



Hi!
How can I help you?



Many AI/ML projects stall before scaling...

According to S&P Global data, **51%** of AI projects
are in production but not delivered at scale.

Source: 2024 Global AI Trends Report by Weka



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Why AI/ML Solutions Fail & What It Takes to Build Ones That Last

Lydia Ray

Sr Analytics Solution Architect
AWS

Anastasia Tzeveleka

Sr GenAI/ML Solution Architect
AWS

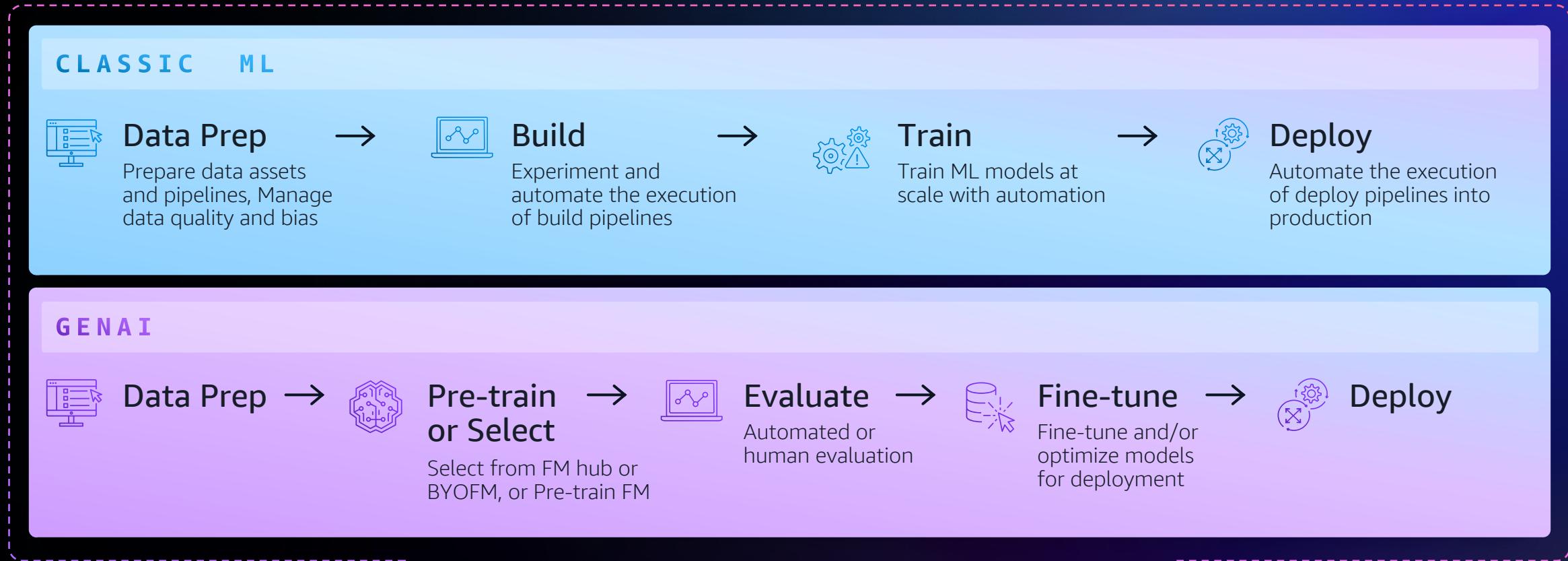


What's On

- 01** Why Building AI/ML Platforms is Hard
- 02** Getting The Data Right
- 03** Accelerating AI/ML lifecycle
- 04** Building AI/MLOps foundation
- 05** Key Takeaways & Lessons Learned

Why Building AI/ML Solution is Hard

Classic ML and GenAI lifecycle



Building an AI/ML solution is hard,

not because of algorithms, but because of everything else around them.



