Build, train, and deploy ML models with Amazon SageMaker

Jonathan Dion
Senior Technical Evangelist
Amazon Web Services
@jotdion
linkedin.com/in/jotdion

Agenda

- The AWS ML Stack
- ML Services overview
- Labs

- What we’ll cover today:
  - Preparing our data set
  - Training and deploying with built-in algorithms
  - Finding optimal hyperparameters with automatic model tuning
  - Deploying multiple models for A/B testing

Our mission at AWS

Put machine learning in the
hands of every developer

MACHINE LEARNING IS HAPPENING IN COMPANIES OF EVERY SIZE AND INDUSTRY

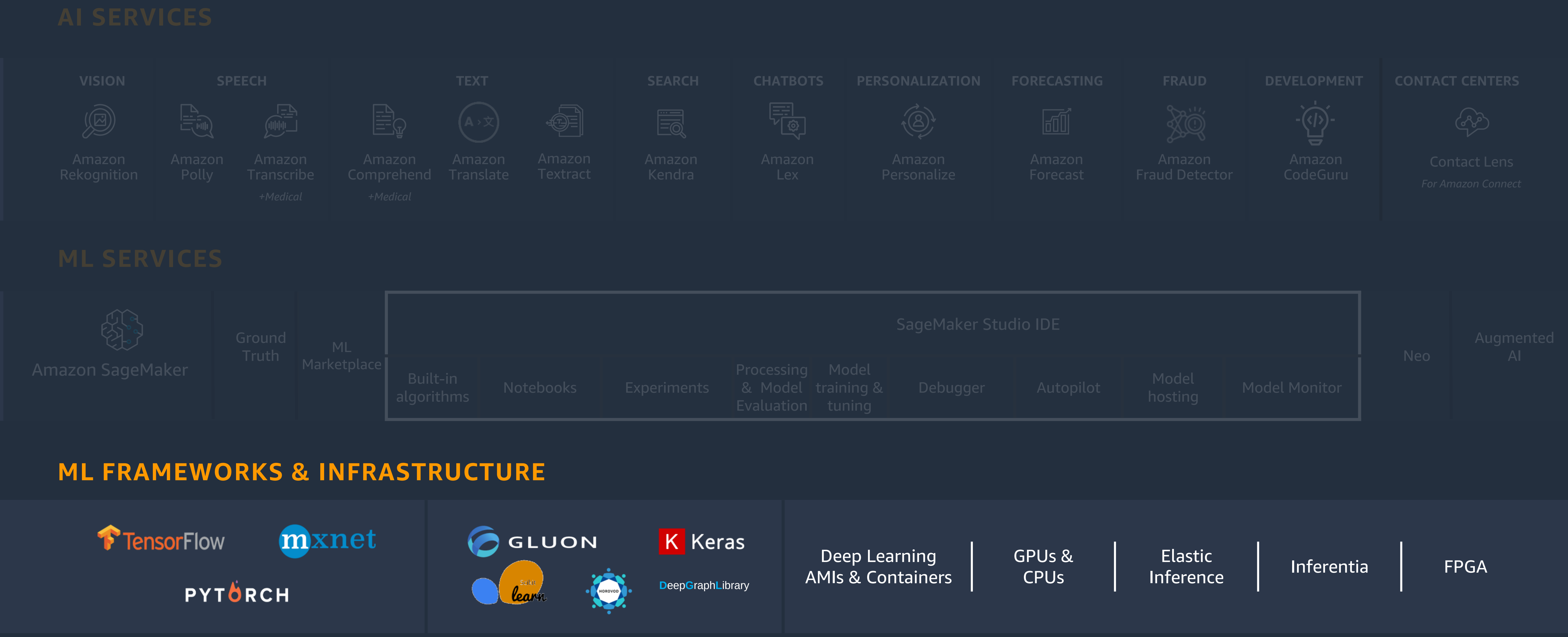
Tens of thousands customers have chosen AWS for their ML workloads | More than twice as many customers using ML than any other cloud provider

© 2020, Amazon Web Services, Inc. or its Affiliates.

aws

The AWS ML Stack

Broadest and most complete set of Machine Learning capabilities



AI SERVICES

VISION: Amazon Rekognition

SPEECH: Amazon Polly, Amazon Transcribe *+Medical*

TEXT: Amazon Comprehend *+Medical*, Amazon Translate, Amazon Textract

SEARCH: Amazon Kendra

CHATBOTS: Amazon Lex

PERSONALIZATION: Amazon Personalize

FORECASTING: Amazon Forecast

FRAUD: Amazon Fraud Detector

DEVELOPMENT: Amazon CodeGuru

CONTACT CENTERS: Contact Lens *For Amazon Connect*

ML SERVICES

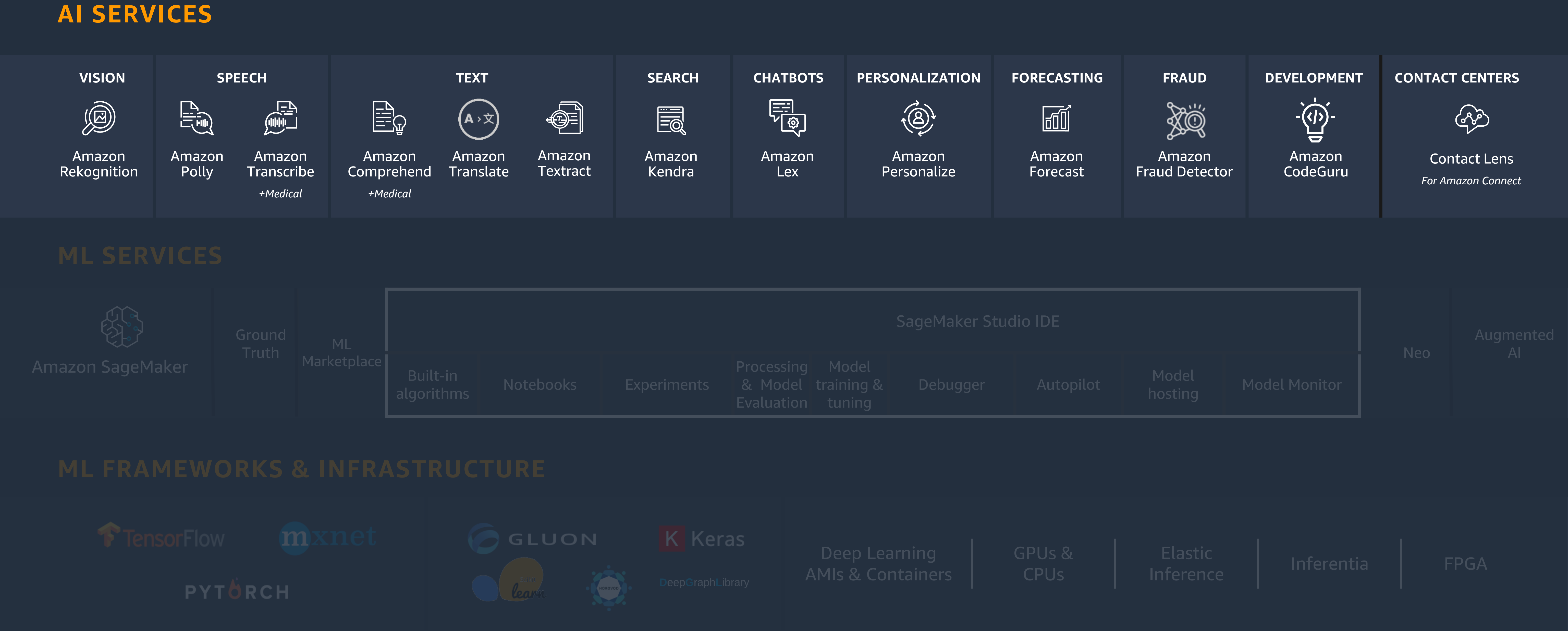
Amazon SageMaker | Ground Truth | ML Marketplace | SageMaker Studio IDE: Built-in algorithms, Notebooks, Experiments, Processing & Model Evaluation, Model training & tuning, Debugger, Autopilot, Model hosting, Model Monitor | Neo | Augmented AI

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow, mxnet, PYTORCH | GLUON, Keras, scikit learn, HOROVOD, DeepGraphLibrary | Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia | FPGA

The AWS ML Stack

Broadest and most complete set of Machine Learning capabilities



AI SERVICES

VISION
- Amazon Rekognition

SPEECH
- Amazon Polly
- Amazon Transcribe *+Medical*

TEXT
- Amazon Comprehend *+Medical*
- Amazon Translate
- Amazon Textract

SEARCH
- Amazon Kendra

CHATBOTS
- Amazon Lex

PERSONALIZATION
- Amazon Personalize

FORECASTING
- Amazon Forecast

FRAUD
- Amazon Fraud Detector

DEVELOPMENT
- Amazon CodeGuru

CONTACT CENTERS
- Contact Lens *For Amazon Connect*

ML SERVICES

Amazon SageMaker

Ground Truth

ML Marketplace

SageMaker Studio IDE

Built-in algorithms | Notebooks | Experiments | Processing & Model Evaluation | Model training & tuning | Debugger | Autopilot | Model hosting | Model Monitor

