RV College of
Engineering

Go, Change the world

Machine Learning Operations (MLOPS)

comet ML

PRESENTED BY:

AKSHIT AGARWAL	(1RV22AI004)
RAKESH H G	(1RV22AI042)
NISHANTH UDUPA	(1RV22AI033)
NITINKUMAR LONI	(1RV22AI034)

AGENDA

- Introduction
- Why Comet ML
- Key Components of Comet ML
- Key Benefits of Comet ML
- System requirements
- Installation Process
- Model versioning
- Limitation of Comet ML

Introduction

- CometML is a powerful and versatile machine learning **experiment tracking platform** that enables data scientists, researchers, and developers to efficiently manage, monitor, and **compare their machine learning workflows.**
- Designed to enhance collaboration and productivity, CometML provides a suite of tools for tracking experiments, optimizing models, and visualizing results.
- With its seamless integration into popular machine learning frameworks such as TensorFlow, PyTorch, and Scikit-learn, CometML allows **users to log hyperparameters, metrics, data versions, and even custom visualizations**.
- Its dashboard makes it easy to analyze trends, identify performance bottlenecks, and reproduce results, **ensuring transparency and reproducibility** in your ML projects.

Introduction - Comet ML

- **Purpose-Built for Experimentation**: Unlike AWS, Azure, or Google Vertex AI, where experiment tracking is a secondary feature, CometML is designed explicitly for tracking experiments. It provides in-depth visibility into hyperparameters, metrics, code versions, and data versions in a single dashboard.
- **Framework Compatibility:** CometML integrates seamlessly with most machine learning frameworks (e.g., TensorFlow, PyTorch, Scikit-learn, XGBoost) without being tied to a specific ecosystem like AWS or Azure.
- **Hyperparameter Optimization**: CometML allows you to perform hyperparameter sweeps and visualize how specific changes impact results, often with more granularity than standard features in AWS or Azure.

Introduction - Comet ML

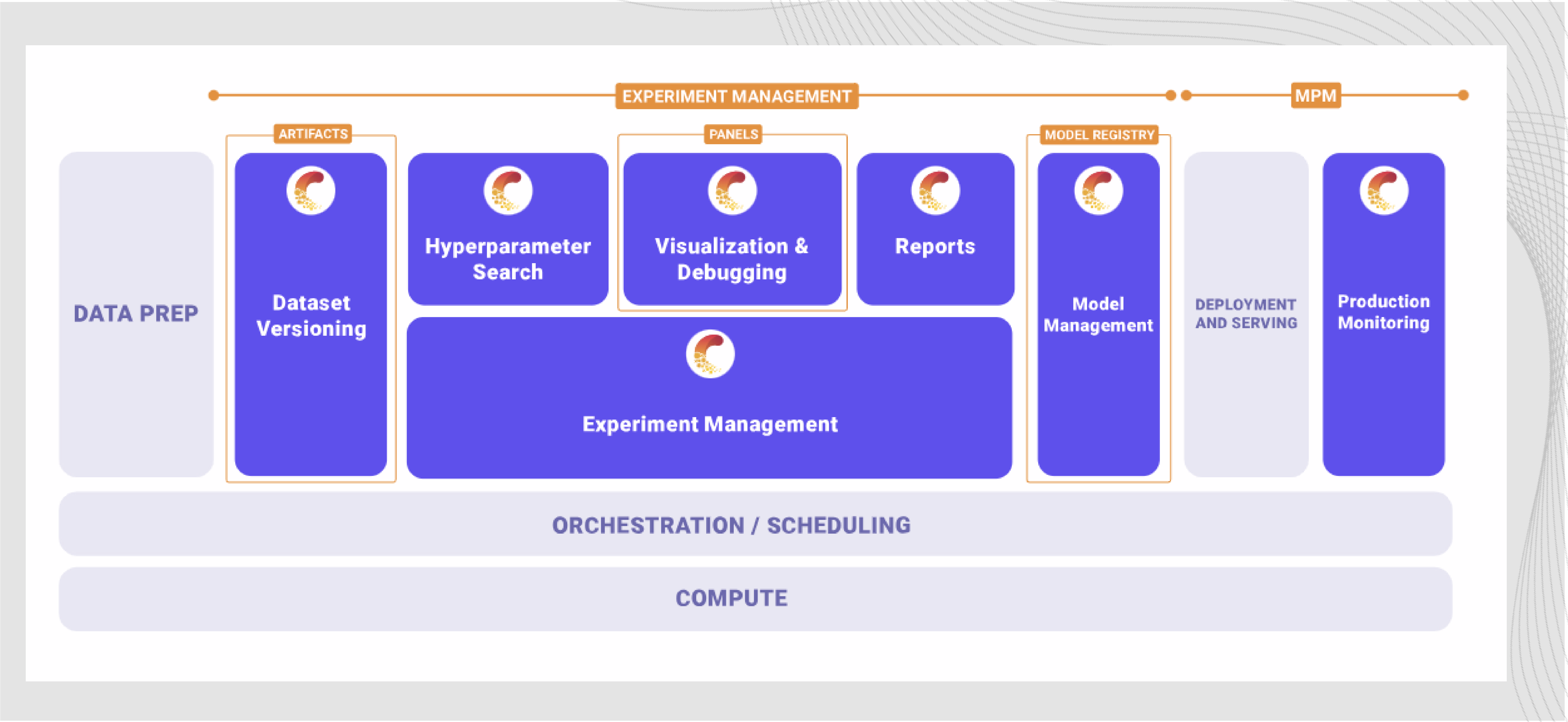
- **Rich Visualizations:** You can track custom metrics, create live plots, and analyze experiment trends in ways that are often not as easily accessible in generic deployment tools.
- **Cost Efficiency:** CometML can be significantly more affordable for experiment tracking since it's not designed for full production pipelines. AWS or Azure tools can become expensive as you scale, especially when their experiment tracking features are bundled with broader infrastructure costs.

