manually
Level 2: Automated Training	• Reproducible experiments • Automated pipelines for data prep & training
Level 3: Automated Model Deployment	• CI/CD for models • Testing, monitoring & A/B deployments
Level 4: Full MLOPS Automated Operations	• End-to-end automation (train → deploy → monitor → retrain) • Continuous feedback loop, reliable & scalable ML in production

https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/mlops-maturity-model mlflow™

STREAMLINING THE MACHINE LEARNING LIFECYCLE

WHAT IS mlflow™ ?

1. Open-source platform for managing the ML lifecycle.
2. Facilitates experiment tracking, model versioning, and deployment.
3. Supports multiple ML frameworks and libraries.

https://mlflow.org/docs/latest/ml/

mlflow · ML Docs · API Reference · Community · Version: 3.4.0rc0 (latest)

- MLflow
- MLflow 3.0
- Getting Started
- Machine Learning
  - Traditional ML
  - Deep Learning
- Build
  - MLflow Tracking
  - MLflow Model
  - MLflow Datasets
- Evaluate
- Deploy
- Webhooks
- Team Collaboration
- API References
- More

MLflow

MLflow: A Tool for Managing the Machine Learning Lifecycle

MLflow is an open-source platform, purpose-built to assist machine learning practitioners and teams in handling the complexities of the machine learning process. MLflow focuses on the full lifecycle for machine learning projects, ensuring that each phase is manageable, traceable, and reproducible.

MLflow Getting Started Resources

If this is your first time exploring MLflow, the tutorials and guides here are a great place to start. The emphasis in each of these is getting you up to speed as quickly as possible with the basic functionality, terms, APIs, and general best practices of using MLflow in order to enhance your learning in area-specific guides and tutorials.

Learn about MLflow | MLflow Basics | MLflow Models Introduction | Traditional ML | Deep Learning

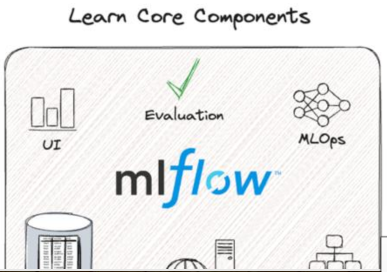
Learn about the core components of MLflow

Quickstarts

Get Started with MLflow in our 5-minute tutorial

Guides

Learn the core components of MLflow with this in-depth guide to Tracking



Search (Cmd + Ctrl + U)

Record Comments Share

7:14

7

mlflow.org/docs/latest/ml/

mlflow ML Docs API Reference Community Ver

- MLflow
- MLflow 3.0
- Getting Started
- Machine Learning
- Build
- Evaluate
- Deploy
- Webhooks
- Team Collaboration
- API References
- More

MLflow

MLflow: A Tool for Managing the Machine Learning Lifecycle

MLflow is an open-source platform, purpose-built to assist machine learning practitioners and teams in h complexities of the machine learning process. MLflow focuses on the full lifecycle for machine learning p each phase is manageable, traceable, and reproducible.

MLflow Getting Started Resources

If this is your first time exploring MLflow, the tutorials and guides here are a great place to start. The emp these is getting you up to speed as quickly as possible with the basic functionality, terms, APIs, and gene using MLflow in order to enhance your learning in area-specific guides and tutorials.

Learn about MLflow MLflow Basics MLflow Models Introduction Traditional ML

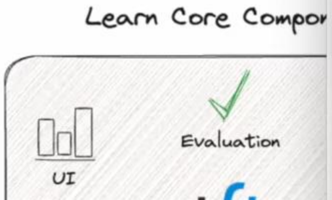
Learn about the core components of MLflow

Quickstarts

Get Started with MLflow in our 5-minute tutorial

Guides

Learn the core components of MLflow with this in-depth guide to Tracking



mlflow POPULAR COMPONENTS

1. Tracking	2. Models	3. Model Registry
- Log and query experiments. - Record parameters, metrics, and artifacts. - Compare and visualize results.	- Log and store models in various formats. - Serve models for inference.	- Centralized store for managing model versions. - Track model lineage and stages (e.g., staging, production).

