SageMaker MLOps

Paul Hargis
AWS Solutions Architect
AI/ML Specialist

Pranav Murthy
AWS Solutions Architect
AI/ML Specialist

Agenda

Exploratory Data Analysis

Model Training and Model Tuning

Inference and Hosting

Operationalize Your Workflows

Flexible Exploratory Data Analysis

Exploratory Data Analysis

Amazon SageMaker Data Wrangler

A faster, visual way to aggregate and prepare data for machine learning

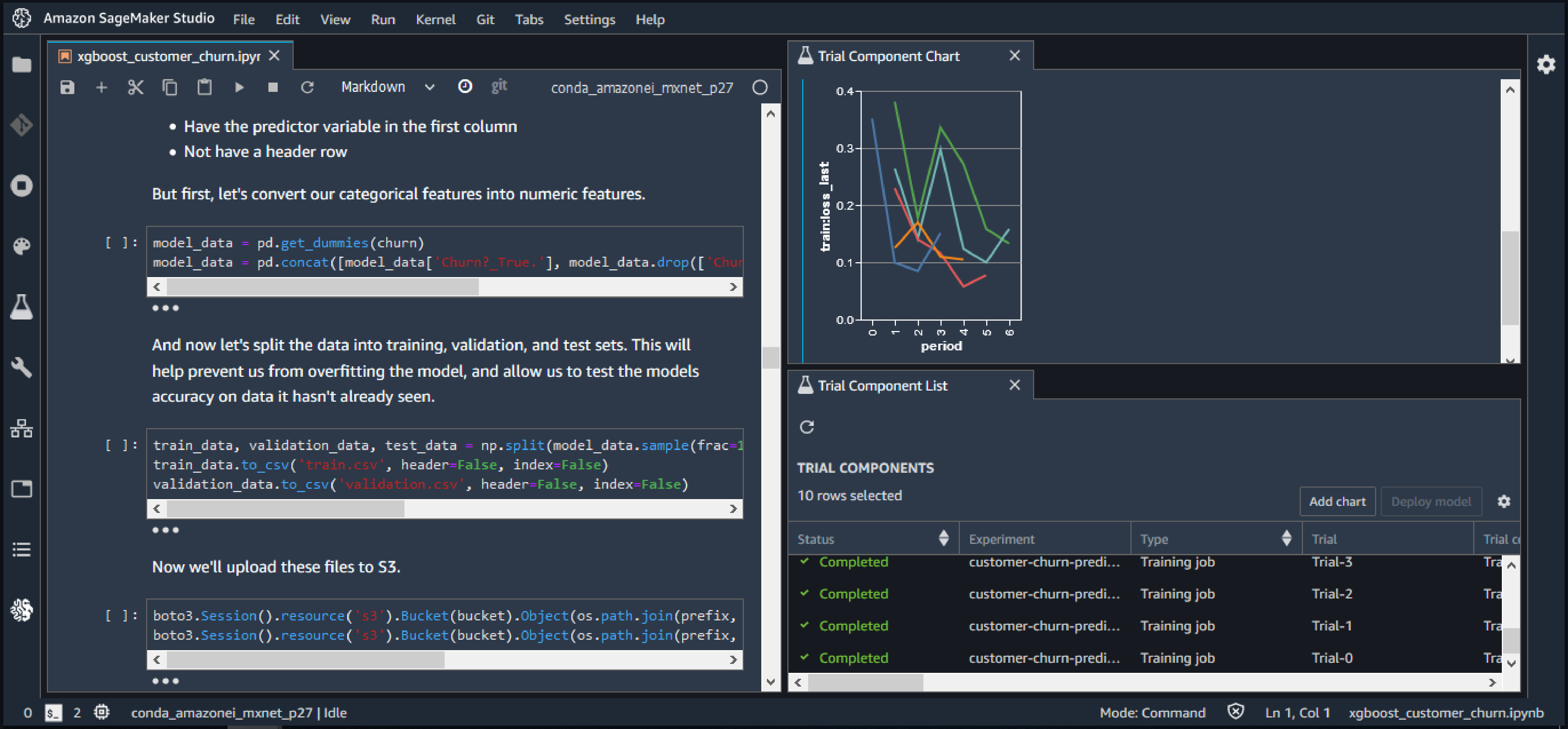
SageMaker Studio Notebook EDA

Locally Prepare and Analyze your data using SageMaker Studio Notebook

SageMaker Processing Job

To analyze data and evaluate machine learning models on Amazon SageMaker, use Amazon SageMaker Processing

Amazon SageMaker Studio to Quickly Analyze Data and Build Models



- Have the predictor variable in the first column
- Not have a header row

But first, let's convert our categorical features into numeric features.

```
[ ]: model_data = pd.get_dummies(churn)
     model_data = pd.concat([model_data['Churn?_True.'], model_data.drop(['Chur
```

And now let's split the data into training, validation, and test sets. This will help prevent us from overfitting the model, and allow us to test the models accuracy on data it hasn't already seen.

```
[ ]: train_data, validation_data, test_data = np.split(model_data.sample(frac=1
     train_data.to_csv('train.csv', header=False, index=False)
     validation_data.to_csv('validation.csv', header=False, index=False)
```

Now we'll upload these files to S3.

```
[ ]: boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
     boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
```

TRIAL COMPONENTS

10 rows selected

Status	Experiment	Type	Trial	Trial c
✓ Completed	customer-churn-predi...	Training job	Trial-3	Tra
✓ Completed	customer-churn-predi...	Training job	Trial-2	Tra
✓ Completed	customer-churn-predi...	Training job	Trial-1	Tra
✓ Completed	customer-churn-predi...	Training job	Trial-0	Tra

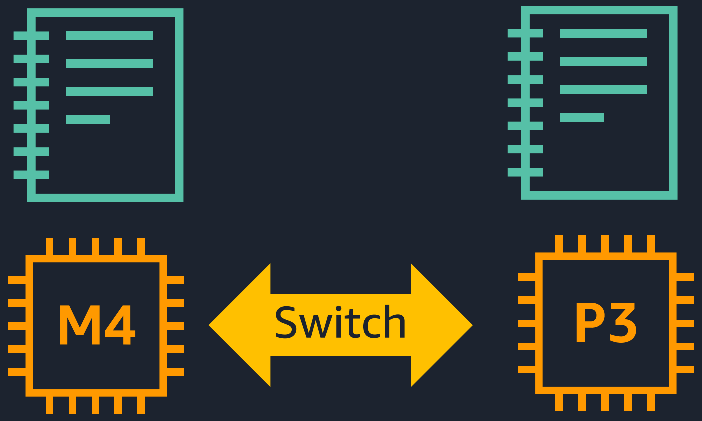
0 2 conda_amazonei_mxnet_p27 | Idle   Mode: Command   Ln 1, Col 1   xgboost_customer_churn.ipynb

aws machine learning

Fast-start, shareable notebooks

- No wrangling of environments
- Start-up new notebooks in < 30 seconds
- One-click sharing of notebook and all its dependencies
- Easily switch instance types to match workload

- Exploratory data analysis
- Coding new algorithm
- Analyzing results



Training and testing deep neural networks

Enables notebook cost reduction of ~50%

Amazon SageMaker Processing

Managed solution for data processing and model evaluation jobs

Fully managed

Fully managed clusters for distributed processing

Bring your own custom processing script

Bring your own script for feature engineering, data validation, model evaluation, and model interpretation