To summarise if the organization is developing a model where experiment tracking is highly important and utilizes different environments and workflows, then CometML is the go-to choice.

RV College of Engineering

WHY COMET ML ?

Go, Change the world



WHY COMET ML ?

Experiment Tracking : Automatically logs hyperparameters, metrics, artifacts (models, logs, datasets), and system configurations (e.g., GPU/CPU usage) during training.
Provides centralized dashboards for tracking and comparing experiments.

Reproducibility : Logs the code version, dataset version, and environment (libraries, dependencies) used for each experiment.
Ensures experiments can be re-run with the exact same configuration.

Model Performance Monitoring : Tracks performance metrics in real-time and provides tools to visualize training progress.
Offers comparisons of metrics across experiments for insights into model improvements.

WHY COMET ML ?

Collaboration : Centralized dashboards and experiment logs can be shared among team members.
Enables commenting and discussion within the platform.

Visualization : Provides visualizations for loss curves, feature importance, confusion matrices, and more.

Hyperparameter Optimization : Integrates with hyperparameter optimization tools like Optuna and Ray Tune to manage experiments and visualize results.

Compliance and Documentation : Logs detailed experiment histories, making it easy to produce documentation or audit trails.

Key Components Of CometML

1. Experiment Tracking

Free:
Logs parameters, metrics, and output files for easy comparison.

Paid/Enterprise:
Real-time collaboration, advanced visualizations, and historical tracking with unlimited experiments.

2. Model Management

Free: Track and version models with metadata.

Paid/Enterprise:
Deploy models and manage complex model lifecycles with automatic model versioning and rollback features.

3.Data and Artifact Management

Free:
Upload datasets, results, and logs to track and visualize.

Paid/Enterprise:
Enhanced artifact management, with support for large datasets, and storing custom files, model checkpoints, and results for future use or collaboration.

4. Collaboration & Sharing

Free:
Share experiments, collaborate on notebooks.

Paid/Enterprise:
Real-time collaboration, team-wide sharing, and project-level access control.

Key Components Of CometML

Free vs Paid/Enterprise Features

Free Tier

- Track experiments, log parameters and metrics.
- Upload small datasets and models.
- Limited access to advanced collaboration and visualization tools.