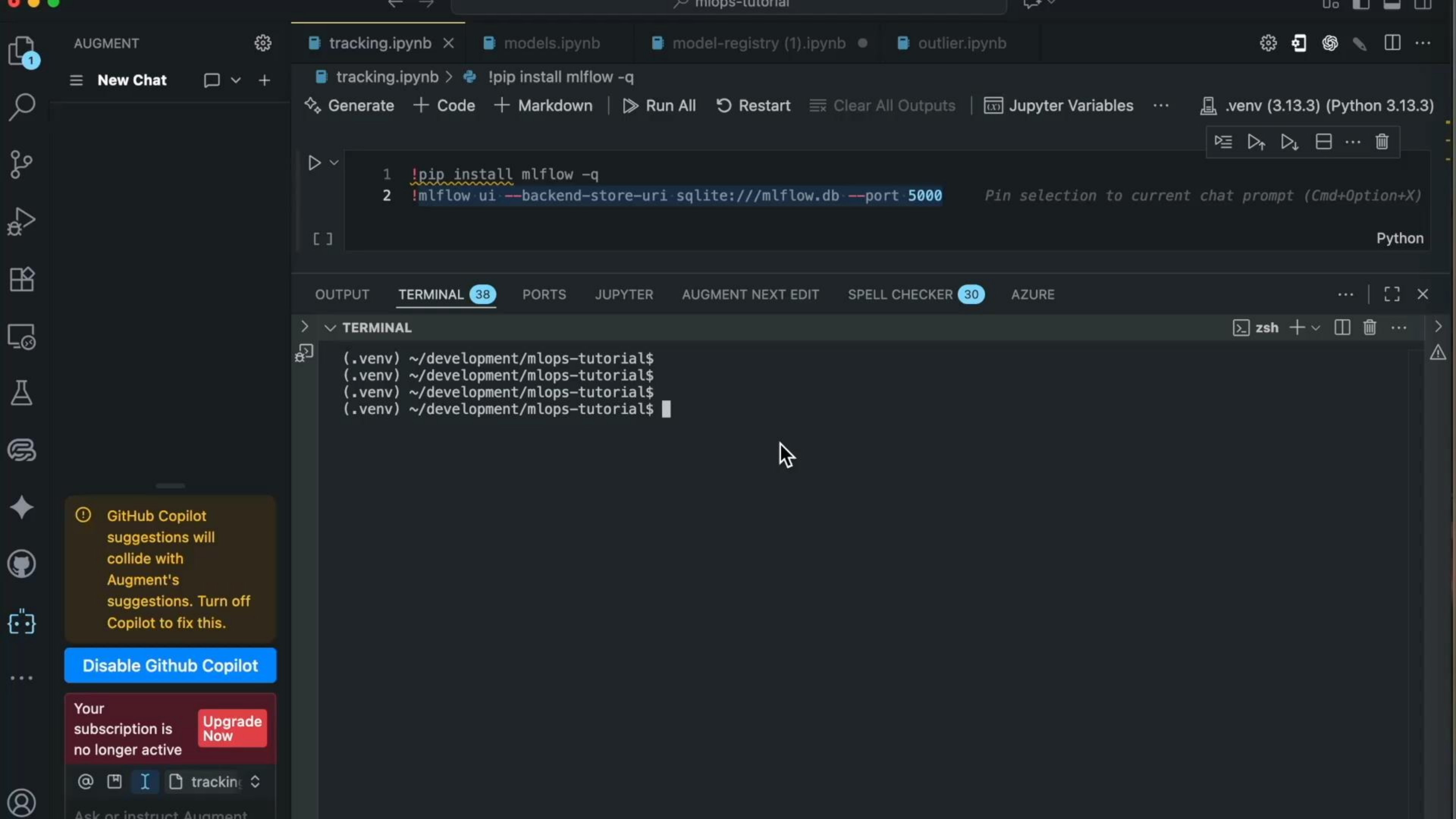
https://mlflow.org/docs/latest/ml/

GETTING STARTED WITH mlflow

1. Installation:
   pip install mlflow

2. Run on local system:
   mlflow ui –backend-store-uri sqlite:///mlflow.db –port 5000

3. Run on server:
   mlflow server -h 0.0.0.0 -p 5000 --backend-store-uri postgresql://DB_USER:DB_PASSWORD@DB_ENDPOINT:5432/DB_NAME --default-artifact-root s3://S3_BUCKET_NAME


mlops-tutorial
AUGMENT
New Chat
tracking.ipynb
models.ipynb
model-registry (1).ipynb
outlier.ipynb
tracking.ipynb > !pip install mlflow -q
Generate
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
.venv (3.13.3) (Python 3.13.3)
1 !pip install mlflow -q
2 !mlflow ui --backend-store-uri sqlite:///mlflow.db --port 5000
Pin selection to current chat prompt (Cmd+Option+X)
Python
OUTPUT
TERMINAL 38
PORTS
JUPYTER
AUGMENT NEXT EDIT
SPELL CHECKER 30
AZURE
TERMINAL
zsh
(.venv) ~/development/mlops-tutorial$
(.venv) ~/development/mlops-tutorial$
(.venv) ~/development/mlops-tutorial$
(.venv) ~/development/mlops-tutorial$
GitHub Copilot suggestions will collide with Augment's suggestions. Turn off Copilot to fix this.
Disable Github Copilot
Your subscription is no longer active
Upgrade Now
tracking
Ask or instruct Augment

