Session 3:
Productionising NLP Model - II

PRODUCTIONISING NLP MODEL - II

- Create an end-to-end pipeline with the following steps
  - Pre-process
  - Train
  - Evaluate
 - Register
  - Deploy
  - Infer

SageMaker PROJECTS

- Create end-to-end ML solutions with CI/CD

- The project template **automates the set-up and implementation**

  - Use your own templates

  - Use the existing ones

- SageMaker Project helps set up

  - Development environment for data scientists

  - CI/CD systems for MLOps engineers

- We used the model build, train and deploy template for our use case

SageMaker EXPERIMENTS

- Ease the process of experiments creation

- Experiments can be populated with trials

  - Pass an extra parameter to the Estimator, defining the name of the experiment that this trial should be attached to

- Run analytics across trials and experiments

- SageMaker Experiments SDK lets you load experiment and trial data in a Pandas DataFrame

- You can also visualize real-time model leaderboards and metric charts
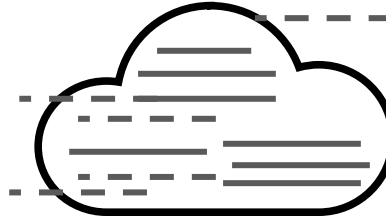
AMAZON SageMaker PIPELINES
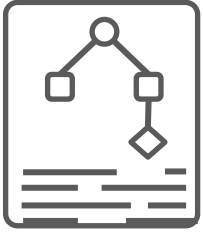
Build ML Workflows

Python SDK for quickly and easily building ML workflows
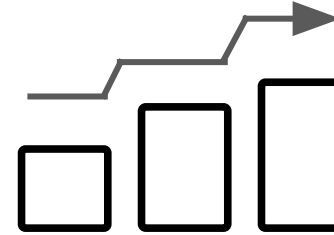
Troubleshoot ML Workflows

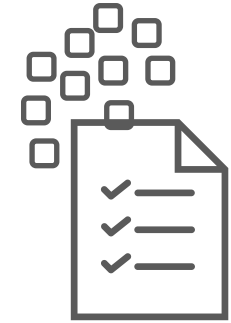SageMaker manages workflow executions and data dependencies, and provides key metrics

Mange Models

Catalogue and manage model versions in the SagaMaker model registry

Automate and Scale MLOps

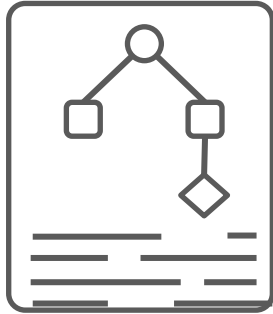Fully managed CI/CO; inbuilt MLOPs templates and support for custom templates
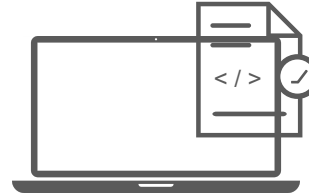
Track Lineage

Inbuilt support for tracking data, code, and model artifact lineage

Easier collaboration between data scientists, ML engineers, DevOps/MLOps; acceleration of ML projects

CAPABILITIES OF SageMaker PIPELINES
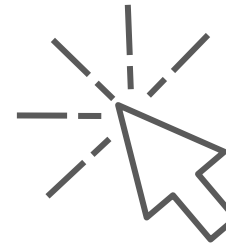
Build ML Workflows

Python SDK
Build workflows comprising parameters, steps and data dependencies

Orchestrate SageMaker Jobs
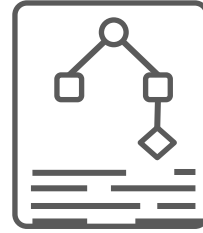Data Wrangler, processing, training, batch transform and model registration

Trigger Executions
Run pipeline from SageMaker Studio, notebooks, SDK and automate executions
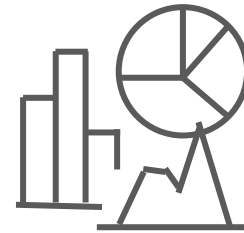
TROUBLESHOOT ML WORKFLOWS

Troubleshoot ML Workflows

Visualizing Pipelines
Visualize pipeline executions and status in real time in SagaMaker Studio

Tracking Real-time Metrics
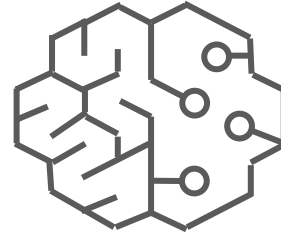Visualize real-time metrics for pipeline steps, e.g., training metrics and logs