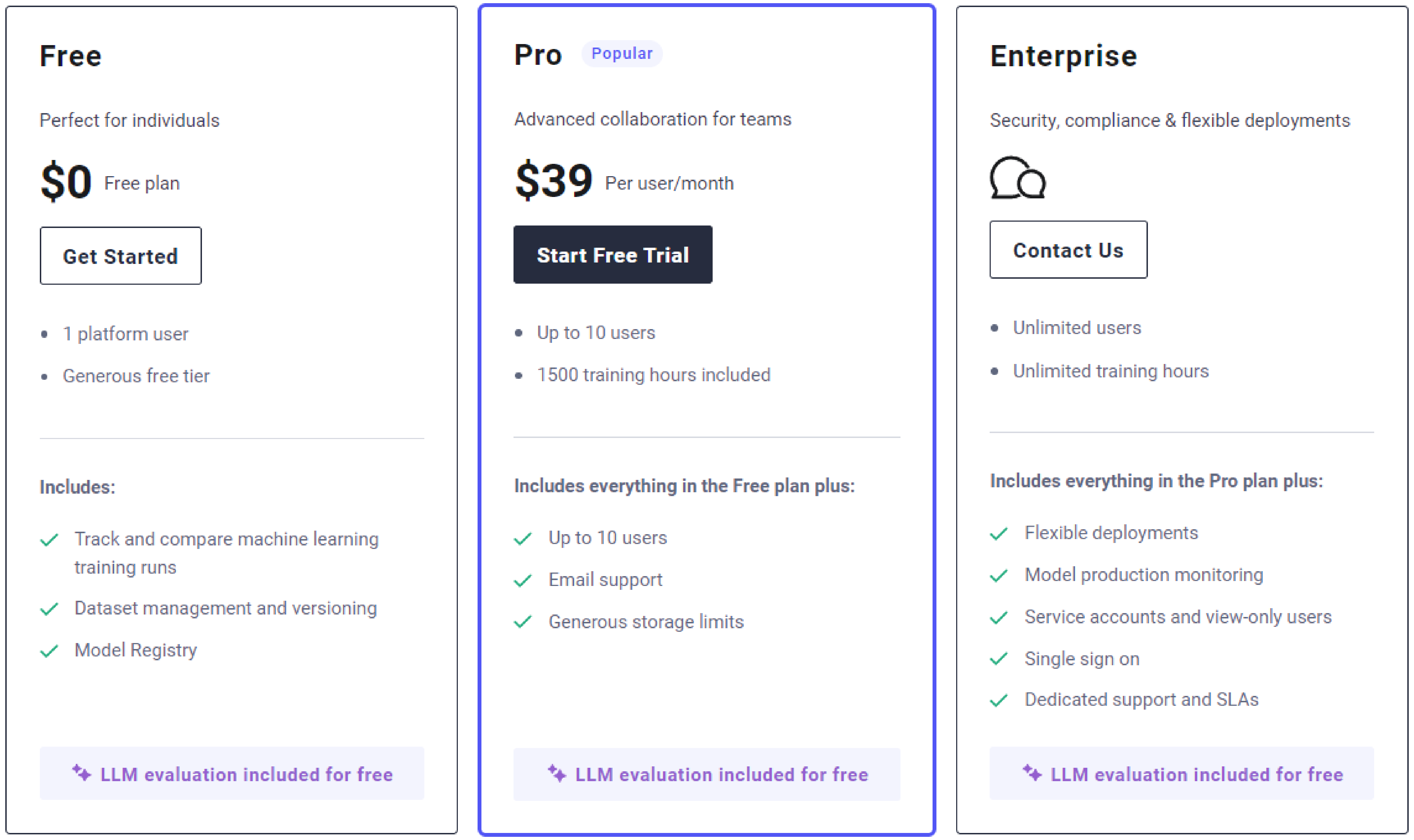
Paid Tier

- Unlimited experiments and artifact storage.
- Enhanced visualization and reporting capabilities.
- Advanced collaboration features with access control.

Enterprise Tier

- Dedicated support and integration with enterprise systems.
- Enhanced security features and user management.
- Customized workflows and dedicated cloud deployment options.