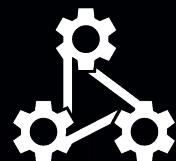
Data Foundation

Poor quality, lack of observability, inconsistent governance



Business Alignment

Difficulty measuring ROI, aligning with operations, and sustaining outcomes



Tool Integration

Fragmented tools, scaling inefficiencies, cost challenges



Security & Compliance

Increasing regulatory pressure, ethical risks, fragile protections



Collaboration

Organizational silos, misaligned priorities, unclear ownership



MLOps Practices

Immature pipelines, manual workflows, poor monitoring

What we are going to talk about today

1 | Getting the data right

- Data Pipeline (ETL/ELT)
- Data Quality
- Data Governance and Lineage
- Data Democratization



Data Foundation

Poor quality, lack of observability, inconsistent governance

2 | Accelerate the AI/ML lifecycle

- Unify the development experience
- Collaborate easily
- Select the right model and train at scale
- Deploy with best price-performance



Tool Integration

Fragmented tools, scaling inefficiencies, cost challenges



MLOps Practices

Immature pipelines, manual workflows, poor monitoring

3 | Build your MLOps foundation

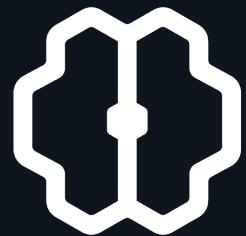
- Automate your ML workflows
- Build CI/CD pipelines
- Track lineage



Collaboration

Organizational silos, misaligned priorities, unclear ownership

THE NEXT GENERATION OF



Amazon SageMaker

The center for all your data, analytics, and AI

Getting The Data Right

ETL/ELT Done Right

MAKE PIPELINES THAT SCALE, AUTOMATED, REUSABLE, EFFICIENT, AND OBSERVABLE



Design **modular, reusable pipeline** components.



Early **filtering and store data efficiently** to cut compute and I/O costs.



Automate **retries**, use event-driven **triggers**, and optimize **scheduling**.



Monitor pipeline health (latency, volume, schema drift).

Data Quality

BUILD TRUST IN DATA, PREVENT SILENT FAILURES, SUPPORT AUTOMATION



Enforce quality checks (e.g., missing values, invalid ranges, duplicates) **before and after ingestion.**



Automate validation by embedding rules within the pipeline to validate the data at each stage.



Use alerting to flag issues early and reduce downstream cost from retraining or bad inference.

Data Governance and Lineage

PROVIDE FULL TRANSPARENCY, ENSURE COMPLIANCE, AND ENABLE CROSS-TEAM CLARITY



Tag datasets with owners, access levels, versions, and expiration dates.



Track lineage from raw ingestion to model features.



Enforce RBAC policies to **control access and audit** usage.



Automate data retention, versioning, and archival to reduce cloud costs and maintain compliance.



Create **shared metadata** for visibility across data science, engineering, and security teams.

Data Democratization

ENABLE DIVERSE ROLES TO USE SHARED DATA WITHOUT FRICTION OR RISK



Provide curated, trusted datasets through a **shareable storage**.



Empower non-technical users (e.g., analysts, product teams) to explore data via notebooks or visual tools.



Standardize metadata naming conventions with **Business Data Catalog**.



Enforce controlled access to **sensitive data** via roles and masking.



Encourage **feedback loops**: let users flag broken datasets, request new fields, or suggest improvements.



See it in action
with Amazon SageMaker



Amazon SageMaker Unified Studio

The center for data, analytics, and AI

[Sign in with SSO](#)

[Sign in with AWS IAM](#)

Accelerating AI/ML Lifecycle



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training a Machine Learning Model

Whether you are **training a ML model** or **fine-tuning a Large Language Model (LLMs)** you need to