1. MLFLOW TRACKING

- **Purpose**:
  - Monitor and compare machine learning experiments.
- **Features**:
  - Log parameters, metrics, and artifacts.
  - Organize experiments into runs.
  - Visualize performance over time.

BASIC FUNCTIONS IN PYTHON PACKAGE

1. Import the package: ***import mlflow***
2. Set mlflow server uri: ***mlflow.set_tracking_uri()***
3. Set experiment unique name: ***mlflow.set_experiment()***
4. Set the scope of MLflow tracking operation: ***with mlflow.start_run()***
    a. Set tag (arbitrary twisted variable- key-value pair): ***mlflow.set_tag()***
    b. Set training and model parameters (epoch, …) : ***mlflow.log_param()***
    c. Set a group of training and model parameters (epoch, …): ***mlflow.log_params()***
    d. Log metrics name and value: ***mlflow.log_metric()***
    e. Save artifact (model files, dataset, …): ***mlflow.log_artifact()***

tracking.ipynb · models.ipynb · model-registry (1).ipynb · outlier.ipynb

EXPLORER

MLOPS-TU...

- .venv
- 1-mlflow-tracking
- mlartifacts
- mlruns
- videos
- data.xlsx
- housing-model.pkl
- mlflow.db
- models.ipynb
- models.py
- outlier.ipynb
- outlier.py
- tracking.ipynb
- tracking.py

OUTLINE

No symbols found in document 'tracking.ipynb'

tracking.ipynb > from sklearn.linear_model import LinearRegression

Generate | + Code | + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | ··· | .venv (3.13.3) (Python 3.13.3)

```python
!pip install mlflow -q
!mlflow ui --backend-store-uri sqlite:///mlflow.db --port 5000
```

Pin selection to current chat prompt (Cmd+Option+X)

[ ] Python

OUTPUT | TERMINAL 53 | PORTS | JUPYTER | AUGMENT NEXT EDIT | SPELL CHECKER 45 | AZURE

TERMINAL

```
INFO:     127.0.0.1:54868 - "GET /api/2.0/mlflow/runs/get?run_uuid=7422ceeb71bb486da76abb01d7d633a9&run_id=7422ceeb71bb486da76abb01d7d633a9 HTTP/1.1" 200 OK
INFO:     127.0.0.1:54868 - "PUT /api/2.0/mlflow-artifacts/artifacts/5/7422ceeb71bb486da76abb01d7d633a9/artifacts/california-housing/housing-model.pkl HTTP/1.1" 200 OK
INFO:     127.0.0.1:54868 - "GET /api/2.0/mlflow/runs/get?run_uuid=7422ceeb71bb486da76abb01d7d633a9&run_id=7422ceeb71bb486da76abb01d7d633a9 HTTP/1.1" 200 OK
INFO:     127.0.0.1:54868 - "POST /api/2.0/mlflow/runs/update HTTP/1.1" 200 OK
INFO:     127.0.0.1:54882 - "GET /ajax-api/2.0/mlflow/experiments/search?max_results=25&order_by=last_update_time+DESC HTTP/1.1" 200 OK
INFO:     127.0.0.1:54882 - "GET /ajax-api/2.0/mlflow/experiments/search?max_results=25&order_by=last_update_time+DESC HTTP/1.1" 200 OK
INFO:     127.0.0.1:54893 - "GET /ajax-api/2.0/mlflow/experiments/get?experiment_id=5 HTTP/1.1" 200 OK
INFO:     127.0.0.1:54895 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54894 - "POST /ajax-api/2.0/mlflow/experiments/search-datasets HTTP/1.1" 200 OK
INFO:     127.0.0.1:54894 - "GET /ajax-api/2.0/mlflow/gateway-proxy?gateway_path=api%2F2.0%2Fendpoints%2F HTTP/1.1" 200 OK
INFO:     127.0.0.1:54895 - "POST /ajax-api/2.0/mlflow/logged-models/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54894 - "GET /ajax-api/2.0/mlflow/model-versions/search?filter=run_id%3D%277422ceeb71bb486da76abb01d7d633a9%27 HTTP/1.1" 200 OK
INFO:     127.0.0.1:54895 - "GET /ajax-api/2.0/mlflow/model-versions/search?filter=tags.%60mlflow.prompt.is_prompt%60+%3D+%27true%27+AND+tags.%60mlflow.prompt.run_ids%60+ILIKE+%22%257422ceeb71bb486da76abb01d7d633a9%25%22 HTTP/1.1" 200 OK
INFO:     127.0.0.1:55589 - "GET /ajax-api/2.0/mlflow/experiments/search?max_results=25&order_by=last_update_time+DESC HTTP/1.1" 200 OK
INFO:     127.0.0.1:55608 - "POST /ajax-api/2.0/mlflow/experiments/delete HTTP/1.1" 200 OK
INFO:     127.0.0.1:55607 - "POST /ajax-api/2.0/mlflow/experiments/delete HTTP/1.1" 200 OK
INFO:     127.0.0.1:55607 - "GET /ajax-api/2.0/mlflow/experiments/search?max_results=25&order_by=last_update_time+DESC HTTP/1.1" 200 OK
INFO:     127.0.0.1:55607 - "GET /ajax-api/2.0/mlflow/registered-models/search?filter=&max_results=25&order_by=name+ASC HTTP/1.1" 200 OK
INFO:     127.0.0.1:55607 - "GET /ajax-api/2.0/mlflow/registered-models/search?filter=tags.%60mlflow.prompt.is_prompt%60+%3D+%
```