CometML Pricing (As on 11/01/2025)



Free

Perfect for individuals

$0 Free plan

Get Started

- 1 platform user
- Generous free tier

Includes:

✓ Track and compare machine learning training runs
✓ Dataset management and versioning
✓ Model Registry

LLM evaluation included for free

Pro Popular

Advanced collaboration for teams

$39 Per user/month

Start Free Trial

- Up to 10 users
- 1500 training hours included

Includes everything in the Free plan plus:

✓ Up to 10 users
✓ Email support
✓ Generous storage limits

LLM evaluation included for free

Enterprise

Security, compliance & flexible deployments

Contact Us

- Unlimited users
- Unlimited training hours

Includes everything in the Pro plan plus:

✓ Flexible deployments
✓ Model production monitoring
✓ Service accounts and view-only users
✓ Single sign on
✓ Dedicated support and SLAs

LLM evaluation included for free

Key Benefits Of CometML

1. **Improved Experiment Management**
    - Effortlessly track and compare experiments with detailed version control.
    - Integrates with popular ML frameworks for seamless tracking.

2. **Seamless Collaboration**
    - Share custom dashboards and experiment results in real time.
    - Receive instant notifications and feedback for faster team iterations.

3. **Advanced Analytics**
    - Log custom metrics and create tailored visualizations for deeper insights.
    - Optimize hyperparameters with advanced visualization tools.

4. **Scalable and Secure**
    - Role-Based Access Control ensures secure team collaboration.
    - Flexible deployment options with cloud or on-premise setups.

System Requirements

For Users (Client-Side)

- **OS**: Windows, macOS, Linux
- **Python Version**: Python 3.6+
- **Dependencies**: TensorFlow, PyTorch, or any ML library
- **Network**: Requires internet connection for logging

For Deployment/Server-Side/ Self-hosting

- **RAM**: Minimum 8GB (recommended 16GB for heavy models).
- **CPU/GPU**: High-performance CPUs and optionally GPUs for faster inference.
- **Storage:** Enough to store datasets, checkpoints, and logs.
- **Network**: Stable internet connection for dashboard access.

Installation Process

For Users (Client-Side)

1.Install CometML using pip: pip install comet_ml

2.Create an account on CometML -

(https://www.comet.com/).

3. **Set up an API key from your CometML dashboard.**

4. **Integrate CometML into your project**

```
from comet_ml import Experiment
    experiment = Experiment(api_key="YOUR_API_KEY",
project_name="music-generation")
```

5. **Log hyperparameters, metrics, and outputs:**

```
    experiment.log_parameter("learning_rate", 0.001)
    experiment.log_metric("loss", loss_value)
    experiment.log_audio("output_music.wav",
step=epoch)
```

For Deployment/Server-Side (Optional)

1.System Requirements:

- Python 3.6+
- Compatible with TensorFlow, PyTorch, Keras, and other ML frameworks.
- Requires internet connection for cloud-based logging.

2.For self-hosted deployments, contact CometML for enterprise solutions.

3.Automate logging for all team projects with API integrations:

```
pip install comet_ml[all]
```

- Use environment variables for keys for secure deployments.

Model Versioning

- Model versioning in CometML is implemented through its Model Registry, which allows you to version, organize, and manage machine learning models efficiently.

- The Model Registry in CometML is designed to streamline model lifecycle management, ensuring that all versions of a model are logged, tracked, and easy to retrieve.