Use SageMaker containers or BYO framework containers

Use a SageMaker optimized framework container or bring your own

Pay-as-you-go

Infrastructure is created on-demand, configured, and terminated automatically

Security and compliance

All of the same security and compliance features

Flexible Model Training and Tuning At Scale

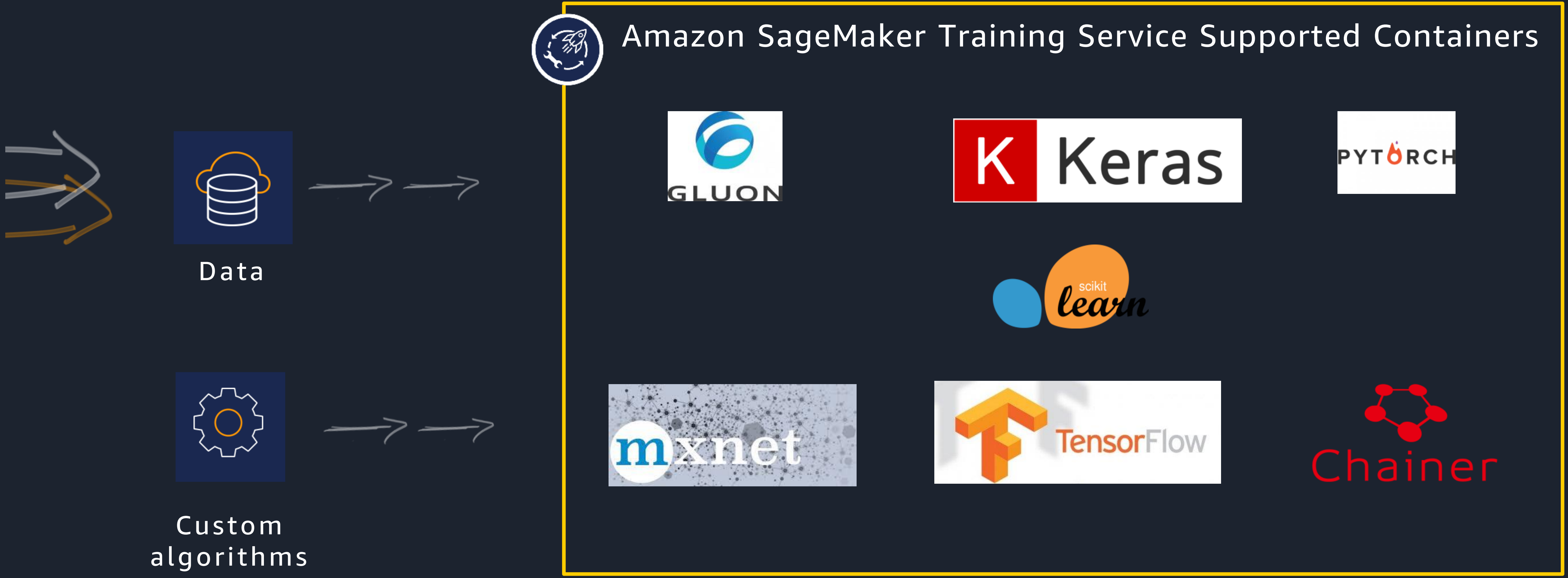
Amazon SageMaker – Built-in Algorithms

Rethinking algorithms design for large scale & streaming data

K-Means XGBoost PCA
Linear Learner Seq2Seq
DeepAR
Factorization Machines
Object Detection
BlazingText
Image Classification
K-Nearest Neighbor
LDA Neural Topic Model
Random Cut Forest
Object2Vec IP Insight
Semantic Segmentation

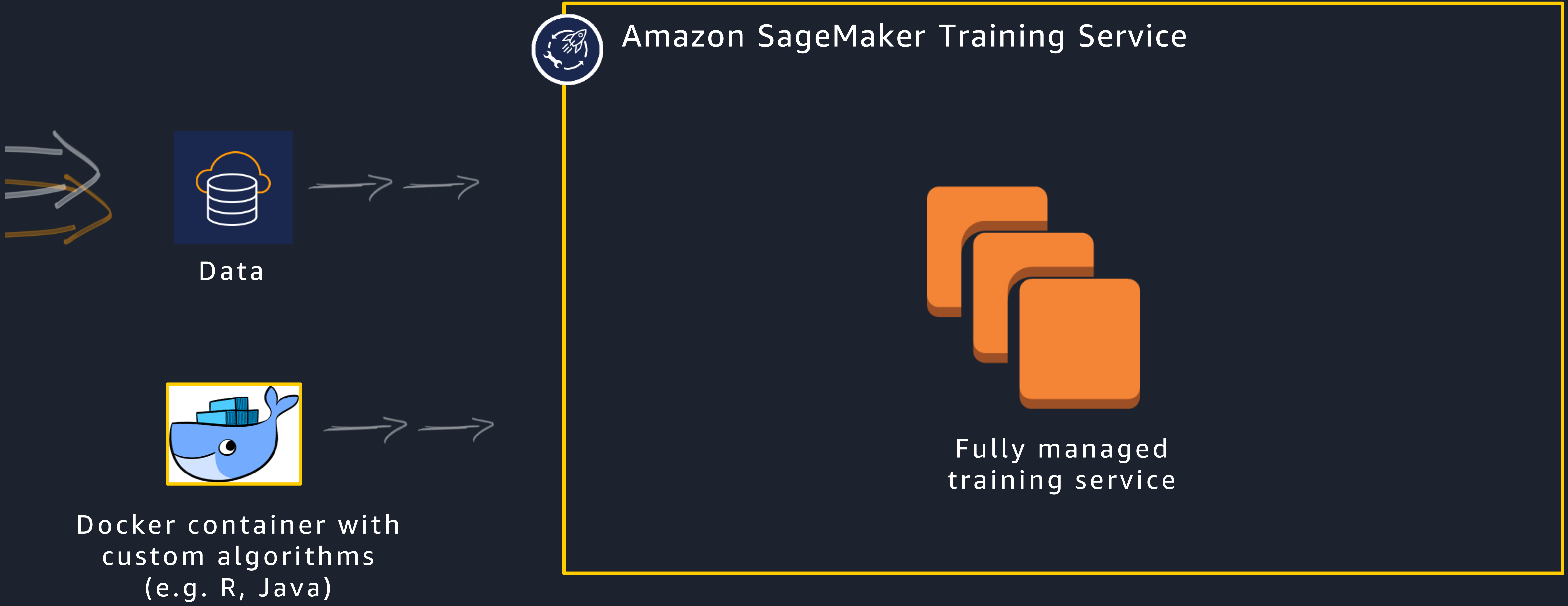
Amazon SageMaker – Bring-Your-Own-Algorithm

Build your own algorithms, SageMaker handles the rest



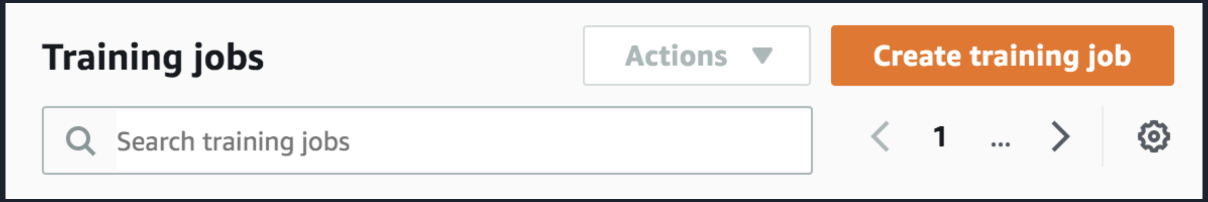
Amazon SageMaker – Bring-Your-Own-Container

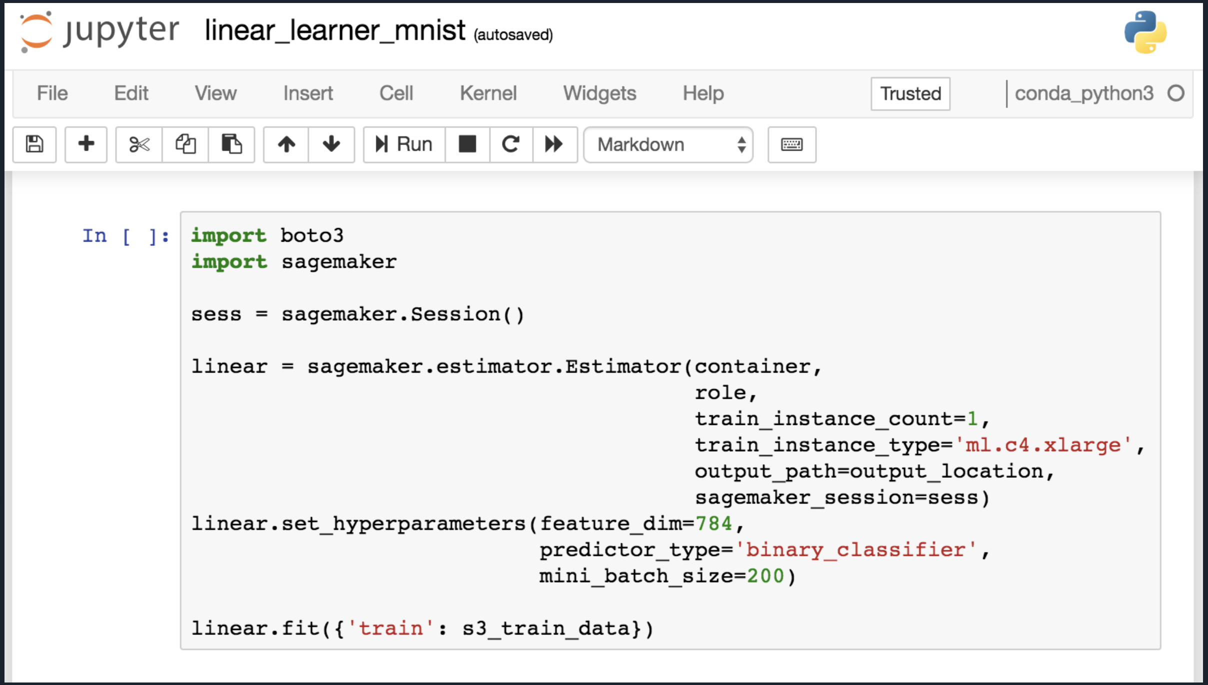
Bring your custom code and container, train at scale in SageMaker