RUN SEARCH

```
from mlflow.tracking import MlflowClient
from mlflow.entities import ViewType
mlflow_tracking_uri = “http://127.0.0.1:5000/”
client = MlflowClient(tracking_uri= mlflow_tracking_uri)
runs = client.search_runs(
    experiment_ids='3',
    filter_string="metrics.mean_squared_error < 0.6",
    run_view_type=ViewType.ACTIVE_ONLY,
    max_results=5,
    order_by=["metrics.mean_squared_error ASC"]
)
for run in runs:
    print(f"run id: {run.info.run_id}, rmse: {run.data.metrics['mean_squared_error']:.4f}")
```

AUGMENT

New Chat

tracking.ipynb | models.ipynb | model-registry (1).ipynb | outlier.ipynb

tracking.ipynb > 3.2. Model Search > from mlflow.tracking import MlflowClient

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables ··· .venv (3.13.3) (Python 3.13.3)

3.2. Model Search

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
from mlflow.tracking import MlflowClient
from mlflow.entities import ViewType
client = MlflowClient(tracking_uri="http://127.0.0.1:5000/")

runs = client.search_runs(
    experiment_ids="8",
    filter_string="metrics.r2_score < 0.7",
    run_view_type=ViewType.ACTIVE_ONLY,
    max_results=5,
    order_by=["metrics.r2_score ASC"]
)
for run in runs:
    print(f"experiment id: 8, run id: {run.info.run_id}, r2_score: {run.data.metrics['r2_score']:.4f}")
```

[10] ✓ 0.0s Python

GitHub Copilot suggestions will collide with Augment's suggestions. Turn off Copilot to fix this.

Disable Github Copilot

Your subscription is no longer active | Upgrade Now

OUTPUT | TERMINAL 28 | JUPYTER | AUGMENT NEXT EDIT | SPELL CHECKER 20 | AZURE

TERMINAL