Troubleshooting
Stop pipeline executions, cache steps and re-run pipelines
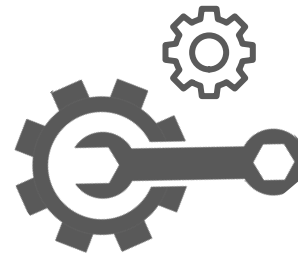
MANAGE MODELS

Manage Models

Amazon SageMaker Model Registry
Catalogue models and model versions, register models from the pipeline and save metadata and evolution metrics
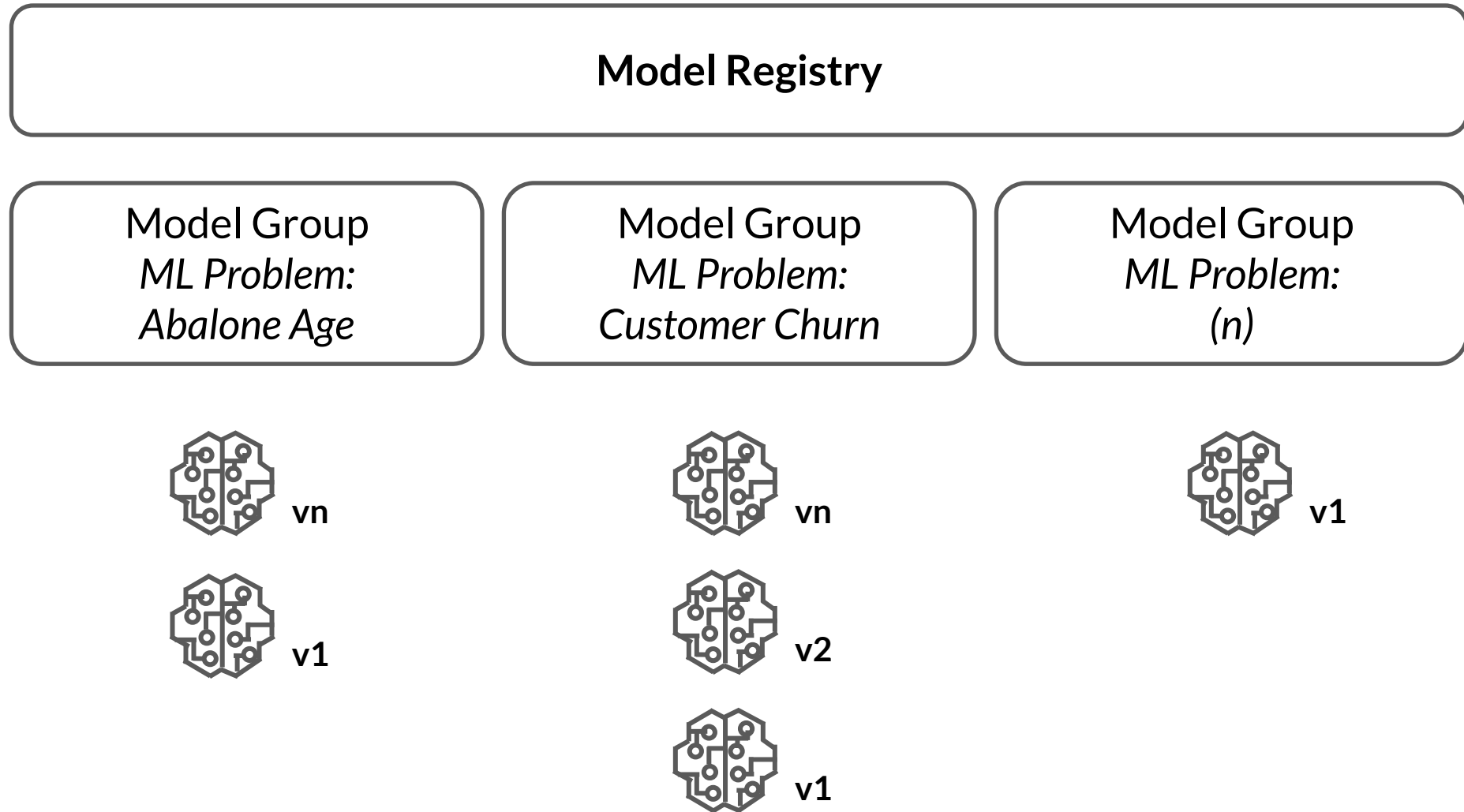
Inbuilt Approval Workflow
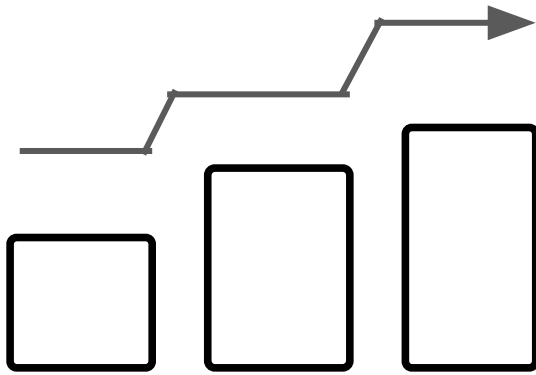Approve/reject model versions manually or automatically in the pipeline execution