Amazon SageMaker Training Service
Data
Fully managed training service
Docker container with custom algorithms (e.g. R, Java)

Launching Training Jobs from Jupyter

Launch Training

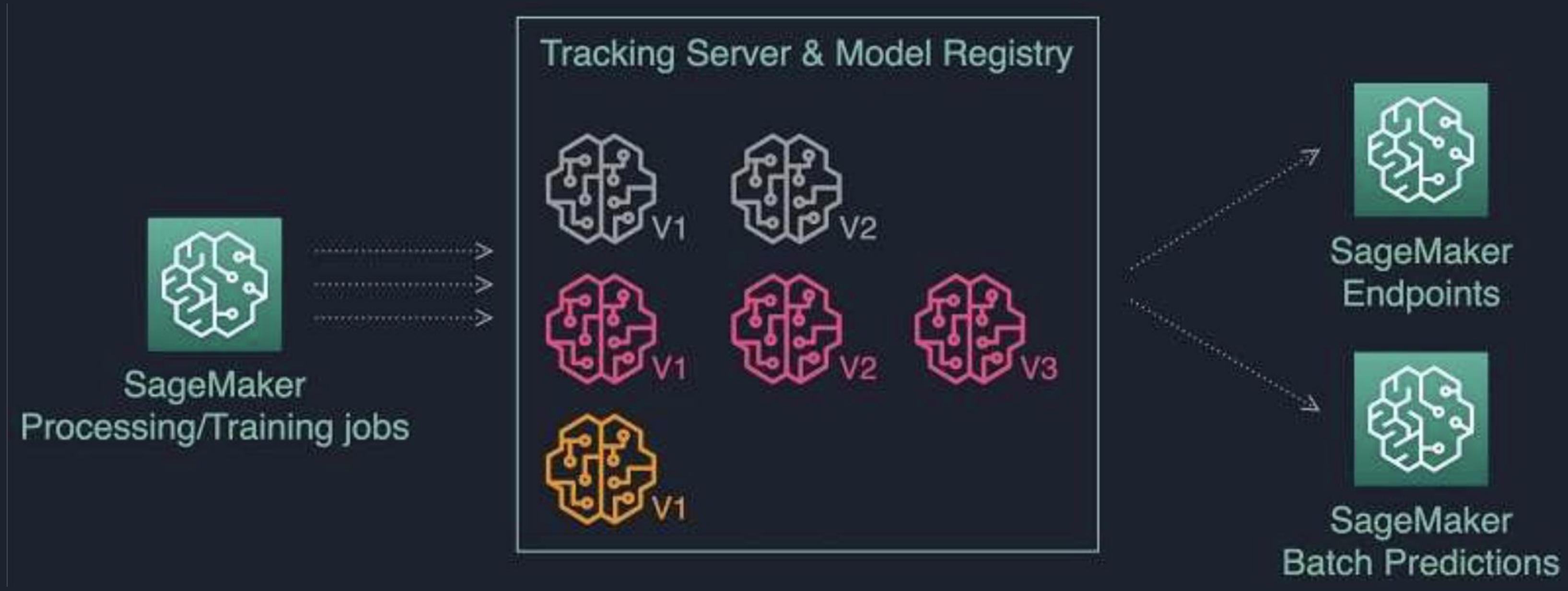



jupyter linear_learner_mnist (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
conda_python3
Run
Markdown

```
In [ ]: import boto3
        import sagemaker

        sess = sagemaker.Session()

        linear = sagemaker.estimator.Estimator(container,
                                               role,
                                               train_instance_count=1,
                                               train_instance_type='ml.c4.xlarge',
                                               output_path=output_location,
                                               sagemaker_session=sess)
        linear.set_hyperparameters(feature_dim=784,
                                   predictor_type='binary_classifier',
                                   mini_batch_size=200)

        linear.fit({'train': s3_train_data})
```

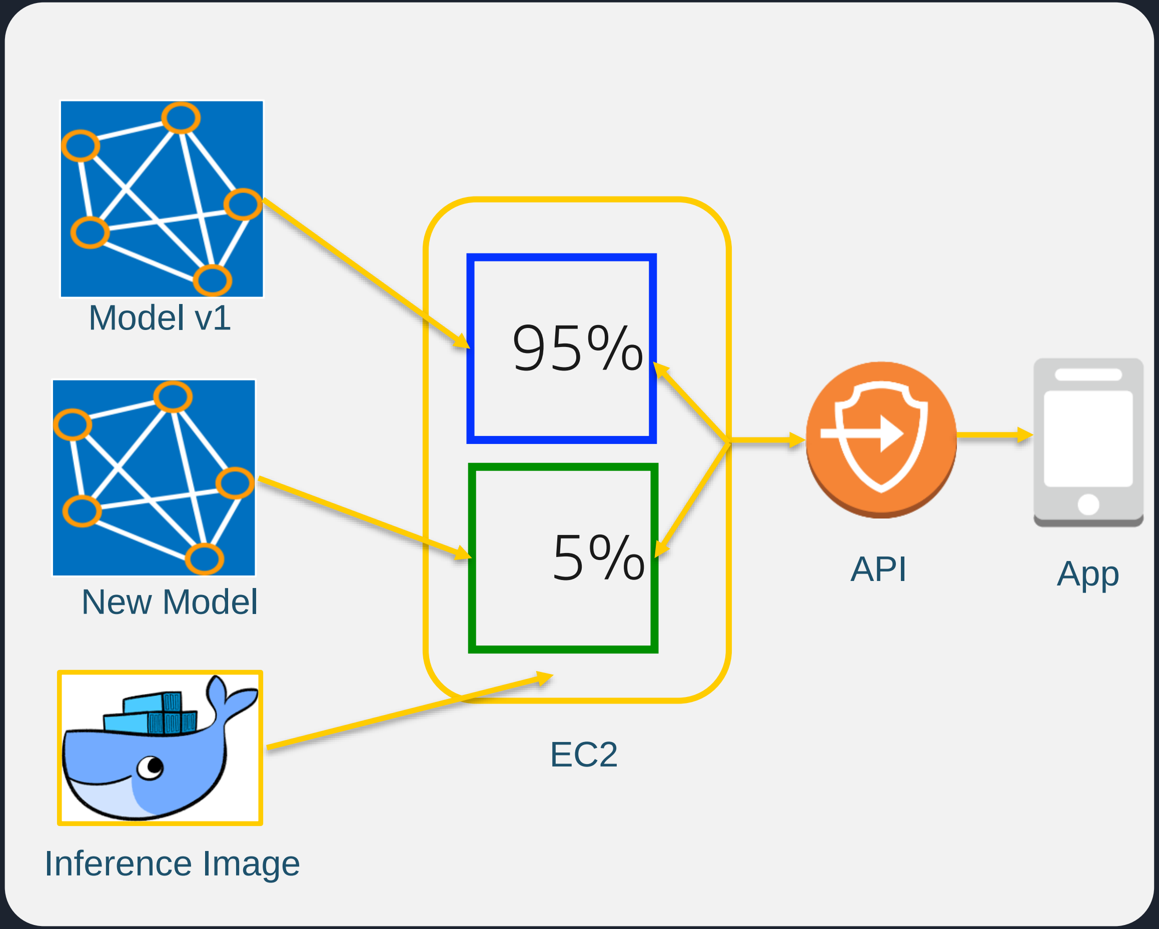
Model Registry

SageMaker Model Registry


Tracking Server & Model Registry
V1
V2
V1
V2
V3
V1
SageMaker Processing/Training jobs
SageMaker Endpoints
SageMaker Batch Predictions