```
INFO:     127.0.0.1:54522 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54533 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54548 - "POST /api/2.0/mlflow/runs/search HTTP/1.1" 400 Bad Request
INFO:     127.0.0.1:54548 - "POST /api/2.0/mlflow/runs/search HTTP/1.1" 400 Bad Request
INFO:     127.0.0.1:54562 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54562 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54562 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54579 - "POST /api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:54609 - "POST /api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
```

2. MLFLOW MODELS

- **Purpose**:
  - Manage and serve machine learning models.
- **Features**:
  - Log models in different formats (e.g., TensorFlow, PyTorch).
  - Serve models via REST APIs.
  - Deploy models to various platforms.

MLFLOW MODELS WITH ML PYTHON PACKAGES

1. Logging models:

   mlflow.<framework>.log_model(our_model, name=”mlflow_model_path”)

2. Loading models:

   model_uri = ‘models:/{model_id}’
   Load_model = mlflow.pyfunc.load_model(model_uri)

MLflow Models

MLFLOW MODELS WITH SKLEARN (EXAMPLE)

1. mlflow.sklearn.log_model(model, name="housing_linear_regression_2")
   Mlflow_model_path
   a. MLmodel
   b. conda .yaml
   c. model.pkl
   d. python_env.yaml
   e. requirements.txt
2. modelid = "m-b0af883a2203425789159bba8937beef"
3. model_uri = 'models:/{}'.format(modelid)
4. load_model = mlflow.pyfunc.load_model(model_uri)

arch (Cmd + Ctrl + U)

Record | Comments | Share

5 MB
ytes
2 MB
3 MB
2 MB
ytes

19

tracking.ipynb | models.ipynb | model-registry (1).ipynb | outlier.ipynb

tracking.ipynb > from sklearn.linear_model import LinearRegression

Generate | + Code | + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline

```python
                                                  df['median_house_value'], test_size=0.3, random_state
    model = LinearRegression()
    model.fit(xtrain, ytrain)
    pred = model.predict(xtest)
    mse = mean_squared_error(ytest, pred)
    mae = mean_absolute_error(ytest, pred)
    r2 = r2_score(ytest, pred)
    with open("housing-model.pkl", "wb") as f:
            pickle.dump(model, f)

    mlflow.log_metric("mean_squared_error",  mse)
    mlflow.log_metric("mean_absolute_error",  mae)
    mlflow.log_metric("r2_score",  r2)
    mlflow.log_artifact(local_path="housing-model.pkl", artifact_path="california-housing")
```

[5] ✓ 0.3s

```
2025/09/15 15:02:07 INFO mlflow.tracking.fluent: Experiment with name 'california-housing-exp110' does not exist
🏃 View run rare-fly-738 at: http://127.0.0.1:5000/#/experiments/7/runs/2b2a18fa15cc458aadc59fdb53aeb506
🧪 View experiment at: http://127.0.0.1:5000/#/experiments/7
```

OUTPUT | TERMINAL 53 | JUPYTER | AUGMENT NEXT EDIT | SPELL CHECKER 45 | AZURE

TERMINAL

```
INFO:     127.0.0.1:56220 - "POST /ajax-api/2.0/mlflow/logged-models/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:56232 - "GET /ajax-api/2.0/mlflow/model-versions/search?filter=tags.%60mlflow.prompt.is_prom
```

3. MLFLOW MODEL REGISTRY

- **Purpose**:
  - Centralized management of model lifecycle.
- **Features**:
  - Version control for models.
  - Track model lineage and metadata.
  - Transition models between stages (e.g., from development to production)

PROBLEM: WITHOUT A MODEL REGISTRY

1. Deploying new ML models often requires messy back-and-forth emails.
2. Hard to track:
    a. What changed between versions
    b. Preprocessing steps, dependencies, hyperparameters
    c. Rollback to previous models during incidents
3. Leads to inefficiency & risk of losing reproducibility.

MODEL REGISTRY WORKFLOW

1. Experiment Tracking: log parameters, metrics, artifacts, models.
2. Register Models: select best performing ones.
3. Assign Stages:
    a. *Staging*: under review/testing.
    b. *Production*: live model.
    c. *Archived*: deprecated models (can rollback).
4. Model Metadata: size, training time, error metrics, dependencies.

MODELS REGISTRY WITH PYTHON PACKAGES

1. model_id = "m-609655c524b342f4ae1fe947af81945e"
2. model_name = "test7"
3. model_uri = f"models:/{model_id}"
4. **mlflow.register_model**(model_uri=model_uri, name=model_name)

3- MLflow Model Registry

3.1. Model Registry

```python
import mlflow
mlflow.set_tracking_uri("http://127.0.0.1:5000/")
model_id = "m-4fa61ed2ab9243e2b39f8dcb0a4508d0" # Find this model_id in your mlflow ui first and replace it
model_name = "housing_linear_regression_3"
model_uri = f"models:/{model_id}"
mlflow.register_model(model_uri=model_uri, name=model_name)
```

```
Successfully registered model 'housing_linear_regression_3'.
2025/09/16 11:10:31 INFO mlflow.store.model_registry.abstract_store: Waiting up to 300 seconds for model version to finish cre
Created version '1' of model 'housing_linear_regression_3'.
```

TERMINAL