Neo

Augmented AI

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow | mxnet | PYTORCH | GLUON | Keras | learn | HOROVOD | DeepGraphLibrary

Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia | FPGA

The AWS ML Stack

Broadest and most complete set of Machine Learning capabilities

AI SERVICES

VISION	SPEECH		TEXT			SEARCH	CHATBOTS	PERSONALIZATION	FORECASTING	FRAUD	DEVELOPMENT	CONTACT CENTERS
Amazon Rekognition	Amazon Polly	Amazon Transcribe *+Medical*	Amazon Comprehend *+Medical*	Amazon Translate	Amazon Textract	Amazon Kendra	Amazon Lex	Amazon Personalize	Amazon Forecast	Amazon Fraud Detector	Amazon CodeGuru	Contact Lens *For Amazon Connect*

ML SERVICES

Amazon SageMaker

Ground Truth

ML Marketplace

SageMaker Studio IDE

- Built-in algorithms
- Notebooks
- Experiments
- Processing & Model Evaluation
- Model training & tuning
- Debugger
- Autopilot
- Model hosting
- Model Monitor

Neo

Augmented AI

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow

mxnet

PYTORCH

GLUON

Keras

learn

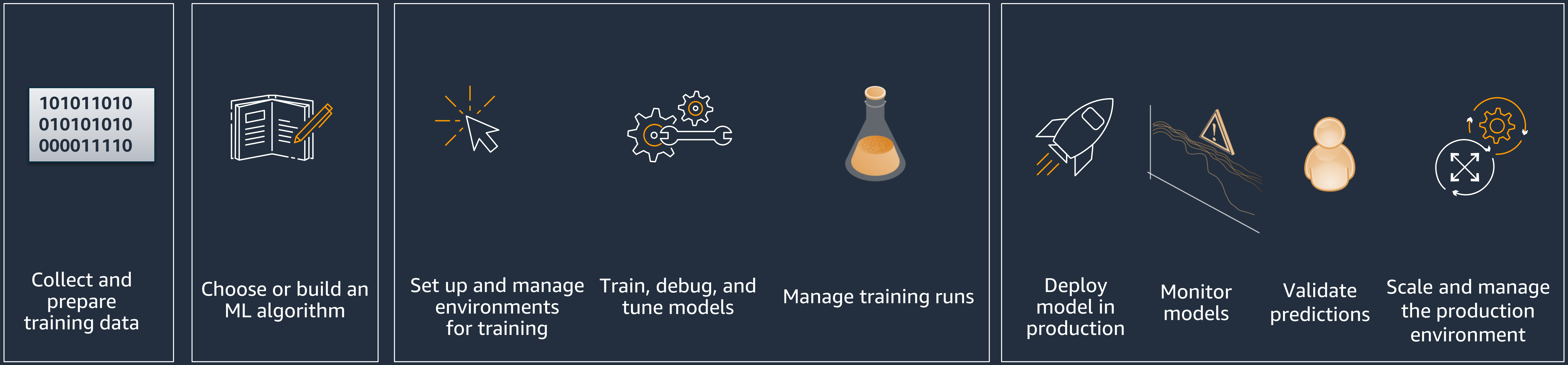
HOROVOD

DeepGraphLibrary

Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia | FPGA

aws

The machine learning workflow is iterative and complex



Prepare

101011010
010101010
000011110

Collect and prepare training data

Build

Choose or build an ML algorithm

Train & Tune

Set up and manage environments for training

Train, debug, and tune models

Manage training runs

Deploy & Manage

Deploy model in production

Monitor models

Validate predictions

Scale and manage the production environment

Amazon SageMaker helps you build, train, and deploy models

Web-based IDE for machine learning

Automatically build and train models

Prepare

Fully managed data processing jobs and data labeling workflows

Collect and prepare training data

Build

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

Train & Tune

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

Deploy & Manage

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

Amazon SageMaker helps you build, train, and deploy models

Prepare

Build

Train & Tune

Deploy & Manage

Web-based IDE for machine learning

Automatically build and train models

Prepare

Fully managed data processing jobs and data labeling workflows

101011010
010101010
000011110

Collect and prepare training data

Build

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

Train & Tune

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

Deploy & Manage

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

aws

Amazon SageMaker Studio

Fully integrated development environment (IDE) for machine learning

Collaboration at scale

Share notebooks without tracking code dependencies

Easy experiment management

Organize, track, and compare thousands of experiments

Automatic model generation

Get accurate models with full visibility & control without writing code

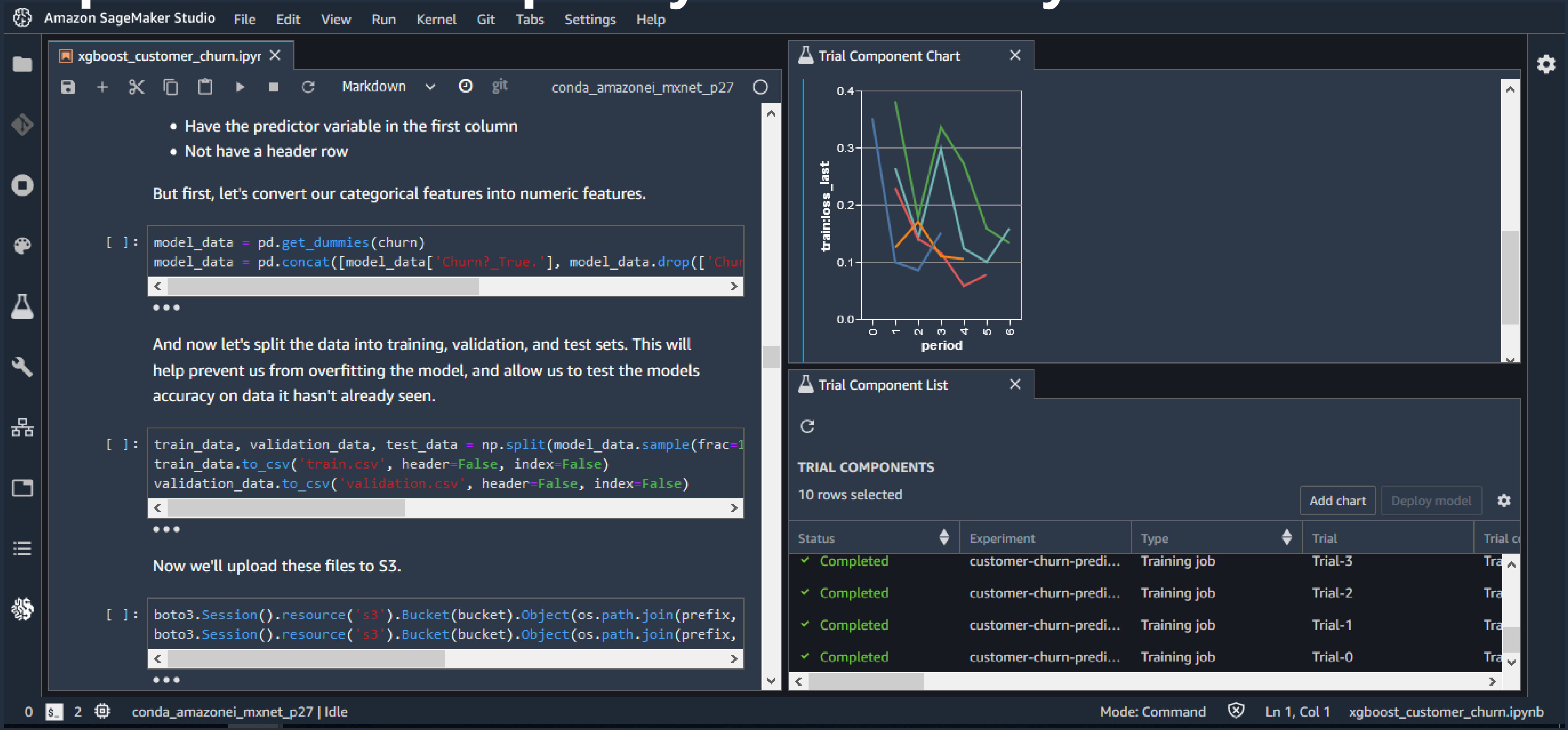
Higher quality ML models

Automatically debug errors, monitor models, & maintain high quality

Increased productivity

Code, build, train, deploy, & monitor in a unified visual interface

Use Amazon SageMaker Studio to update models and see impact on model quality immediately