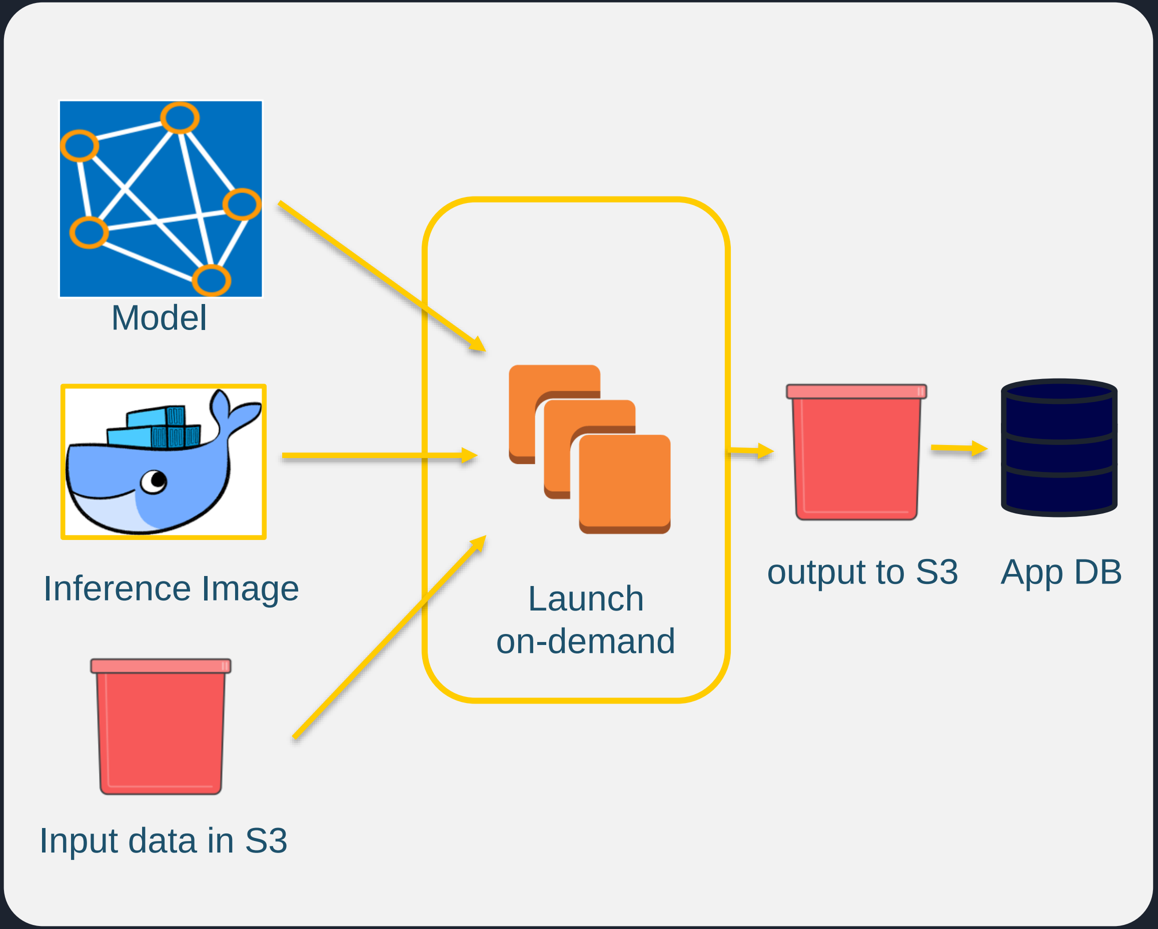
Speedy Deployment at Scale

Amazon SageMaker – Real time inference



- Quick deployment of Restful API endpoint
- Autoscaling across multiple AZs for variable loads
- A/B testing support
- VPC Privatelink support
- Endpoint invocation and resource usage metrics
- No model re-engineering

Amazon SageMaker – Batch inference



- Use trained model to get inference on data in S3
- Launch fully managed transient inference resources
- Seamless integration with S3 for both input and output

Operationalize Your Workflows with SageMaker Pipelines and Projects

Challenges with creating a complete workflow for the ML lifecycle

1. Building, training, and deploying models is an iterative process

2. Getting models from concept to production involves multiple steps
   - Create standard code packages for each step of the ML lifecycle
   - Stitch them together into a structure called a workflow
   - Manage dependencies between steps
   - Execute the workflow as an orchestrated sequence

3. Track artifacts for each step of the workflow

4. Deploy and manage the right version of the model, across thousands of models

5. Automate and scale the complete workflow as part of ML Ops

aws machine learning

Amazon SageMaker Pipelines

Build fully automated machine learning workflows at scale

Accelerate machine learning development

With just a few clicks, you can create a fully automated ML workflow, reducing months of coding to a few hours

Automatically track hundreds of model artifacts

Eliminate manual processes by automatically tracking model artifacts to keep a structured audit trail

Scale to thousands of ML models in production

Choose from built-in templates to setup CI/CD pipelines to automate workflows and deploy ML models at scale

Key Features

Amazon SageMaker Pipelines

Compose and manage ML workflows
Create detailed workflows with an easy-to-use Python SDK and manage them visually

Track model lineage for governance and audits
Track code, datasets, and versions for each step of the ML lifecycle

Replay and re-run workflows
Keep models up to date by re-running every step based on custom schedules

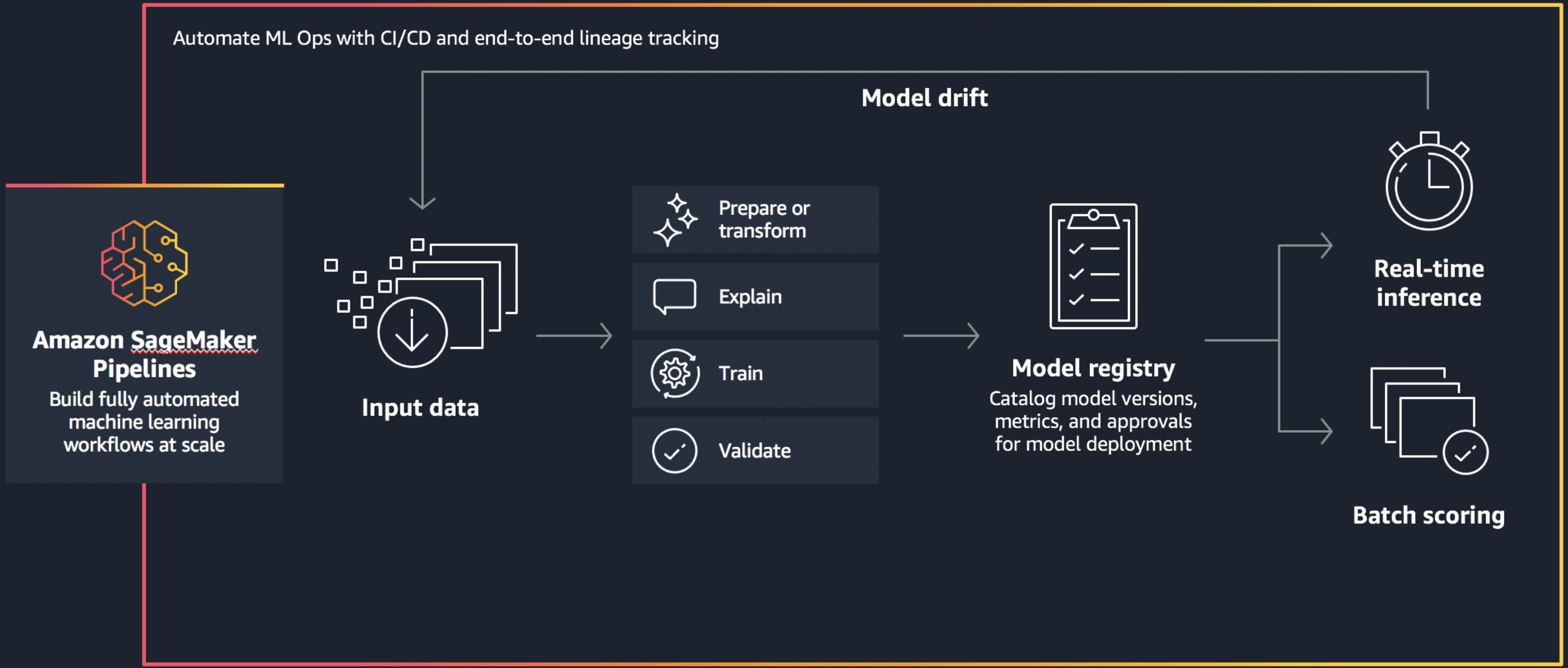
Visually compare, select, and deploy models
Deploy and manage modules through the visual interface of SageMaker Studio