Multi-Account Support
Apply resource policies to deploy model versions in different AWS accounts
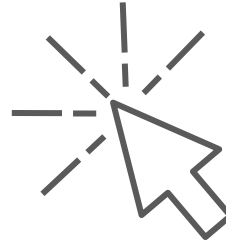
MODEL REGISTRY STRUCTURE

Model Registry

Model Group ML Problem: Abalone Age	Model Group ML Problem: Customer Churn	Model Group ML Problem: (n)

vn

v1

vn

v2

v1

v1

SCALING MLOPS

**Automate and scale MLOps**

MLOps Made Easy With SageMaker Projects
Create a project in SageMaker Studio; get code repository, seed code and the MLOps infrastructure set up for you

Inbuilt and Custom MLOps Templates
MLOps templates for building, deploying, and end-to-end workflows published by SageMaker; Provides support for custom-built templates published in Aws service Catalog
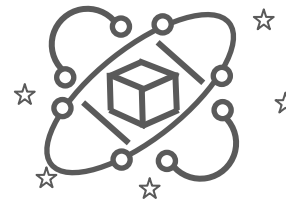
Multi-Account Support
Apply resource policies to deploy model versions in different AWS accounts
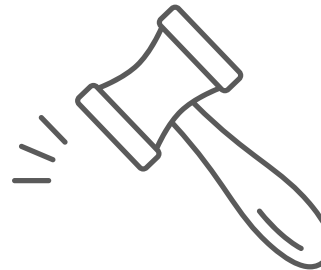
LINEAGE TRACKING

Track Lineage

End-to-End Lineage Tracking
Inbuilt lineage tracking for SageMaker pipelines; tracking data, models and artifacts for SageMaker pipelines

Query Support
View lineage in SageMaker Studio or query via SageMaker SDK: artifact, context, action, association

BYO Lineage
Track custom entities

SageMaker ENDPOINTS

- You can create an endpoint using the endpoint configuration that is created using the CreateEndpointConfig API

- SageMaker uses the endpoint to provision resources and deploy models

- The endpoint name must be unique within an AWS region in your AWS account

MODEL MONITORING IN SageMaker

- Model Monitor provides the following types of monitoring

  - Monitoring data quality

  - Monitoring model quality

  - Monitoring bias drift for models in production

- Set alerts using CloudWatch when there are deviations in the model quality

- Early detection of these deviations enables you to take corrective actions

  - Retraining models

  - Auditing upstream systems

  - Fixing quality issues without having to monitor models manually or build additional tooling
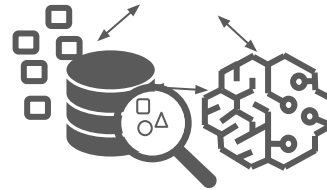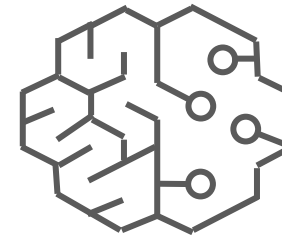
MLOps SUMMARY
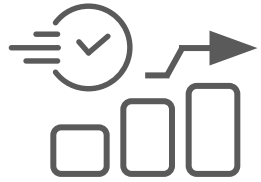
Start small and iterate
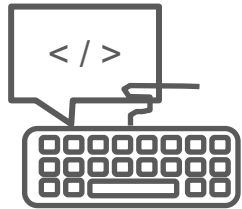
Practise Everything-as-Code for repeatability, consistency and governance

Consider ML-specific inputs, outputs and artifacts for model and pipeline lineage

Use integrated SageMaker services when possible to reduce management and set-up time

Implement a Model Registry as part of your ML pipeline for managing models

SESSION SUMMARY

- Created an end-to-end pipeline with the following steps

  - Pre-process

  - Train

  - Evaluate

  - Register

  - Deploy

  - Infer