- 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

Amazon SageMaker Autopilot

Automatic model creation with full visibility & control

Quick to start

Provide your data in a tabular form & specify target prediction

Automatic model creation

Get ML models with feature engineering & model tuning automatically done

Visibility & control

Get notebooks for your models with source code

Recommendations & Optimization

Get a leaderboard & continue to improve your model

Amazon SageMaker helps you build, train, and deploy models

Prepare | **Build** | **Train & Tune** | **Deploy & Manage**

Web-based IDE for machine learning

Automatically build and train models

Fully managed data processing jobs and data labeling workflows

101011010
010101010
000011110

Collect and prepare training data

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

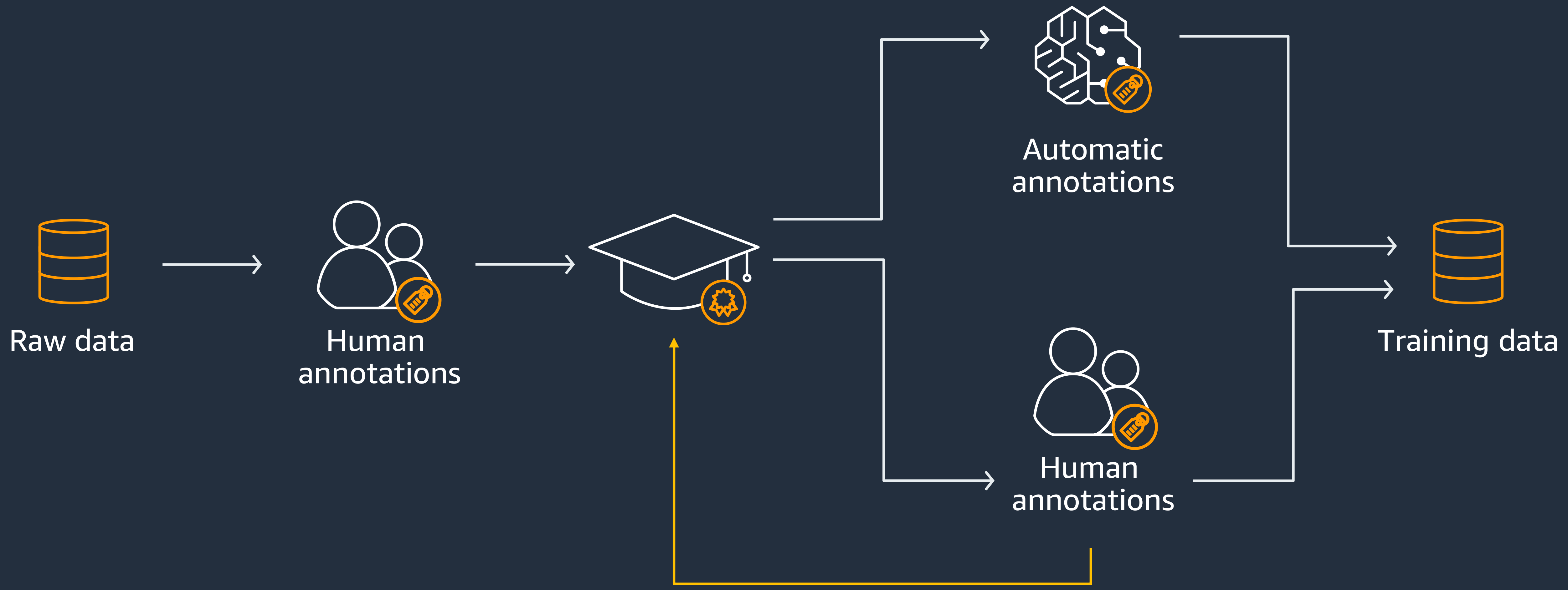
aws

Amazon SageMaker Ground Truth

Build highly accurate training datasets using machine learning

- Reduce data labeling costs by up to 70%
- Access labelers through Amazon Mechanical Turk, Amazon approved vendors, or use private human labelers
- Achieve accurate results quickly

How Amazon SageMaker Ground Truth Works



Raw data

Human annotations

Automatic annotations

Human annotations

Training data

Amazon SageMaker Processing

Analytics jobs for data processing and model evaluation

Fully managed

Achieve distributed processing for clusters

Custom processing

Bring your own script for feature engineering

Container support

Use SageMaker's built-in containers or bring your own

Security and compliance

Leverage SageMaker's security & compliance features

Automatic creation & termination

Your resources are created, configured, & terminated automatically

Amazon SageMaker helps you build, train, and deploy models

Prepare **Build** **Train & Tune** **Deploy & Manage**

Web-based IDE for machine learning

Automatically build and train models

Prepare

Fully managed data processing jobs and data labeling workflows

101011010
010101010
000011110

Collect and prepare training data

Build

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

Train & Tune

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

Deploy & Manage

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

aws

Amazon SageMaker Notebooks

Fast-start sharable notebooks (in preview)

Easy access with Single Sign-On (SSO)

Access your notebooks in seconds

Fully managed and secure

Administrators manage access and permissions

Fast setup

Start your notebooks without spinning up compute resources

Easy collaboration

Share notebooks with a single click

Flexible

Dial up or down compute resources (coming soon)

Amazon SageMaker has built-in algorithms or bring your own

Classification

- Linear Learner
- XGBoost
- K-nearest neighbors

Regression

- Linear Learner
- XGBoost
- K-nearest neighbors

Working with Text

- BlazingText
  - Word2Vec
  - Text Classification

Computer Vision

- Image Classification
- Object Detection
- Semantic Segmentation

Recommendation

- Factorization Machines

Anomaly Detection

- Random Cut Forests
- IP Insights

Sequence Translation

- Seq2Seq

Topic Modeling

- Latent Dirichlet Allocation
- Neural Topic Model

Forecasting

- DeepAR

Clustering

- KMeans

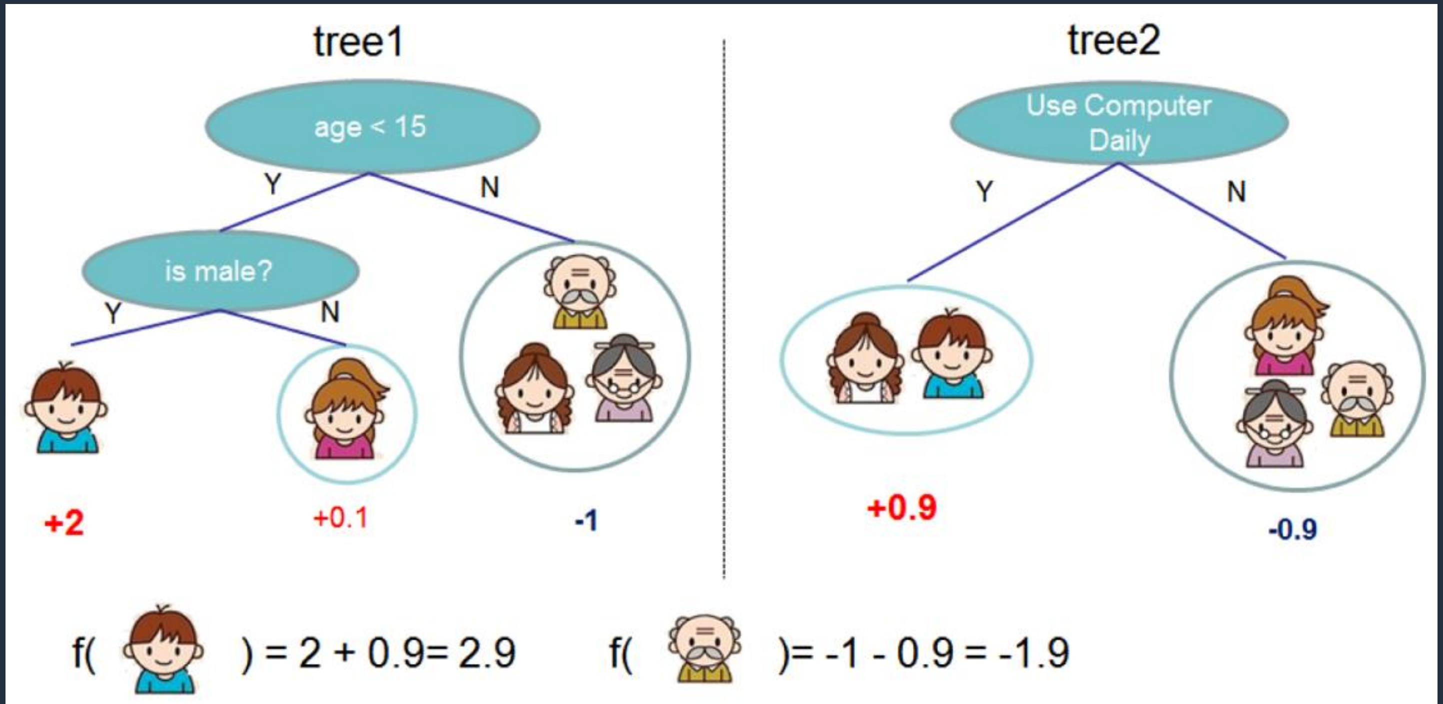
Feature Reduction

- Principal component analysis
- Object2Vec

pink: supervised, blue: unsupervised

XGBoost

- Open-source project
- Popular tree-based algorithm for regression, classification, and ranking
- Handles missing values and sparse data
- Supports distributed training
- Can work with datasets larger than RAM



https://github.com/dmlc/xgboost
https://xgboost.readthedocs.io/en/latest/
https://arxiv.org/abs/1603.02754