Access a central registry of trained models
Use the model registry to choose your best model for deployment to production

Fully managed ML Ops with built-in CI/CD support
Use CI/CD practices to build a fully automated machine learning workflow

aws machine learning

Amazon SageMaker Pipelines - How it works


Automate ML Ops with CI/CD and end-to-end lineage tracking
Model drift
Amazon SageMaker Pipelines
Build fully automated machine learning workflows at scale
Input data
Prepare or transform
Explain
Train
Validate
Model registry
Catalog model versions, metrics, and approvals for model deployment
Real-time inference
Batch scoring

aws machine learning

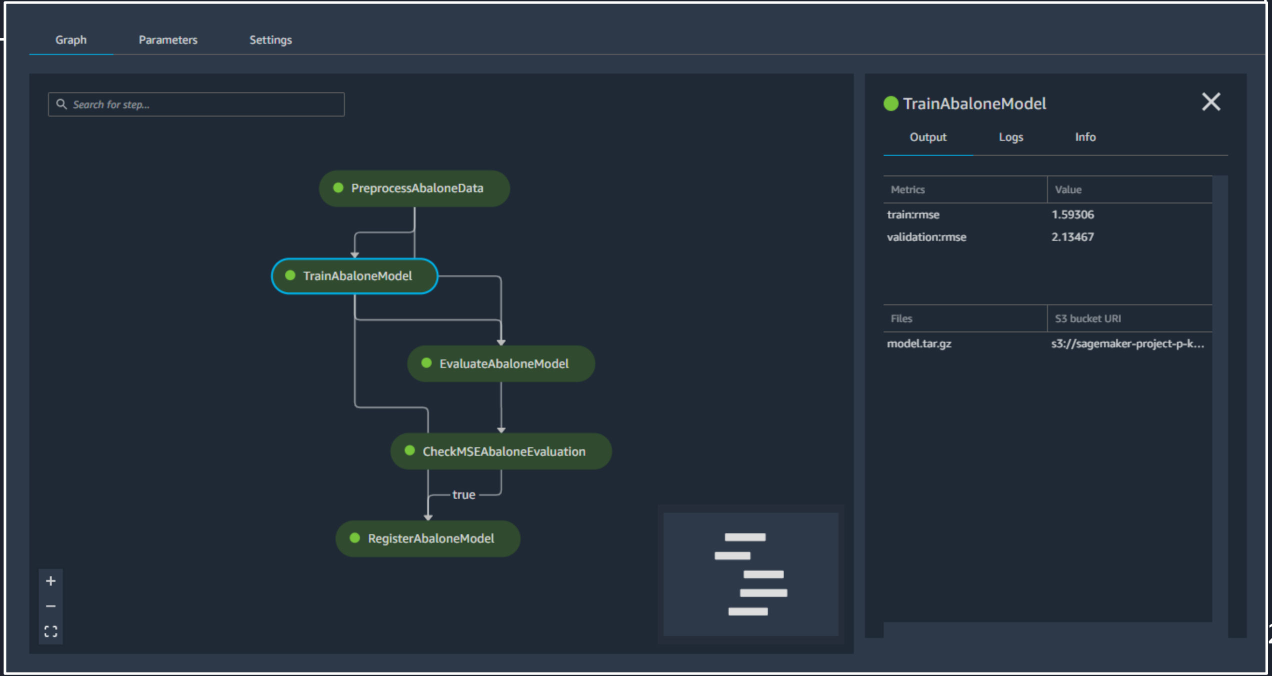
Compose & manage workflows

Create your ML workflows using the Python SDK

Visualize the workflows with SageMaker Studio

Manage every step including data transformations, training, debugging, and optimizing models

```
class MyPipeline(dsl.Pipeline):
  # Pipeline parameter definitions
  training_instance = PipelineParam(string)
  num_trees = PipelineParam(int, default=100)
  tree_depth = PipelineParam(int, default=3)

# Example data query step
get_input_data_step = AthenaStep(query="SELECT a,b,c FROM
table_xyz”)

# Example training step
training_step = XGBoostStep(
  entry_point='scripts/xgb/train.py’,
  source_dir='scripts/xgb’,
  instance_type=training_instance,
  env_vars=[num_trees, tree_depth],
  inputs={'train': processing_step.outputs(‘clean_data')})
```



Metrics	Value
train:rmse	1.59306
validation:rmse	2.13467

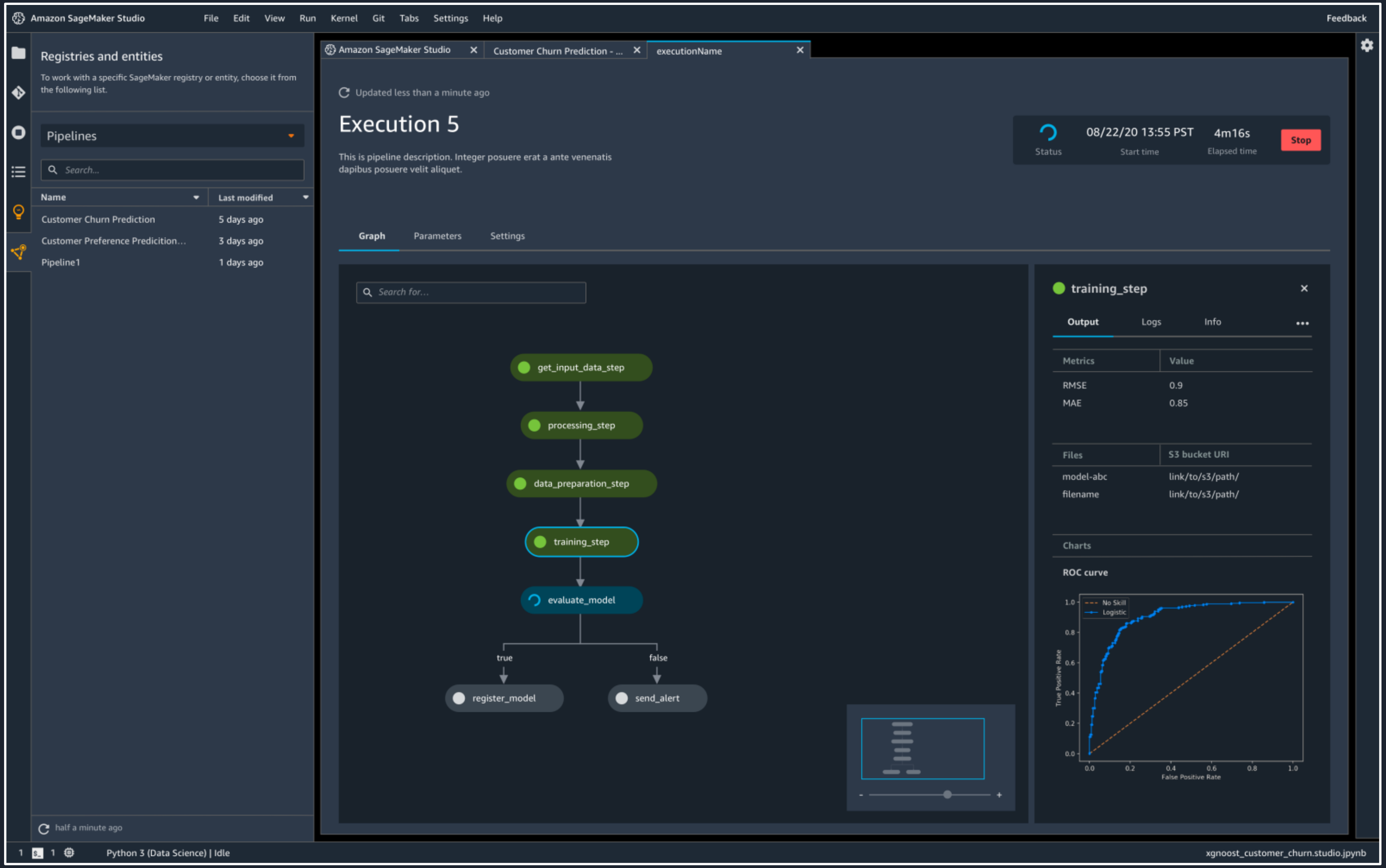
Files	S3 bucket URI
model.tar.gz	s3://sagemaker-project-p-k...