```
requirements.txt HTTP/1.1" 200 OK
INFO:     127.0.0.1:51956 - "PUT /api/2.0/mlflow-artifacts/artifacts/3/models/m-abc9b2927a53483f8f1ba1c2c37b187a/artifacts/MLmodel HTTP/1.1" 200 OK
INFO:     127.0.0.1:51956 - "PUT /api/2.0/mlflow-artifacts/artifacts/3/models/m-abc9b2927a53483f8f1ba1c2c37b187a/artifacts/model.pkl HTTP/1.1" 200 OK
INFO:     127.0.0.1:51956 - "PUT /api/2.0/mlflow-artifacts/artifacts/3/models/m-abc9b2927a53483f8f1ba1c2c37b187a/artifacts/conda.yaml HTTP/1.1" 200 OK
INFO:     127.0.0.1:51956 - "PATCH /api/2.0/mlflow/logged-models/m-abc9b2927a53483f8f1ba1c2c37b187a HTTP/1.1" 200 OK
```

VERSION SEARCH

1. from mlflow.tracking import MlflowClient
2. model_name = "housing_linear_regression_400”
3. client = MlflowClient(tracking_uri="http://127.0.0.1:5000/")

4. latest_versions = **client.get_latest_versions**(name=model_name)
5. for version in latest_versions:
6.   print(f"version: {version.version}, stage: {version.current_stage}")

tracking.ipynb | models.ipynb | model-registry (1).ipynb | outlier.ipynb

Users > reza > Downloads > model-registry (1).ipynb > 3.2. Version Search > from mlflow.tracking import MlflowClient

Generate | + Code | + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | ··· | .venv (3.13.3) (Py

3.2. Version Search

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
from mlflow.tracking import MlflowClient
model_name = "housing_linear_regression_400"
client = MlflowClient(tracking_uri="http://127.0.0.1:5000/")

latest_versions = client.get_latest_versions(name=model_name)

for version in latest_versions:
    print(f"version: {version.version}, stage: {version.current_stage}")
```

[5] ✓ 0.0s

```
version: 1, stage: None
/var/folders/dx/hkx51pqs1vxc2zsq74nxsjp80000gn/T/ipykernel_2799/3548075878.py:5: FutureWarning: ``mlflow.tracking.cl
  latest_versions = client.get_latest_versions(name=model_name)
```

3.3. Stage Transition

OUTPUT | TERMINAL 43 | JUPYTER | AUGMENT NEXT EDIT | SPELL CHECKER 35 | AZURE

TERMINAL

```
INFO:     127.0.0.1:55908 - "POST /api/2.0/mlflow/registered-models/get-latest-versions HTTP/1.1" 200 OK
INFO:     127.0.0.1:55946 - "GET /ajax-api/2.0/mlflow/experiments/search?max_results=25&order_by=last_update_time+DE
/1.1" 200 OK
INFO:     127.0.0.1:55973 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:55973 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:56044 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:56044 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:56083 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:56083 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
```

AUGMENT

New Chat

GitHub Copilot suggestions will collide with Augment's suggestions. Turn off Copilot to fix this.

Disable Github Copilot

Your subscription is no longer active | Upgrade Now

Ask or instruct Augment

STAGE TRANSITION *(NONE -> STAGING -> PRODUCTION -> ARCHIVED)*

```
model_version = 1
new_stage = "Staging"
model_name = "housing_linear_regression_400"

client.transition_model_version_stage(
    name=model_name,
    version=model_version,
    stage=new_stage,
    archive_existing_versions=False
)
```

tracking.ipynb | models.ipynb | model-registry (1).ipynb | outlier.ipynb

Users > reza > Downloads > model-registry (1).ipynb > 3.3. Stage Transition > model_version=1

Generate | + Code | + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | .venv (3.13.3) (Python 3.13.3)

3.2. Version Search

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
from mlflow.tracking import MlflowClient
model_name = "CatDogClassifier"
client = MlflowClient(tracking_uri="http://127.0.0.1:5000/")

latest_versions = client.get_latest_versions(name=model_name)

for version in latest_versions:
    print(f"version: {version.version}, stage: {version.current_stage}")
```

[6] ✓ 0.0s Python