AWS Marketplace

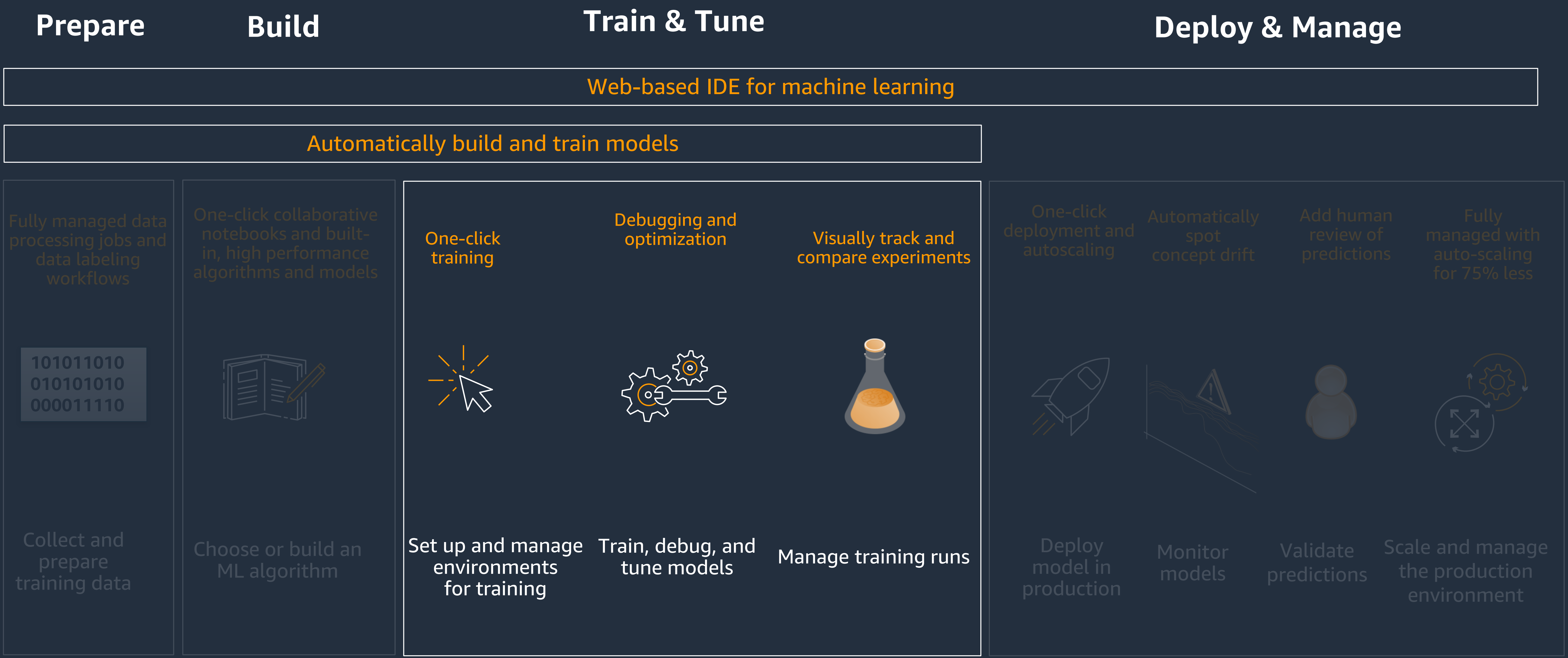
You can shop for algorithms, models, and data in AWS Marketplace

Browse or search AWS Marketplace

Subscribe in a single click

Available in Amazon SageMaker

Amazon SageMaker helps you build, train, and deploy models



Prepare

Build

Train & Tune

Deploy & Manage

Web-based IDE for machine learning

Automatically build and train models

Prepare

Fully managed data processing jobs and data labeling workflows

Collect and prepare training data

Build

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

Train & Tune

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

Deploy & Manage

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

Train your model with one click using Amazon SageMaker

Train with your own algorithms

Distributed by default

Train on a data stream

Single pass training

Not memory bound

Checkpoint for re-training

Amazon SageMaker Automatic Model Tuning

Automatically tune hyperparameters across algorithms

Tuning at scale

Adjust thousands of different combinations of algorithm parameters

Automated

Uses ML to find the best parameters

Faster

Eliminate days or weeks of tedious manual work

Examples

Decision Trees

Tree depth
Max leaf nodes
Gamma
Eta
Lambda
Alpha

Neural Networks

Number of layers
Hidden layer width
Learning rate
Embedding dimensions
Dropout

Amazon SageMaker Experiments

Organize, track, and compare training experiments

Tracking at scale

Track parameters & metrics across experiments & users

Custom organization

Organize experiments by teams, goals, & hypotheses

Visualization

Easily visualize experiments and compare

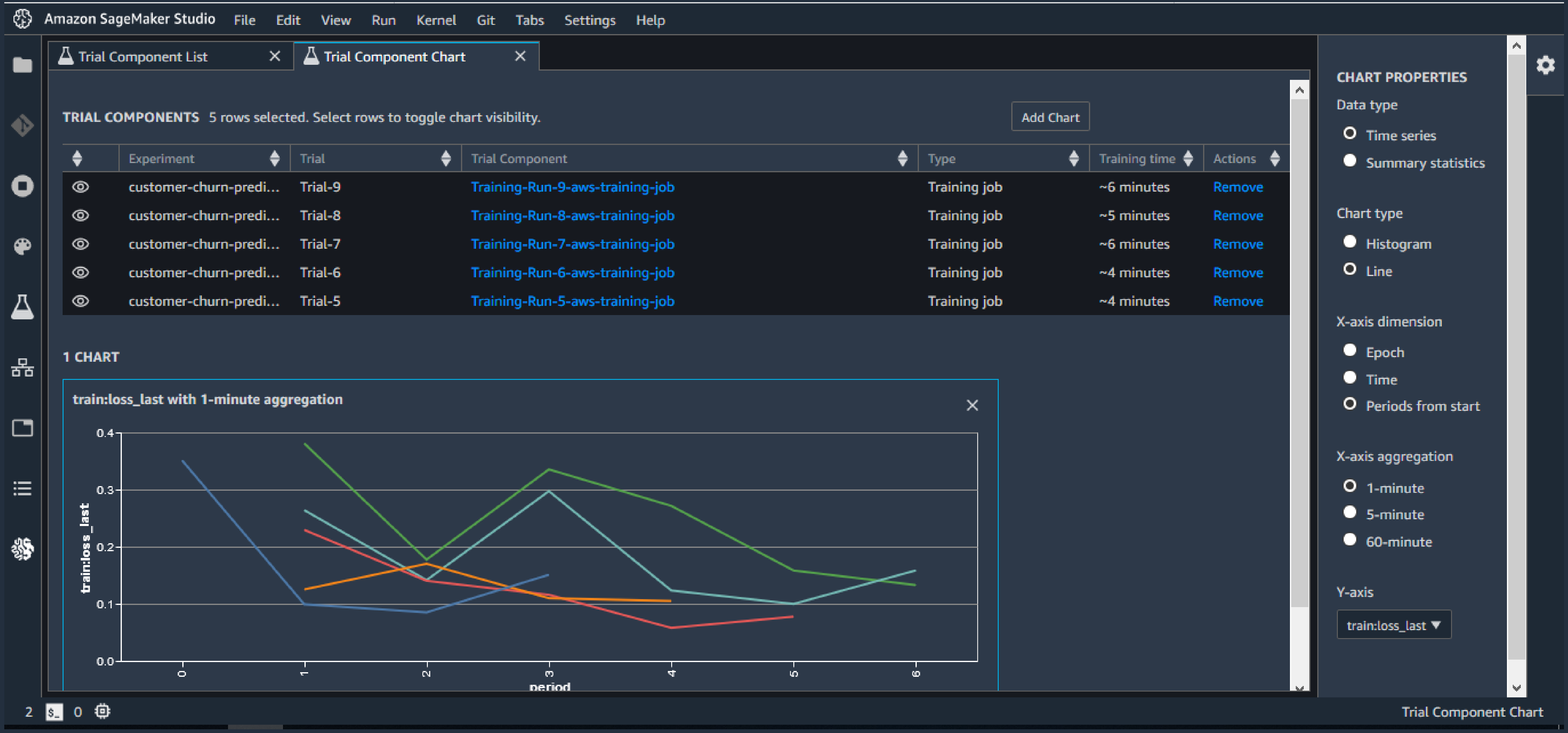
Metrics and logging

Log custom metrics using the Python SDK & APIs

Fast Iteration

Quickly go back & forth & maintain high-quality

Use Amazon SageMaker Experiments to track and manage thousands of experiments



TRIAL COMPONENTS 5 rows selected. Select rows to toggle chart visibility.