Visualize status of model pipelines

Follow completed steps and monitor steps in progress

Understand the output from each step with the output logs

Monitor, change, and manage the parameters for each step



To work with a specific SageMaker registry or entity, choose it from the following list.

Name	Last modified
Customer Churn Prediction	5 days ago
Customer Preference Predicition...	3 days ago
Pipeline1	1 days ago

Execution 5

08/22/20 13:55 PST 4m16s

This is pipeline description. Integer posuere erat a ante venenatis dapibus posuere velit aliquet.

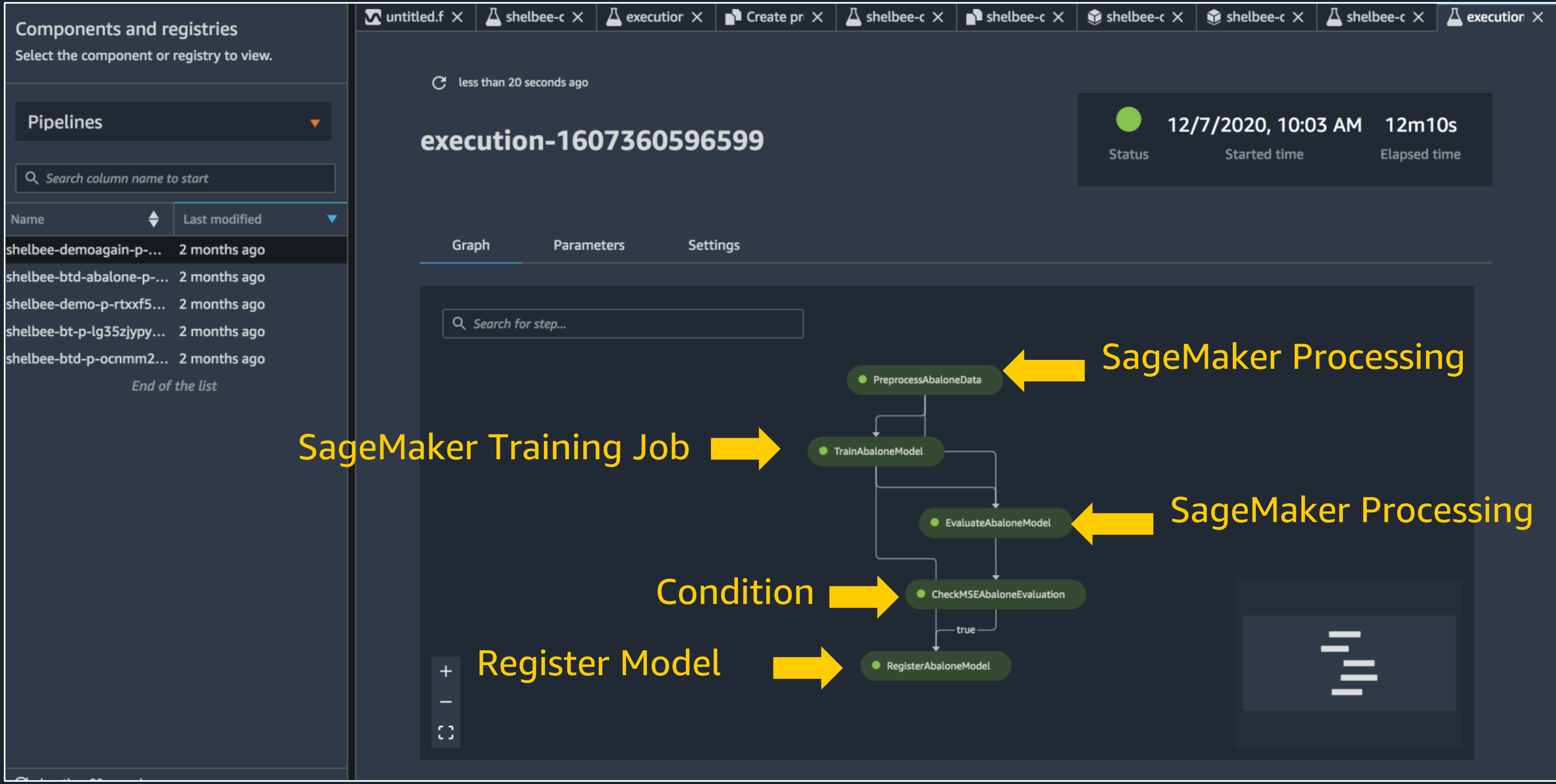
training_step

Metrics	Value
RMSE	0.9
MAE	0.85

Files	S3 bucket URI
model-abc	link/to/s3/path/
filename	link/to/s3/path/

Create workflows with built-in Steps

- Processing
- Training
- Condition
- BatchTransform
- RegisterModel
- CreateModel



Components and registries
Select the component or registry to view.

Pipelines

Name	Last modified
shelbee-demoagain-p-...	2 months ago
shelbee-btd-abalone-p-...	2 months ago
shelbee-demo-p-rtxxf5...	2 months ago
shelbee-bt-p-lg35zjypy...	2 months ago
shelbee-btd-p-ocnmm2...	2 months ago