Iterate fast

Allocate suitable compute
for training and
evaluation

Use low-code tools for
initial development

Collaborate easily

Scale

Scale with distributed
training

Run training on multiple
nodes, fast

Track experiments

Run multiple model
variations simultaneously

Track experiments

Automate evaluation

Establish criteria for
model promotion to prod

Diagnose

Debug

Review logs

Profile performance



Amazon SageMaker AI for Large Scale Model Training



Resilient environment

Self-healing clusters reduce training time by up to 20%



Streamline distributed training

SageMaker distributed training libraries improve performance by up to 20%



Managed training environment: **Training Jobs**

Focus on ML without the need to manage infrastructure

Optimized resources utilization: **Hyperpod**

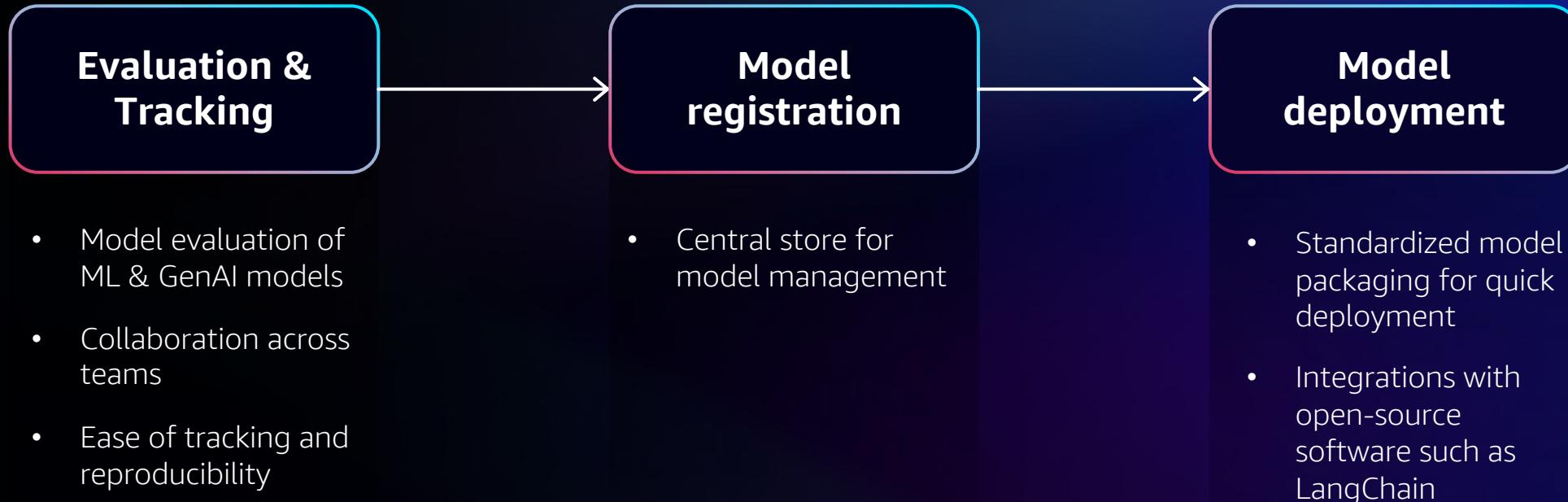
Control over compute environment and workload scheduling

AND

AND

OR

mlflow™ on Amazon SageMaker AI for experiment tracking



Amazon SageMaker JumpStart

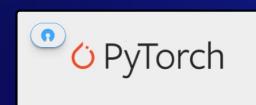
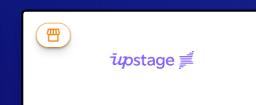
MODEL HUB WITH FOUNDATION MODELS, BUILT-IN ALGORITHMS, AND PREBUILT AIML SOLUTIONS