	Experiment	Trial	Trial Component	Type	Training time	Actions
👁	customer-churn-predi...	Trial-9	Training-Run-9-aws-training-job	Training job	~6 minutes	Remove
👁	customer-churn-predi...	Trial-8	Training-Run-8-aws-training-job	Training job	~5 minutes	Remove
👁	customer-churn-predi...	Trial-7	Training-Run-7-aws-training-job	Training job	~6 minutes	Remove
👁	customer-churn-predi...	Trial-6	Training-Run-6-aws-training-job	Training job	~4 minutes	Remove
👁	customer-churn-predi...	Trial-5	Training-Run-5-aws-training-job	Training job	~4 minutes	Remove

1 CHART

train:loss_last with 1-minute aggregation

CHART PROPERTIES

Data type

○ Time series

● Summary statistics

Chart type

● Histogram

○ Line

X-axis dimension

● Epoch

● Time

○ Periods from start

X-axis aggregation

○ 1-minute

● 5-minute

● 60-minute

Y-axis

train:loss_last ▼

Amazon SageMaker Debugger

Analysis and debugging, explainability, and alert generation

Relevant data capture

Data is automatically captured for analysis

Data analysis & debugging

Analyze & debug data with no code changes

Automatic error detection

Errors are automatically detected based on rules

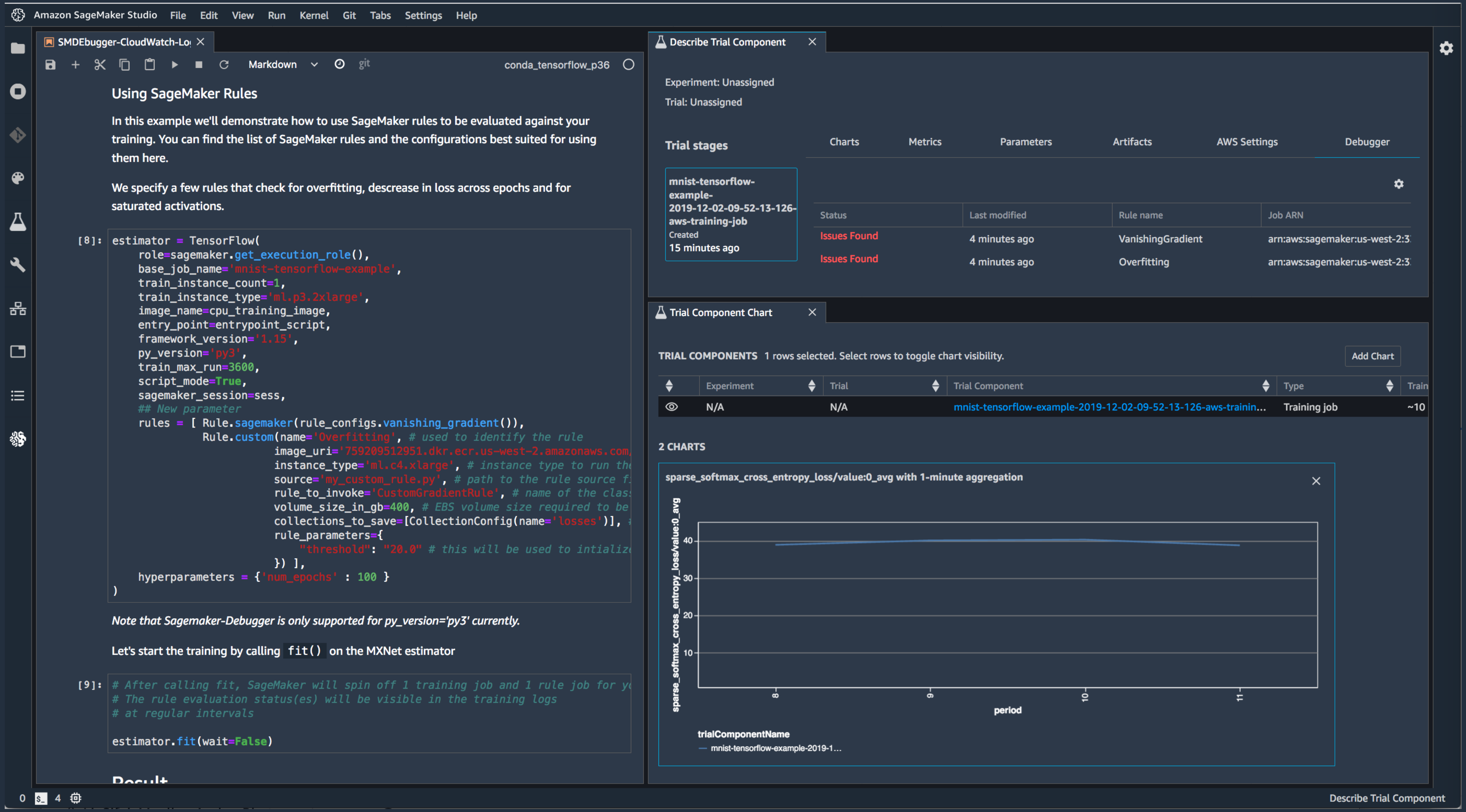
Improved productivity with alerts

Take corrective action based on alerts

Visual analysis and debugging

Visually analyze & debug from SageMaker Studio

Use Amazon SageMaker Debugger to identify issues such as vanishing gradients



Using SageMaker Rules

In this example we'll demonstrate how to use SageMaker rules to be evaluated against your training. You can find the list of SageMaker rules and the configurations best suited for using them here.

We specify a few rules that check for overfitting, descrease in loss across epochs and for saturated activations.

```
[8]: estimator = TensorFlow(
         role=sagemaker.get_execution_role(),
         base_job_name='mnist-tensorflow-example',
         train_instance_count=1,
         train_instance_type='ml.p3.2xlarge',
         image_name=cpu_training_image,
         entry_point=entrypoint_script,
         framework_version='1.15',
         py_version='py3',
         train_max_run=3600,
         script_mode=True,
         sagemaker_session=sess,
         ## New parameter
         rules = [ Rule.sagemaker(rule_configs.vanishing_gradient()),
                   Rule.custom(name='Overfitting', # used to identify the rule
                               image_uri='759209512951.dkr.ecr.us-west-2.amazonaws.com,
                               instance_type='ml.c4.xlarge', # instance type to run the
                               source='my_custom_rule.py', # path to the rule source f
                               rule_to_invoke='CustomGradientRule', # name of the clas
                               volume_size_in_gb=400, # EBS volume size required to be
                               collections_to_save=[CollectionConfig(name='losses')], #
                               rule_parameters={
                                   "threshold": "20.0" # this will be used to intialize
                               }) ],
         hyperparameters = {'num_epochs' : 100 }
     )
```

Note that Sagemaker-Debugger is only supported for py_version='py3' currently.

Let's start the training by calling `fit()` on the MXNet estimator

```
[9]: # After calling fit, SageMaker will spin off 1 training job and 1 rule job for yo
     # The rule evaluation status(es) will be visible in the training logs
     # at regular intervals

     estimator.fit(wait=False)
```

Result

Experiment: Unassigned

Trial: Unassigned

Trial stages

Charts | Metrics | Parameters | Artifacts | AWS Settings | Debugger

mnist-tensorflow-example-2019-12-02-09-52-13-126-aws-training-job
Created
15 minutes ago

Status	Last modified	Rule name	Job ARN
Issues Found	4 minutes ago	VanishingGradient	arn:aws:sagemaker:us-west-2:3
Issues Found	4 minutes ago	Overfitting	arn:aws:sagemaker:us-west-2:3

TRIAL COMPONENTS 1 rows selected. Select rows to toggle chart visibility.