End of the list

less than 20 seconds ago

execution-1607360596599

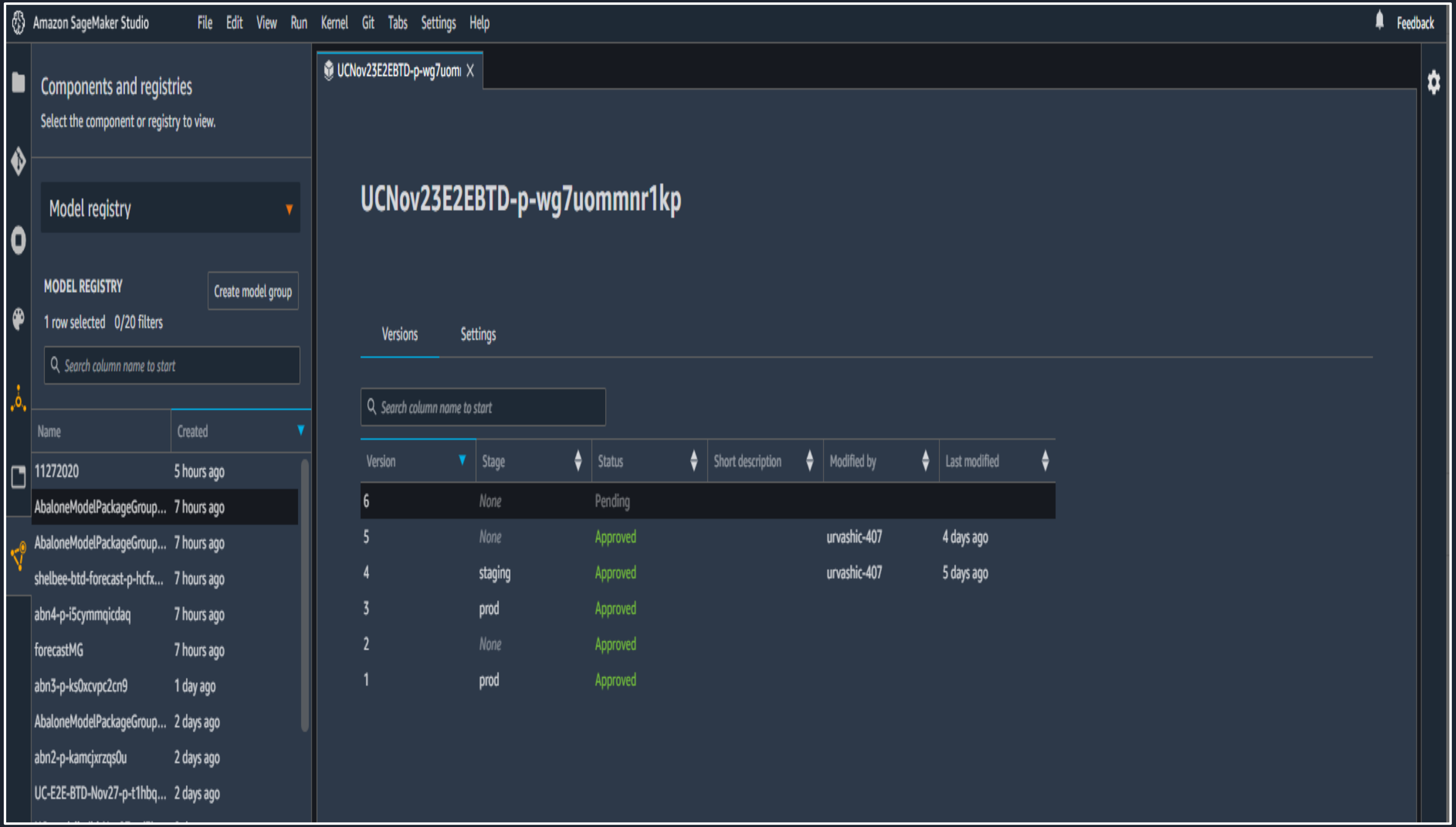
Status: ● | Started time: 12/7/2020, 10:03 AM | Elapsed time: 12m10s

Choose models from the model registry

Model registry is a central repository of trained models

Register modes intended for production in the model registry

Access the model registry through SageMaker Studio or the Python SDK



Components and registries

Select the component or registry to view.

Model registry

MODEL REGISTRY

Create model group

1 row selected 0/20 filters

Search column name to start

Name	Created
11272020	5 hours ago
AbaloneModelPackageGroup...	7 hours ago
AbaloneModelPackageGroup...	7 hours ago
shelbee-btd-forecast-p-hcfx...	7 hours ago
abn4-p-i5cymmqicdaq	7 hours ago
forecastMG	7 hours ago
abn3-p-ks0xcvpc2cn9	1 day ago
AbaloneModelPackageGroup...	2 days ago
abn2-p-kamcjxrzqs0u	2 days ago
UC-E2E-BTD-Nov27-p-t1hbq...	2 days ago

UCNov23E2EBTD-p-wg7uommnr1kp

Versions Settings

Search column name to start

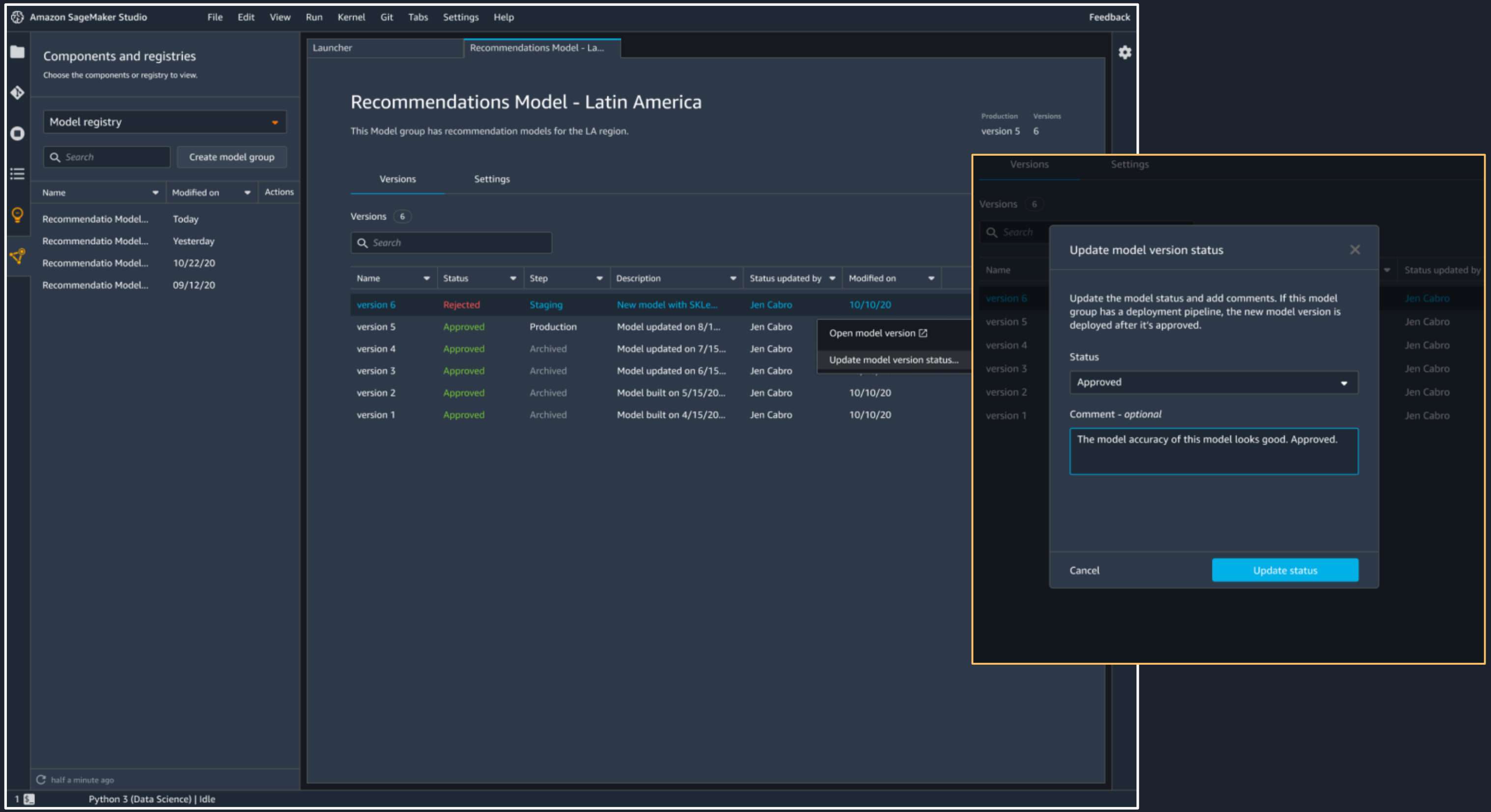
Version	Stage	Status	Short description	Modified by	Last modified
6	*None*	Pending			
5	*None*	Approved		urvashic-407	4 days ago
4	staging	Approved		urvashic-407	5 days ago
3	prod	Approved			
2	*None*	Approved			
1	prod	Approved			

Execute CI/CD pipelines to update models

Use CI/CD practices to automate deployment

Operationalize machine learning at scale

Keep your models up to date and accurate



Components and registries

Choose the components or registry to view.

Model registry

Name	Modified on	Actions
Recommendatio Model...	Today	
Recommendatio Model...	Yesterday	
Recommendatio Model...	10/22/20	
Recommendatio Model...	09/12/20	

Recommendations Model - Latin America

This Model group has recommendation models for the LA region.

Production	Versions
version 5	6

Versions 6

Name	Status	Step	Description	Status updated by	Modified on
version 6	Rejected	Staging	New model with SKLe...	Jen Cabro	10/10/20
version 5	Approved	Production	Model updated on 8/1...	Jen Cabro	
version 4	Approved	Archived	Model updated on 7/15...	Jen Cabro	
version 3	Approved	Archived	Model updated on 6/15...	Jen Cabro	
version 2	Approved	Archived	Model built on 5/15/20...	Jen Cabro	10/10/20
version 1	Approved	Archived	Model built on 4/15/20...	Jen Cabro	10/10/20

Update model version status

Update the model status and add comments. If this model group has a deployment pipeline, the new model version is deployed after it's approved.