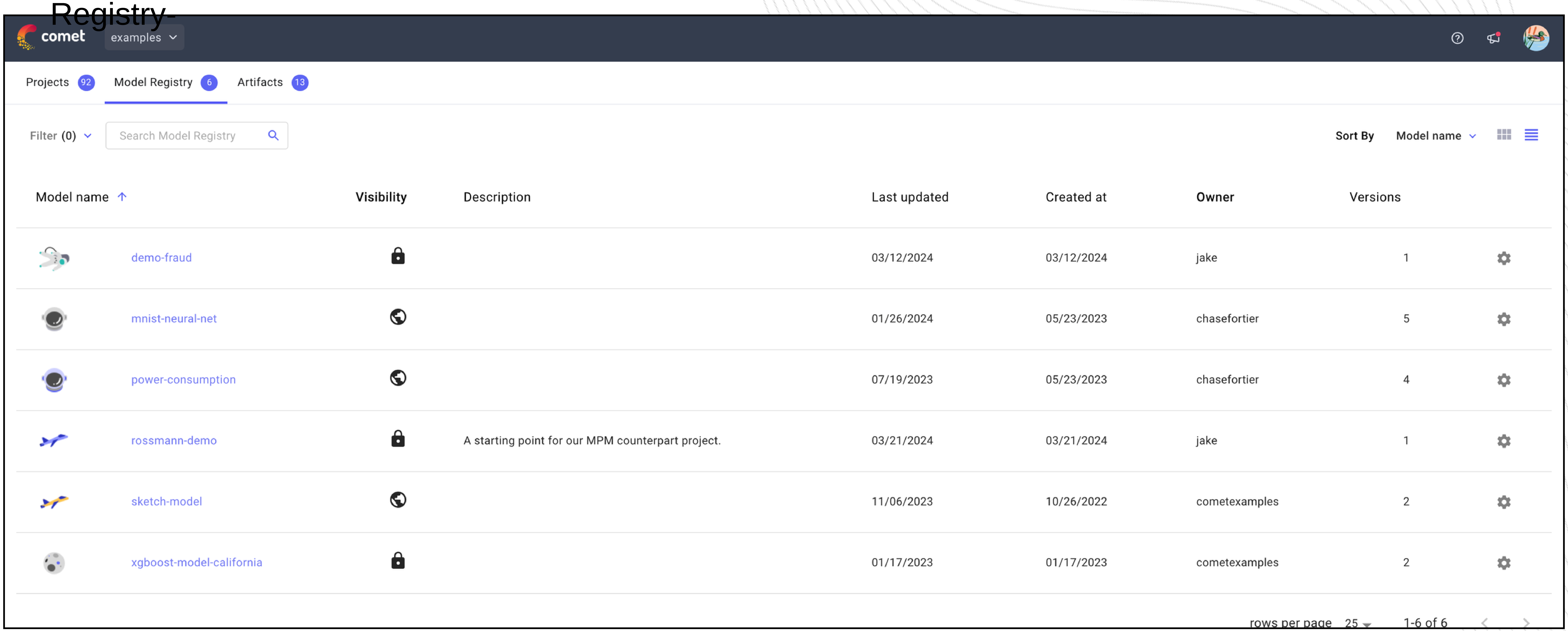
- CometML automatically versions models when they are logged to the Model Registry.

- If no specific version is provided, CometML assigns incremental versions (e.g., v1, v2, v3, etc.).

- Each model version is tied to the experiment that generated it, making it traceable.