Over 250+ publicly available foundation models

Hundreds of built-in algorithms with pretrained models from popular model hubs

Fully customizable solutions for common use cases with reference architectures

Share AI and ML models and notebooks across your organization

 HuggingFace Explore hundreds of popular and trending models from HuggingFace. View 368 models >	 Meta Explore popular and trending models from Meta including Llama, Code Llama, and more. View 44 models >	 AI21Labs Explore popular and trending models from AI21 Labs including Jurassic and more. View 6 models >	 stability.ai Explore popular and trending models from StabilityAI including Stable Diffusion and more. View 11 models >
 Cohere Explore popular and trending models from Cohere including Command, Rerank, and more. View 12 models >	 TensorFlow Explore popular and trending models from TensorFlow for computer vision and NLP tasks. View 319 models >	 PyTorch Explore popular and trending models from PyTorch for computer vision and NLP tasks. View 34 models >	 Upstage Explore popular and trending models from Upstage including Solar mini chat model and more. View 4 models >
 LightOn Explore popular and trending models from LightOn including mini-instruct models. View 2 models >	 NCSoft Explore popular and trending models from NCSoft including VARCO LLM models. View 3 models >	 LG CNS Explore popular and trending models from LG CNS including EXAONE Atelier and more. View 1 models >	 Jina AI Explore popular and trending models from Jina AI including Jina Embeddings model and more. View 1 models >
 Voyage AI Explore popular and trending models from Voyage AI including Voyage-2 Embedding Model and more. View 3 models >	 Nomic Explore popular and trending models from Nomic including Nomic Embed Model and more. View 2 models >	 EvolutionaryScale, PBC Explore frontier, generative language models for biology from EvolutionaryScale, including ESM3. View 1 models >	 Amazon Explore popular and trending models from AWS for computer vision, NLP, and tabular tasks. View 36 models >



Deploy AI & ML models

USING PURPOSE BUILT TOOLS

Infrastructure management and monitoring

- ✓ Infrastructure choice
- ✓ Serving stack installation
- ✓ Operational health
- ✓ High Availability
- ✓ Automatic node replacement
- ✓ Storage configuration
- ✓ Auto-scaling

Model hosting

- ✓ Deployment modes
- ✓ Model versioning
- ✓ Fast model downloads
- ✓ Low code/no code set-up
- ✓ Model release management
- ✓ Deployment guardrails
- ✓ Load testing
- ✓ Update with no availability loss

Monitoring

- ✓ Drift detection, data quality monitoring
- ✓ Performance degradation alerts
- ✓ Feedback loops
- ✓ model auditability, and regulatory compliance

Cost optimization

- ✓ Infrastructure right sizing
- ✓ Optimize price-performance for use case
- ✓ Model compilation



Amazon SageMaker AI for Model Deployment



Wide selection of infrastructure

70+ instance types with varying levels of compute and memory to meet the needs of every use case.



Deploy models in production for inference for any use case

From low latency and high throughput to long-running inference



Cost-effective deployment

Reduce inference cost by at least 50% with multi-model/multi-container endpoints, serverless inference, and elastic scaling



Shadow testing & automatic deployment guardrails

Validate the performance of new ML models against production models. Minimize risk when deploying new model using linear, canary, or blue green traffic switching



Built-in integration for MLOps

ML workflows, CI/CD, feature management, lineage tracking, and model management



Large model inference container

Achieve best price performance with the latest inference optimizations tools, model servers, and libraries packaged into a single container