```
version: 1, stage: None
/var/folders/dx/hkx51pqs1vxc2zsq74nxsjp80000gn/T/ipykernel_2799/3548075878.py:5: FutureWarning: ``mlflow.tracking.client.Mlfl
  latest_versions = client.get_latest_versions(name=model_name)
```

AUGMENT

New Chat

GitHub Copilot suggestions will collide with Augment's suggestions. Turn off Copilot to fix this.

Disable Github Copilot

Your subscription is no longer active

Upgrade Now

model-

Ask or instruct Augment

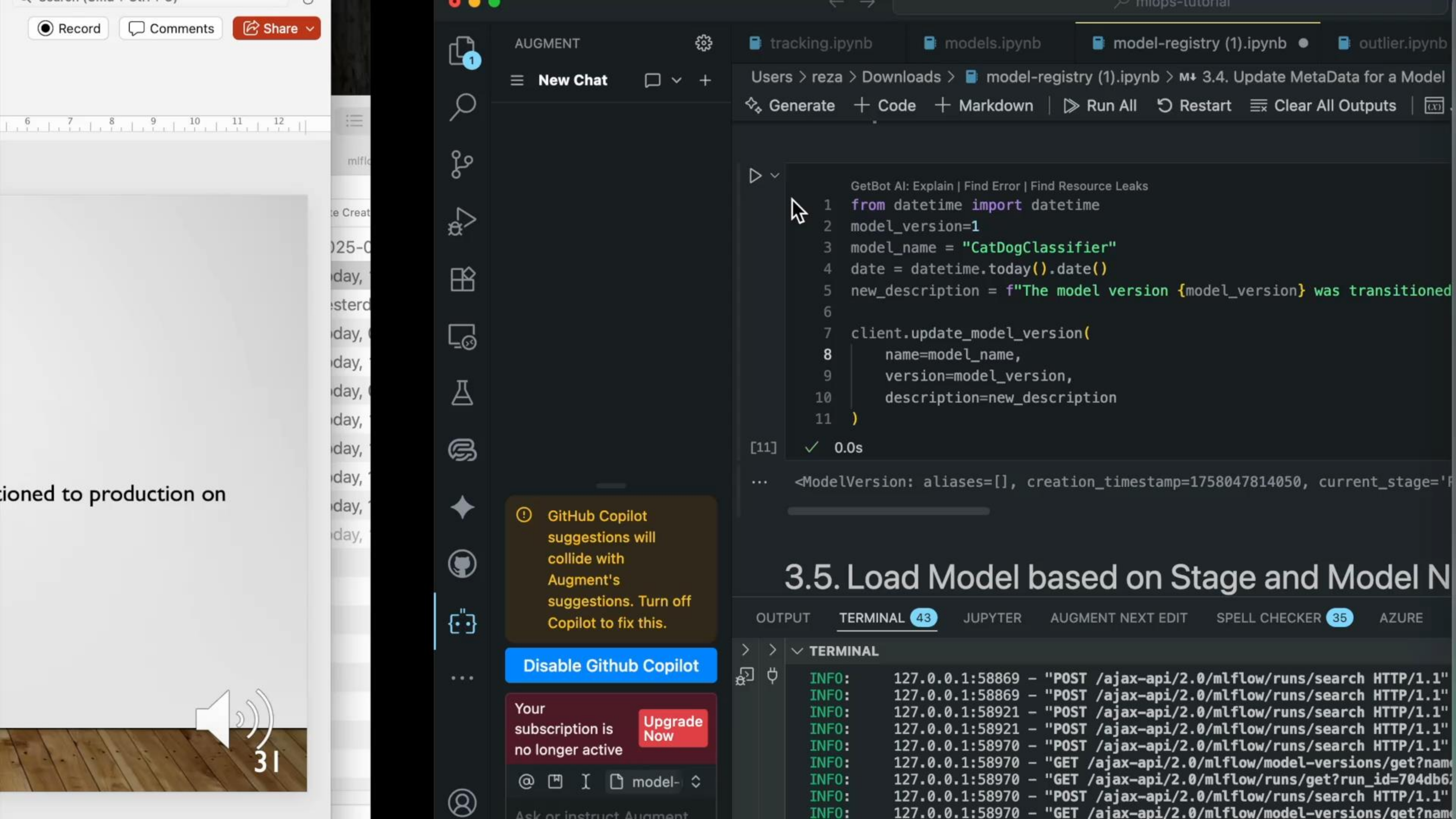
OUTPUT | TERMINAL 43 | JUPYTER | AUGMENT NEXT EDIT | SPELL CHECKER 35 | AZURE

TERMINAL