Status: Approved

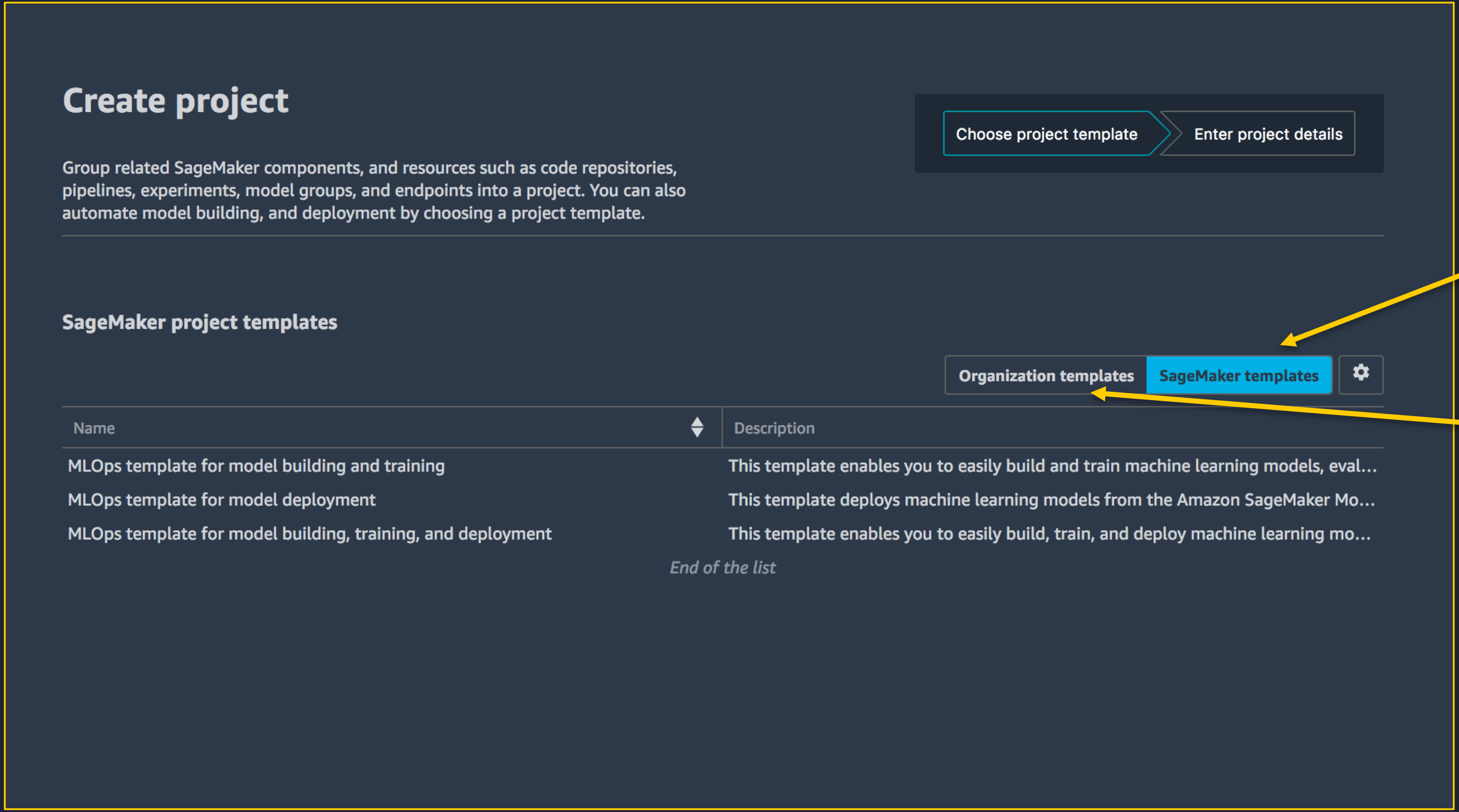
Comment - *optional*: The model accuracy of this model looks good. Approved.

aws machine learning

Execute CI/CD pipelines to update models

Projects

Project Templates



Create project

Group related SageMaker components, and resources such as code repositories, pipelines, experiments, model groups, and endpoints into a project. You can also automate model building, and deployment by choosing a project template.

SageMaker project templates

Name	Description
MLOps template for model building and training	This template enables you to easily build and train machine learning models, eval…
MLOps template for model deployment	This template deploys machine learning models from the Amazon SageMaker Mo…
MLOps template for model building, training, and deployment	This template enables you to easily build, train, and deploy machine learning mo…

End of the list

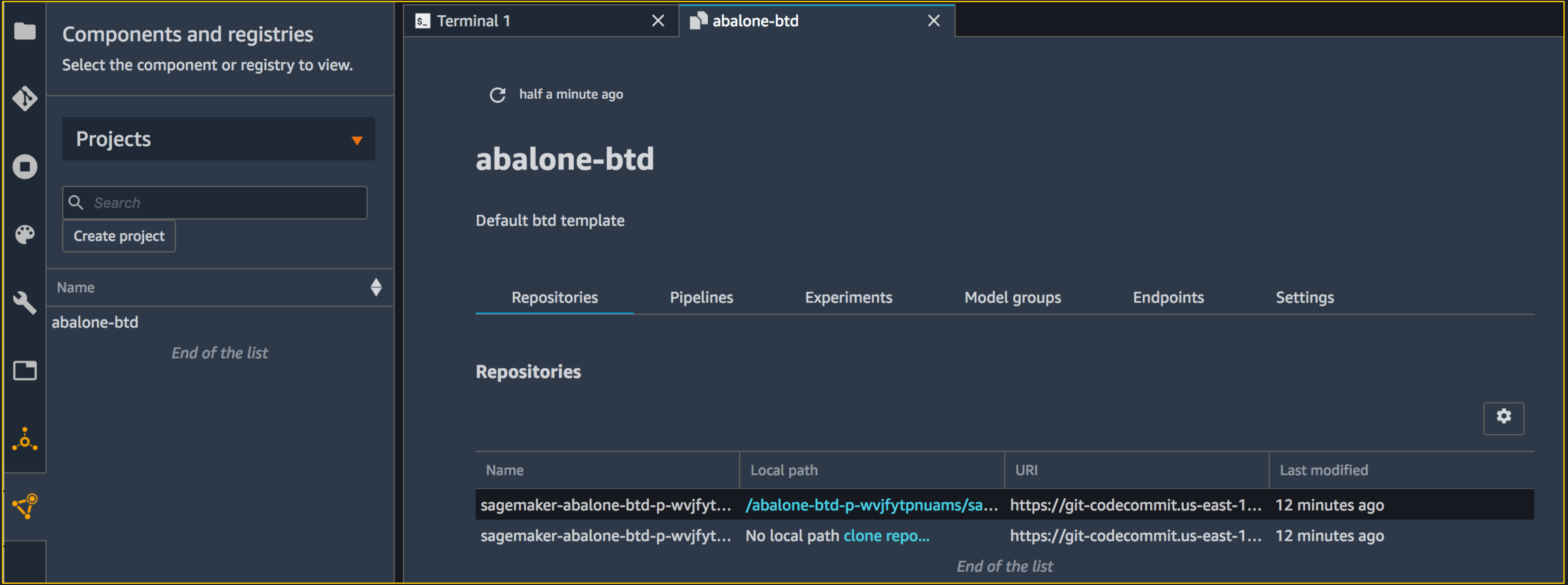
Built-In Templates

Custom Templates

Execute CI/CD pipelines to update models

Projects

Integrate with source control



Components and registries

Select the component or registry to view.

Projects

Search

Create project

Name
abalone-btd

End of the list

half a minute ago

abalone-btd

Default btd template

Repositories

Name	Local path	URI	Last modified
sagemaker-abalone-btd-p-wvjfyt...	/abalone-btd-p-wvjfytpnuams/sa...	https://git-codecommit.us-east-1...	12 minutes ago
sagemaker-abalone-btd-p-wvjfyt...	No local path clone repo...	https://git-codecommit.us-east-1...	12 minutes ago

End of the list

Thank you

aws machine learning