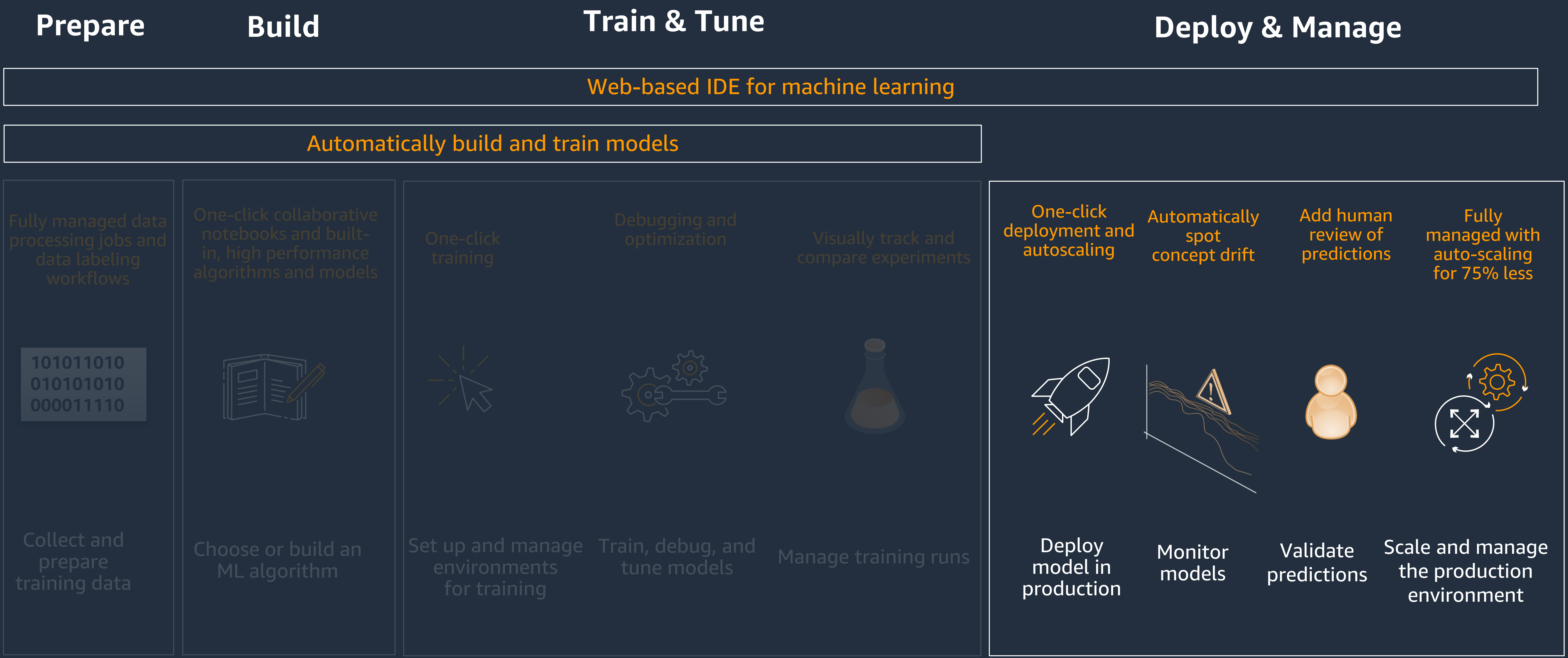
	Experiment	Trial	Trial Component	Type	Train
👁	N/A	N/A	mnist-tensorflow-example-2019-12-02-09-52-13-126-aws-trainin...	Training job	~10

2 CHARTS

sparse_softmax_cross_entropy_loss/value:0_avg with 1-minute aggregation

sparse_softmax_cross_entropy_loss/value:0_avg
40
30
20
10
8
9
10
11
period
trialComponentName
mnist-tensorflow-example-2019-1...

Amazon SageMaker helps you build, train, and deploy models



Prepare

Build

Train & Tune

Deploy & Manage

Web-based IDE for machine learning

Automatically build and train models

Fully managed data processing jobs and data labeling workflows

Collect and prepare training data

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

Amazon SageMaker is fully managed

One click model deployment

Auto-scaling

Low latency and high throughput

Bring your own model

Python SDK

Deploy multiple models on an endpoint

Amazon SageMaker Model Monitor

Continuous monitoring of models in production

Automatic data collection

Data is automatically collected from your endpoints

Continuous Monitoring

Define a monitoring schedule and detect changes in quality against a pre-defined baseline

Flexibility with rules

Use built-in rules to detect data drift or write your own rules for custom analysis

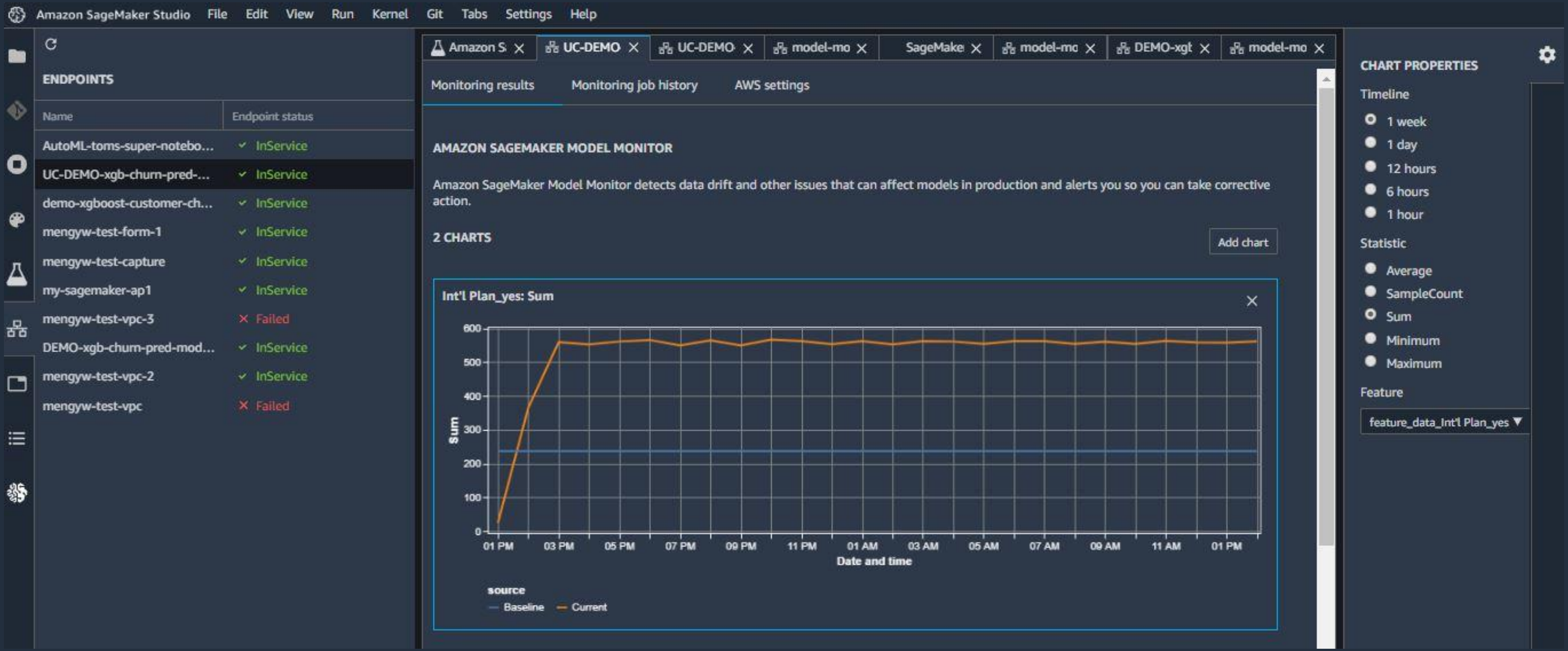
Visual data analysis

See monitoring results, data statistics, and violation reports in SageMaker Studio

CloudWatch Integration

Automate corrective actions based on Amazon CloudWatch alerts

Use Amazon SageMaker Model Monitor to identify model drift and take action



ENDPOINTS

Name	Endpoint status
AutoML-toms-super-notebo...	✓ InService
UC-DEMO-xgb-churn-pred-...	✓ InService
demo-xgboost-customer-ch...	✓ InService
mengyw-test-form-1	✓ InService
mengyw-test-capture	✓ InService
my-sagemaker-ap1	✓ InService
mengyw-test-vpc-3	✕ Failed
DEMO-xgb-churn-pred-mod...	✓ InService
mengyw-test-vpc-2	✓ InService
mengyw-test-vpc	✕ Failed

AMAZON SAGEMAKER MODEL MONITOR

Amazon SageMaker Model Monitor detects data drift and other issues that can affect models in production and alerts you so you can take corrective action.