```
INFO:     127.0.0.1:57200 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57200 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57237 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57237 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57280 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57280 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57365 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57365 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
INFO:     127.0.0.1:57444 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1" 200 OK
```

UPDATE METADATA FOR A MODEL

1. from datetime import datetime
2. model_version=1
3. model_name = "housing_linear_regression_400"
4. date = datetime.today().date()
5. new_description = f"The model version {model_version} was transitioned to production on {date}"
6. **client.update_model_version**(
   name=model_name,
   version=model_version,
   description=new_description
)


Record
Comments
Share
tioned to production on
31
AUGMENT
New Chat
GitHub Copilot suggestions will collide with Augment's suggestions. Turn off Copilot to fix this.
Disable Github Copilot
Your subscription is no longer active
Upgrade Now
tracking.ipynb
models.ipynb
model-registry (1).ipynb
outlier.ipynb
Users > reza > Downloads > model-registry (1).ipynb > 3.4. Update MetaData for a Model
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
GetBot AI: Explain | Find Error | Find Resource Leaks
from datetime import datetime
model_version=1
model_name = "CatDogClassifier"
date = datetime.today().date()
new_description = f"The model version {model_version} was transitioned
client.update_model_version(
name=model_name,
version=model_version,
description=new_description
)
[11]
0.0s
<ModelVersion: aliases=[], creation_timestamp=1758047814050, current_stage='
3.5. Load Model based on Stage and Model N
OUTPUT
TERMINAL 43
JUPYTER
AUGMENT NEXT EDIT
SPELL CHECKER 35
AZURE
TERMINAL
INFO: 127.0.0.1:58869 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1"
INFO: 127.0.0.1:58869 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1"
INFO: 127.0.0.1:58921 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1"
INFO: 127.0.0.1:58921 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1"
INFO: 127.0.0.1:58970 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1"
INFO: 127.0.0.1:58970 - "GET /ajax-api/2.0/mlflow/model-versions/get?nam
INFO: 127.0.0.1:58970 - "GET /ajax-api/2.0/mlflow/runs/get?run_id=704db6
INFO: 127.0.0.1:58970 - "POST /ajax-api/2.0/mlflow/runs/search HTTP/1.1"

LOAD MODEL BASED ON STAGE AND MODEL NAME

1. model_name = "housing_linear_regression_400"
2. stage = "Production"
3. model = **mlflow.pyfunc.load_model**(f"models:/{model_name}/{stage}")

mlops-tutorial

EXPLORER

MLOPS-TUTORIAL
- .venv
- mlartifacts
- mlruns
- videos
- data.xlsx
- housing-model.pkl
- mlflow.db
- models.ipynb
- models.py
- outlier.ipynb
- outlier.py
- tracking.ipynb
- tracking.py

OUTLINE
- 3- MLflow Model Re...
- 3.1. Model Registry
- 3.2. Version Search
- 3.3. Stage Transition
- 3.4. Update MetaDa...

tracking.ipynb | models.ipynb | model-registry (1).ipynb | outlier.ipynb

Users > reza > Downloads > model-registry (1).ipynb > 3.6. Model Load and Execute > from sklearn.metrics import mean_squared_error

Generate | + Code | + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | ... | .venv (3.13.3) (Python 3.13.3)

```python
stage = "Production"  # Can be None, "Staging", "Production", or "Archived"

model = mlflow.pyfunc.load_model(f"models:/{model_name}/{stage}")     Pin selection to current chat prompt (Cmd+Option+
```

Python

3.6. Model Load and Execute

GetBot AI: Explain | Find Error | Find Resource Leaks

```python
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
import pandas as pd
import kagglehub
import mlflow

def load_data(kagglehub_path: str):
    path = kagglehub.dataset_download(kagglehub_path)
    df = pd.read_csv(path + "/housing.csv")
    df.dropna(inplace=True)
    xtrain, xtest, ytrain, ytest = train_test_split(df.drop(columns=['median_house_value','ocean_proximity']),
                                                     df['median_house_value'], test_size=0.4, random_state=42)
    return xtrain, xtest, ytrain, ytest

%time print(f'rmse: {mean_squared_error(ytest, model.predict(xtest))}')
```

Python

OUTPUT | TERMINAL 58 | JUPYTER | AUGMENT NEXT EDIT | SPELL CHECKER 50 | AZURE

THANK YOU!
ANY QUESTIONS?

1- Tracking
2- Models
3- Model Registry

a- set experiment
b- search runs
c- log models
d- load model
e- register model
f- search model
g- set model stage
h- update model metadata
i- load and execute a model