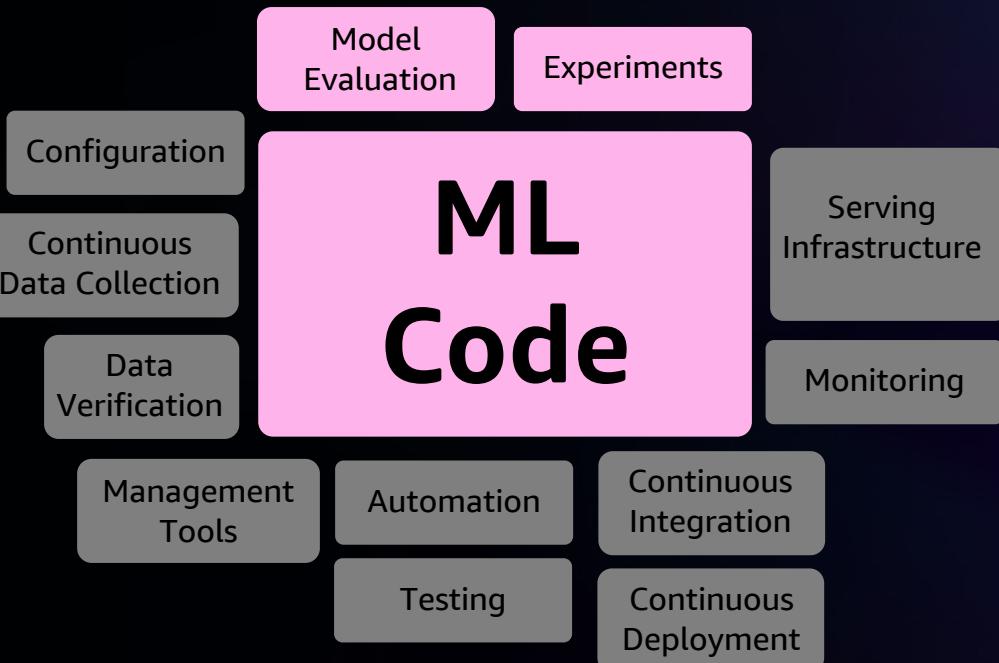


See it in action
with Amazon SageMaker

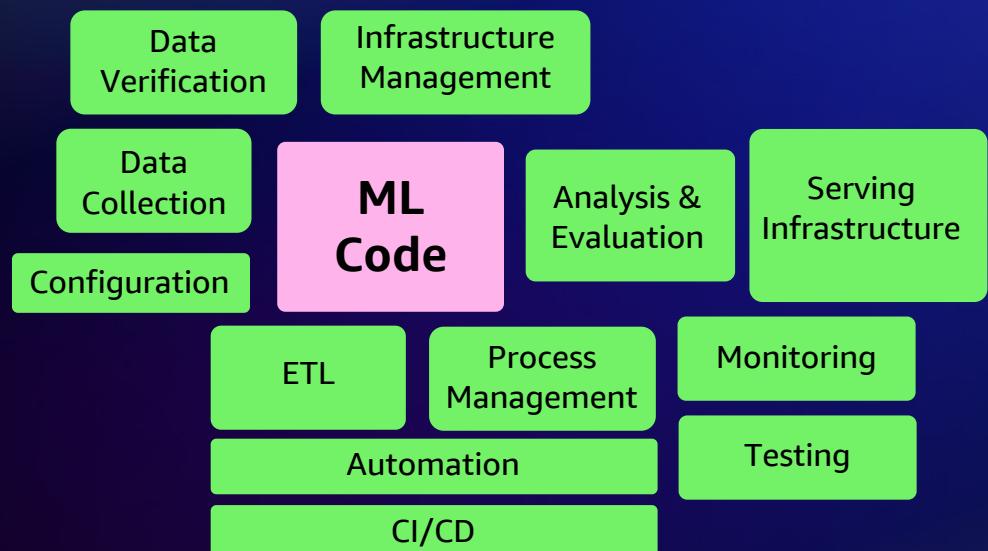
Building AI/MLOps Foundation

ML: Pilots vs Operationalizing

Pilot Phase



Operational Phase



Purpose: Answer the question “Is this possible, and should we proceed?”

Purpose: Put the system in production and achieve desired business value

MLOps Best Practices

Automation & Reproducibility

Use version-controlled infrastructure as code

CI/CD for ML: Automate testing, validation, and deployment

Pipeline Orchestration: Automate end-to-end workflow from data ingestion to deployment