2 CHARTS

CHART PROPERTIES

Timeline

- ○ 1 week
- ● 1 day
- ● 12 hours
- ● 6 hours
- ● 1 hour

Statistic

- ● Average
- ● SampleCount
- ○ Sum
- ● Minimum
- ● Maximum

Feature

feature_data_Int'l Plan_yes ▼

Amazon Augmented AI

Easily build workflows required for human review of predictions

Easily implement human review workflows

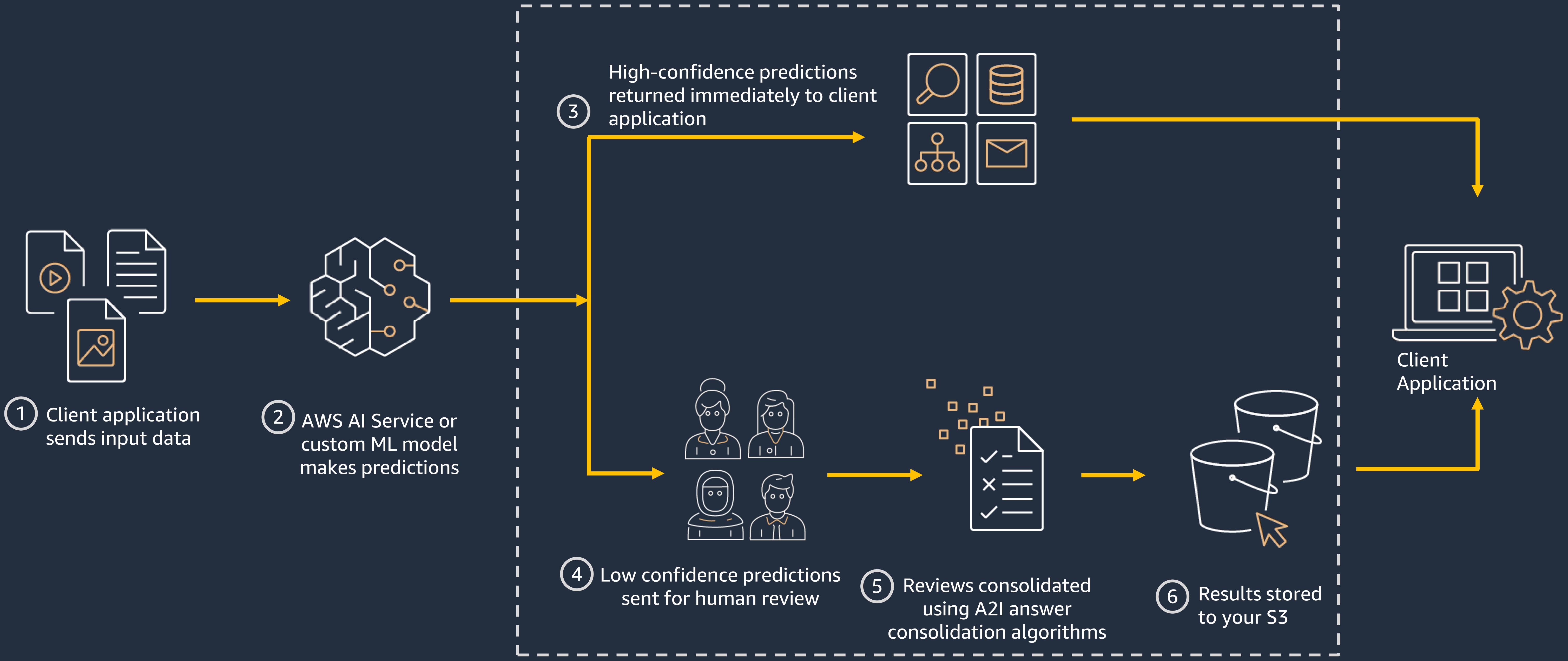
Reduce time to market with pre-built workflows and UIs

Multiple workforce options

Integrate with your custom ML models

Pre-built algorithms to increase accuracy

How Amazon Augmented AI works



1 Client application sends input data

2 AWS AI Service or custom ML model makes predictions

3 High-confidence predictions returned immediately to client application

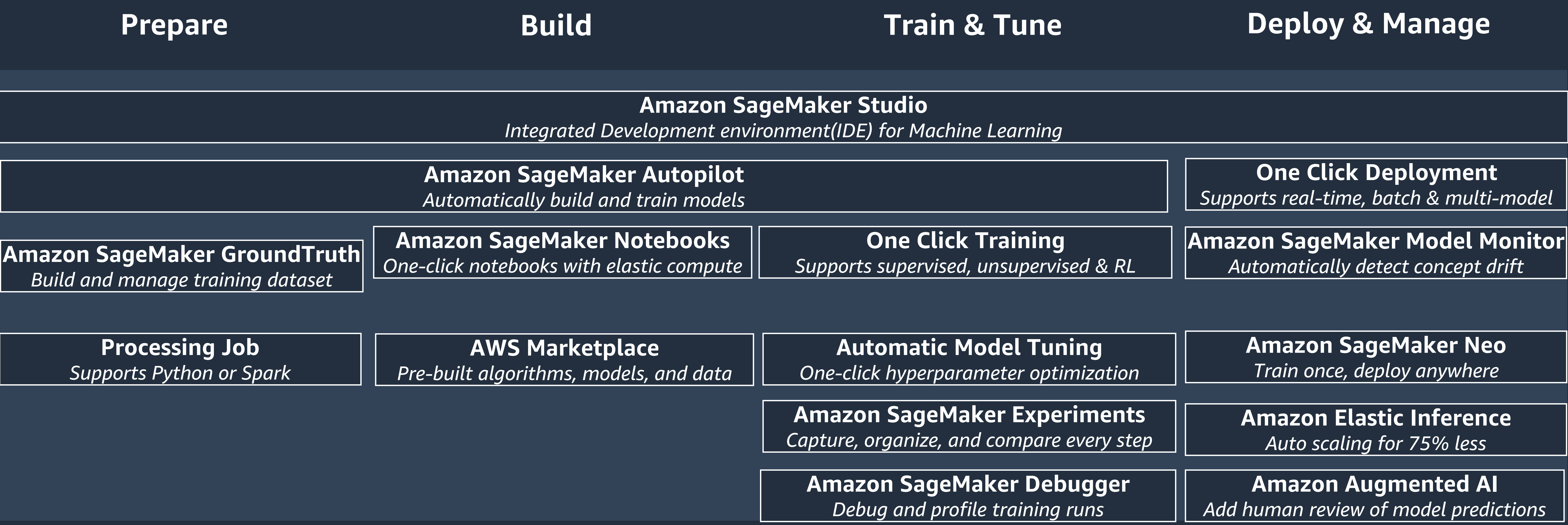
4 Low confidence predictions sent for human review

5 Reviews consolidated using A2I answer consolidation algorithms

6 Results stored to your S3

Client Application

Get started with Amazon SageMaker



Prepare

Build

Train & Tune

Deploy & Manage

Amazon SageMaker Studio
Integrated Development environment(IDE) for Machine Learning