Model Versioning

Model Registry-

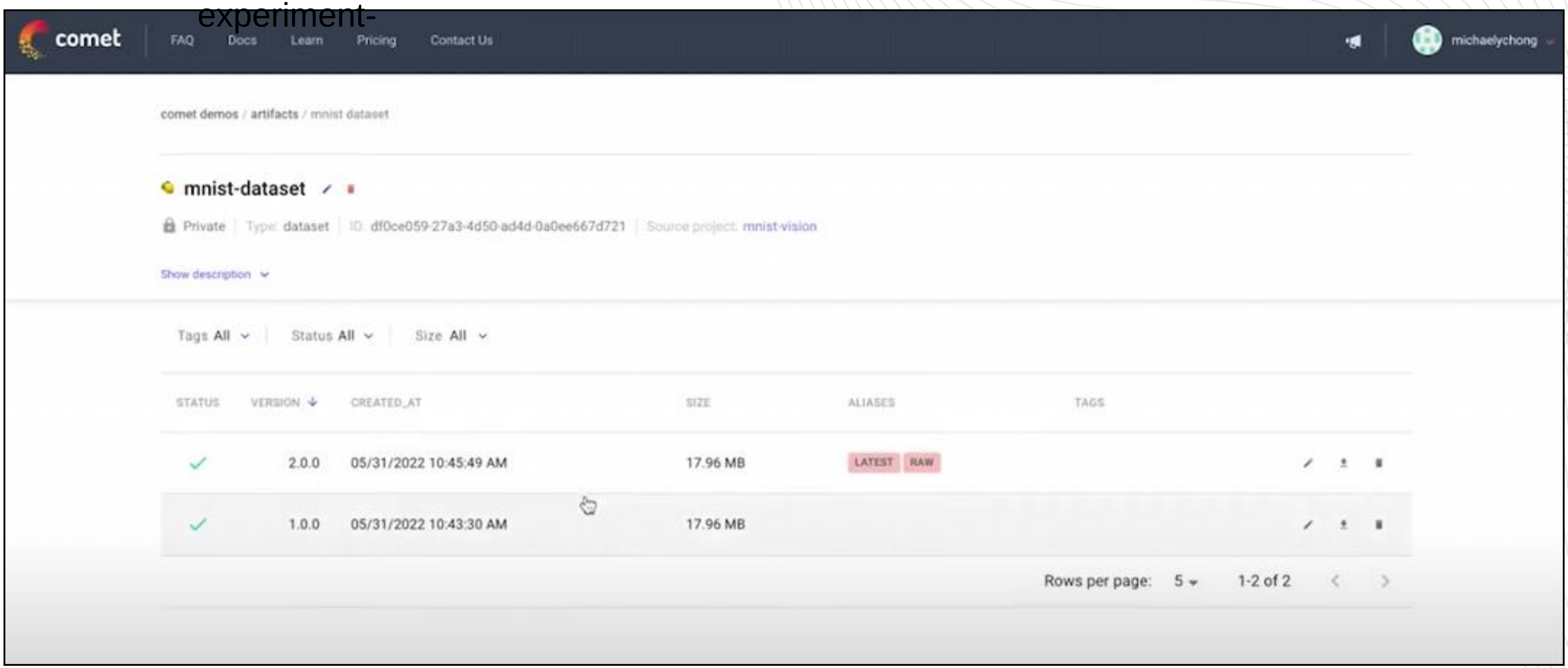


Model name ↑	Visibility	Description	Last updated	Created at	Owner	Versions
demo-fraud	🔒		03/12/2024	03/12/2024	jake	1
mnist-neural-net	🌐		01/26/2024	05/23/2023	chasefortier	5
power-consumption	🌐		07/19/2023	05/23/2023	chasefortier	4
rossmann-demo	🔒	A starting point for our MPM counterpart project.	03/21/2024	03/21/2024	jake	1
sketch-model	🌐		11/06/2023	10/26/2022	cometexamples	2
xgboost-model-california	🔒		01/17/2023	01/17/2023	cometexamples	2

rows per page 25 1-6 of 6

Model Versioning

Model versioning for each experiment-



comet demos / artifacts / mnist dataset

mnist-dataset

Private | Type: dataset | ID: df0ce059-27a3-4d50-ad4d-0a0ee667d721 | Source project: mnist-vision

Show description

Tags All | Status All | Size All

STATUS	VERSION	CREATED_AT	SIZE	ALIASES	TAGS
✓	2.0.0	05/31/2022 10:45:49 AM	17.96 MB	LATEST RAW	
✓	1.0.0	05/31/2022 10:43:30 AM	17.96 MB		

Rows per page: 5 1-2 of 2

Shortcomings of CometML

Data Management : While Comet can track datasets used in experiments, it doesn’t handle dataset creation, cleaning, augmentation, or large-scale storage.

Solution : Use data versioning tools like DVC (Data Version Control) or Pachyderm for managing dataset versions. Implement robust data cleaning and preprocessing pipelines using libraries like Pandas or Apache Spark.

Debugging Complex Pipelines Comet can help log outputs and track progress but doesn’t inherently debug or identify issues in ML pipelines.

Solution : Use debugging tools like PDB in Python or IDE-specific debuggers. Implement robust logging using Loguru or MLFlow to pinpoint errors.

Deployment Monitoring While Comet can track training metrics and logs, post-deployment monitoring (e.g., for model drift or latency) is outside its core scope.

Solution : Use deployment monitoring tools like Evidently AI or Seldon Core for production performance monitoring.

Shortcomings of CometML

Integration and Scaling

Scaling issues depend on infrastructure (e.g., GPUs, clusters) rather than experiment tracking.

Solution : Use orchestration tools like Kubernetes or cloud platforms (AWS, GCP, Azure) for scalability.
Optimize ML pipelines using frameworks like TensorRT for inference.

Data Bias and Fairness

Comet doesn’t analyze data bias or fairness issues inherent in datasets or model outputs.

Solution : Use tools like IBM AI Fairness 360 or Google’s What-If Tool to detect and mitigate bias.

Team Workflow Integration

Workflow integration requires managing tools for version control, CI/CD, and task management beyond what Comet offers.

Solution : Combine Comet with tools like GitHub Actions for CI/CD, Jira for project management, and Slack for communication.

Thank You!