Version artifacts

Governance

Central model registry

Model cards

Track lineage for troubleshooting and compliance

Cross-functional workflows

Automated Evaluation

Established evaluation workflows

Integration and regression testing

Standard metrics and evaluation procedures

Shadow deployments

A/B testing

Monitoring

Logging

Automated alerts

Health dashboards

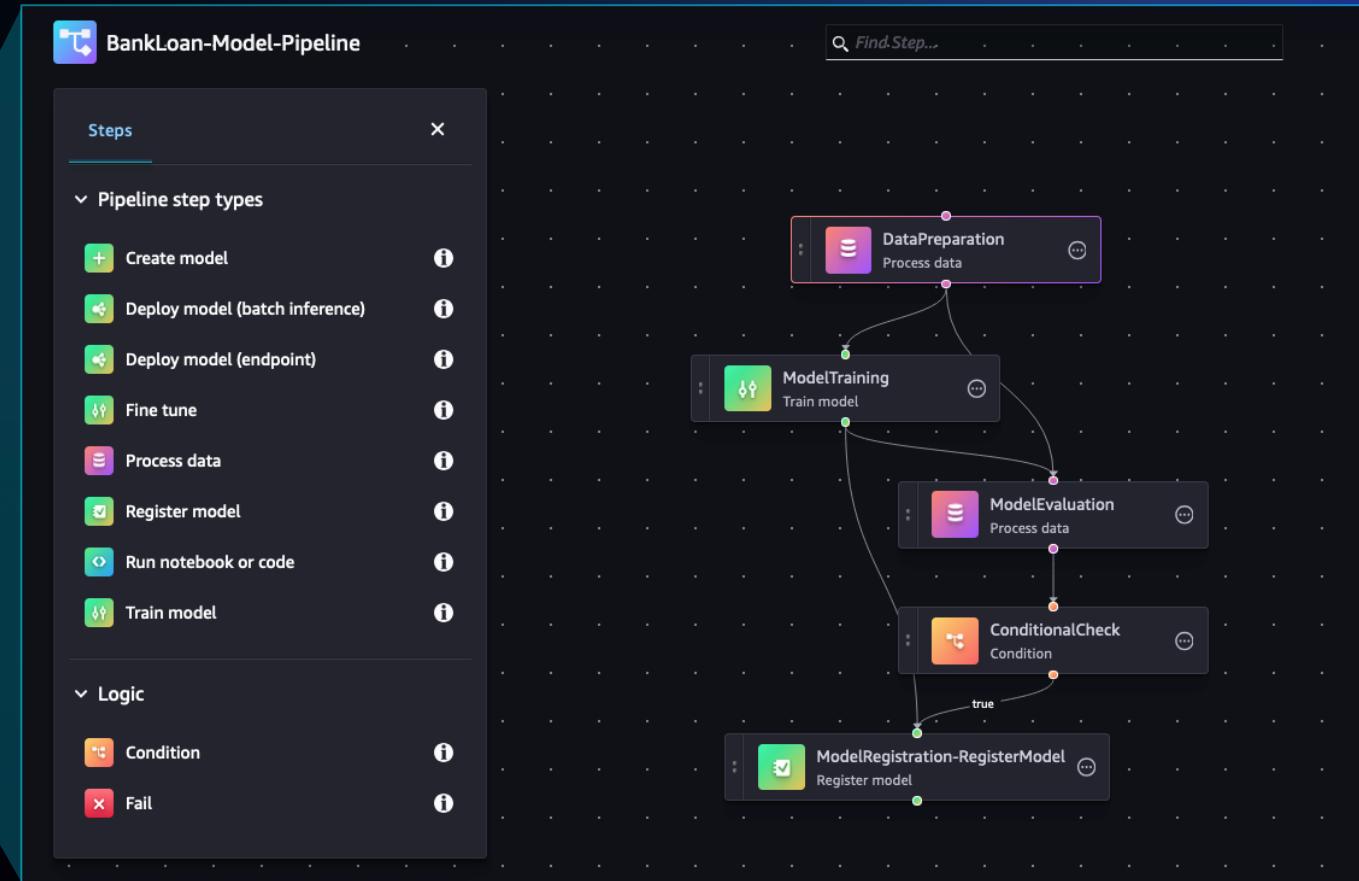
Automated re-training

Explainability tools

SageMaker AI – AI Ops

AI Ops

Drag and drop your ML steps to quickly build a pipeline



SageMaker AI – AI Ops

Select
Integrate fully managed third party GenAI with your SageMaker environment

The screenshot shows the SageMaker AI - AI Ops interface. At the top, there are navigation tabs: Discover, Build, Manage, and a dropdown for the project "Paul_Project_PipelinesTest". Below the navigation, a breadcrumb trail indicates the current location: Home > Projects > Paul_Project_PipelinesTest. A section titled "Partner AI Apps" features a purple icon of a storefront. The text "Discover a broad selection of industry-leading Generative AI and ML development applications." is displayed. Four application cards are listed:

- comet**: Not deployed. Comet is a platform designed to help developers evaluate and monitor machine learning models. [View details >](#)
- deepchecks.**: LLM EVALUATION. Not deployed. Lorizzle ipsum dolor sit stuff, consectetur adipiscing elit. Fizzle sapien velizzle, pot volutizzle, suscipizzle stuff, gravida vizzle, bling bling. [View details >](#)
- fiddler**: Not deployed. Lorizzle ipsum dolor sit stuff, consectetur adipiscing elit. Fizzle sapien velizzle, pot volutizzle, suscipizzle stuff, gravida vizzle, bling bling. [View details >](#)
- LAKERA**: Not deployed. Lakera Guard is a low-latency AI Application Firewall to secure GenAI applications from GenAI-specific threats. [View details >](#)



SageMaker AI – AI Ops

Assist

Have Amazon Q generate your code and quickly insert in your notebook with one click

mair
Create a interactive feature plot using matplotlib inside a cell of a notebook, that showcases distribution of numerical features from a dataframe called df_data. Allow users to choose the feature from a list of numerical features. Use only ipywidgets.

Amazon Q
I'll help you create an interactive plot using matplotlib and ipywidgets that allows users to select and visualize different numerical features from a dataframe. Here's the implementation: [1]

```
import matplotlib.pyplot as plt
import ipywidgets as widgets
from IPython.display import display
import numpy as np

# Get list of numerical columns
numerical_features = df_data.select_dtypes(include=['float64', 'int64'])

# Create the plotting function
def plot_distribution(feature):
    plt.figure(figsize=(10, 6))

    # Plot histogram with density
    plt.hist(df_data[feature], bins=30, density=True)

    # Add kernel density estimation
    density = gaussian_kde(df_data[feature])
    xs = np.linspace(df_data[feature].min(), df_data[feature].max(), 100)
    plt.plot(xs, density(xs), 'r-')

    # Add labels and title
    plt.title(f'Distribution of {feature}')
    plt.xlabel(feature)
    plt.ylabel('Density')
    plt.grid(True, alpha=0.3)
    plt.legend()

    # Show basic statistics
    plt.text(0.02, 0.95,
            f'Mean: {df_data[feature].mean():.2f}, '
            f'StD: {df_data[feature].std():.2f}, '
            f'Median: {df_data[feature].median():.2f}', transform=plt.gca().transData)
```

Untitled.ipynb
import ipywidgets as widgets
from IPython.display import display
import numpy as np

Get list of numerical columns
numerical_features = df_data.select_dtypes(include=['float64', 'int64'])

Create the plotting function
def plot_distribution(feature):
 plt.figure(figsize=(10, 6))

 # Plot histogram with density
 plt.hist(df_data[feature], bins=30, density=True)

 # Add kernel density estimation
 density = gaussian_kde(df_data[feature])
 xs = np.linspace(df_data[feature].min(), df_data[feature].max(), 100)
 plt.plot(xs, density(xs), 'r-')

 # Add labels and title
 plt.title(f'Distribution of {feature}')
 plt.xlabel(feature)
 plt.ylabel('Density')
 plt.grid(True, alpha=0.3)
 plt.legend()

 # Show basic statistics
 plt.text(0.02, 0.95,
 f'Mean: {df_data[feature].mean():.2f},
 f'StD: {df_data[feature].std():.2f},
 f'Median: {df_data[feature].median():.2f}', transform=plt.gca().transData)

Key Takeaways & Lessons Learned

Wrap-Up Thoughts

-  Building AI/ML at scale is hard, but solvable.
-  Get the data right first - without clean, observable, and governed data, models will fail.
-  Automate & scale with MLOps foundation: Pipelines, validation, deployment guardrails.
-  Unify workflow across teams with Amazon SageMaker to reduce silos and accelerate outcomes.

Thank you!

Stay connected with us...

Lydia Ray

 linkedin.com/in/lydiaray

Anastasia Tzeveleka

 linkedin.com/in/atzeveleka

Let's Talk About It