Amazon SageMaker Autopilot
Automatically build and train models

One Click Deployment
Supports real-time, batch & multi-model

Amazon SageMaker GroundTruth
Build and manage training dataset

Amazon SageMaker Notebooks
One-click notebooks with elastic compute

One Click Training
Supports supervised, unsupervised & RL

Amazon SageMaker Model Monitor
Automatically detect concept drift

Processing Job
Supports Python or Spark

AWS Marketplace
Pre-built algorithms, models, and data

Automatic Model Tuning
One-click hyperparameter optimization

Amazon SageMaker Neo
Train once, deploy anywhere

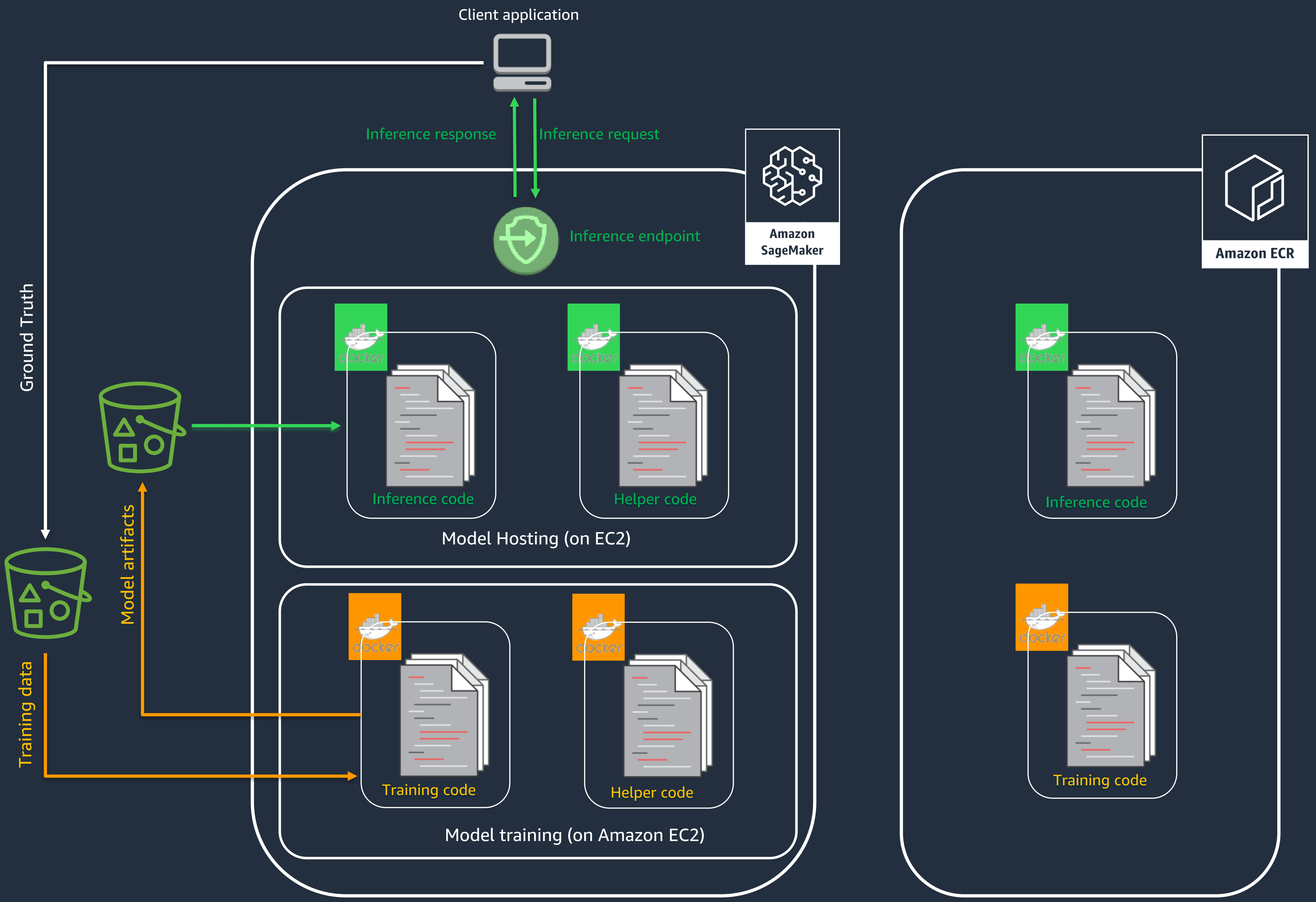
Amazon SageMaker Experiments
Capture, organize, and compare every step

Amazon Elastic Inference
Auto scaling for 75% less

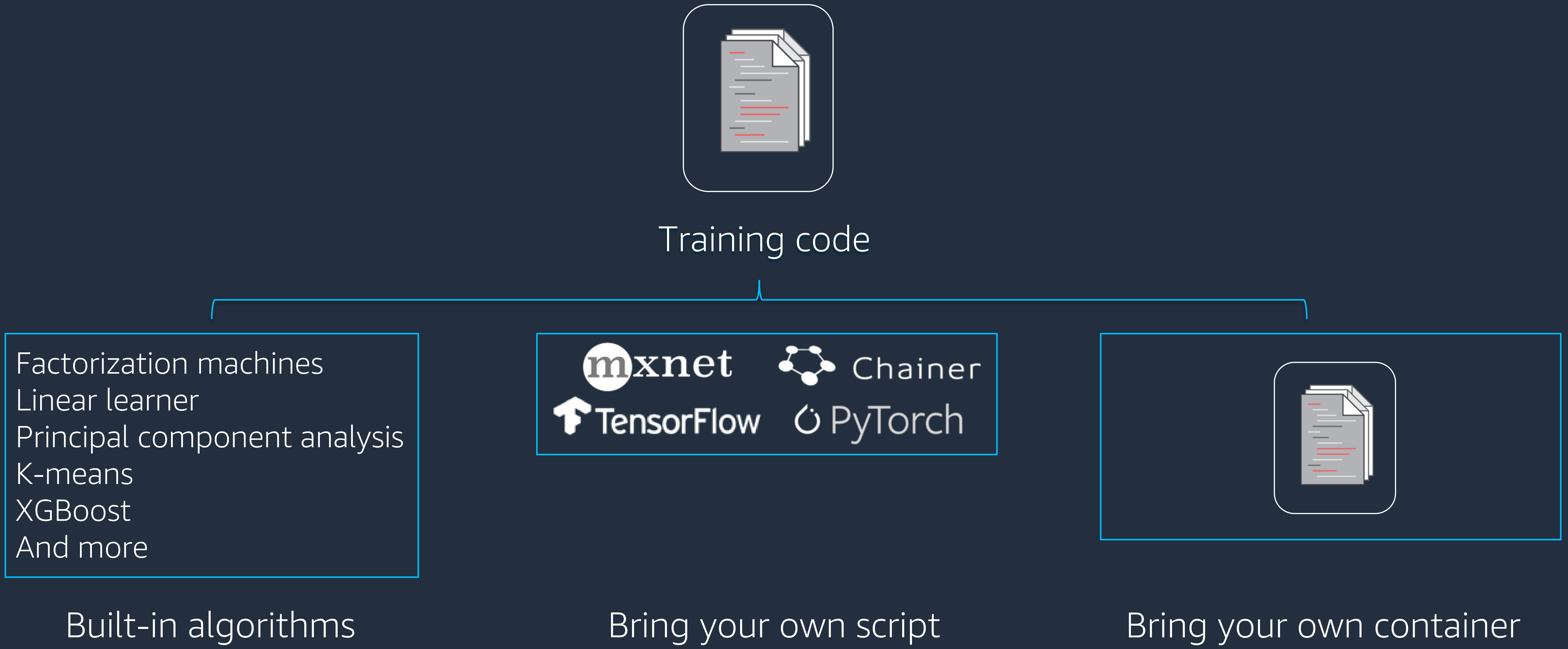
Amazon SageMaker Debugger
Debug and profile training runs

Amazon Augmented AI
Add human review of model predictions

Useful information for the labs


Client application
Inference response
Inference request
Amazon SageMaker
Amazon ECR
Inference endpoint
Ground Truth
Inference code
Helper code
Model Hosting (on EC2)
Inference code
Model artifacts
Training data
Training code
Helper code
Model training (on Amazon EC2)
Training code

Model options



Training code

Factorization machines
Linear learner
Principal component analysis
K-means
XGBoost
And more

mxnet Chainer
TensorFlow PyTorch

Built-in algorithms

Bring your own script

Bring your own container

Amazon SageMaker SDK

- AWS SDK for Python orchestrating all Amazon SageMaker activity
  - Algorithm selection, training, deploying, hyperparameter optimization, and so on
  - There’s also a Spark SDK (Python and Scala), which we won’t cover today

- High-level objects for:
  - Some built-in algos: K-means, PCA, and the like
  - Deep-learning libraries: TensorFlow, MXNet, PyTorch, Chainer
  - Sagemaker.estimator.estimator for everything else

https://github.com/aws/sagemaker-python-sdk

https://sagemaker.readthedocs.io/en/latest/

Confusion matrix

		Predict	
		0	1
Actual	0	True negative	False positive
	1	False negative	True positive

		Predict	
		0	1
Actual	0	3567	71
	1	355	126

Labs

Problem statement

Direct marketing is a common tactic to acquire customers. Because resources and a customer's attention are limited, the goal is to target only the subset of prospects who are likely to engage with a specific offer.

Predicting those potential customers based on readily available information like demographics, past interactions, and environmental factors is a common machine-learning problem.

We will train a model using XGBoost on a bank marketing dataset provided by UCI's ML Repository to predict if a customer will enroll for a term deposit at a bank after one or more phone calls.

Walkthrough: Notebook instance setup

Labs

1. Preparing the data
2. Training our first model with XGBoost
3. Deploying our model
4. Predicting with our model
5. Manually tuning our model
6. Finding optimal hyperparameters with automatic model tuning
7. Deploying our best 2 models
8. Predicting with our best 2 models

bit.ly/2OEdvFM

Resources

Resources

https://ml.aws

https://aws.amazon.com/sagemaker

https://github.com/awslabs/amazon-sagemaker-examples

https://github.com/aws/sagemaker-python-sdk

https://github.com/awslabs/amazon-sagemaker-workshop

Thank you!

Jonathan Dion

Senior Technical Evangelist

Amazon Web Services

@jotdion

linkedin.com/in/jotdion