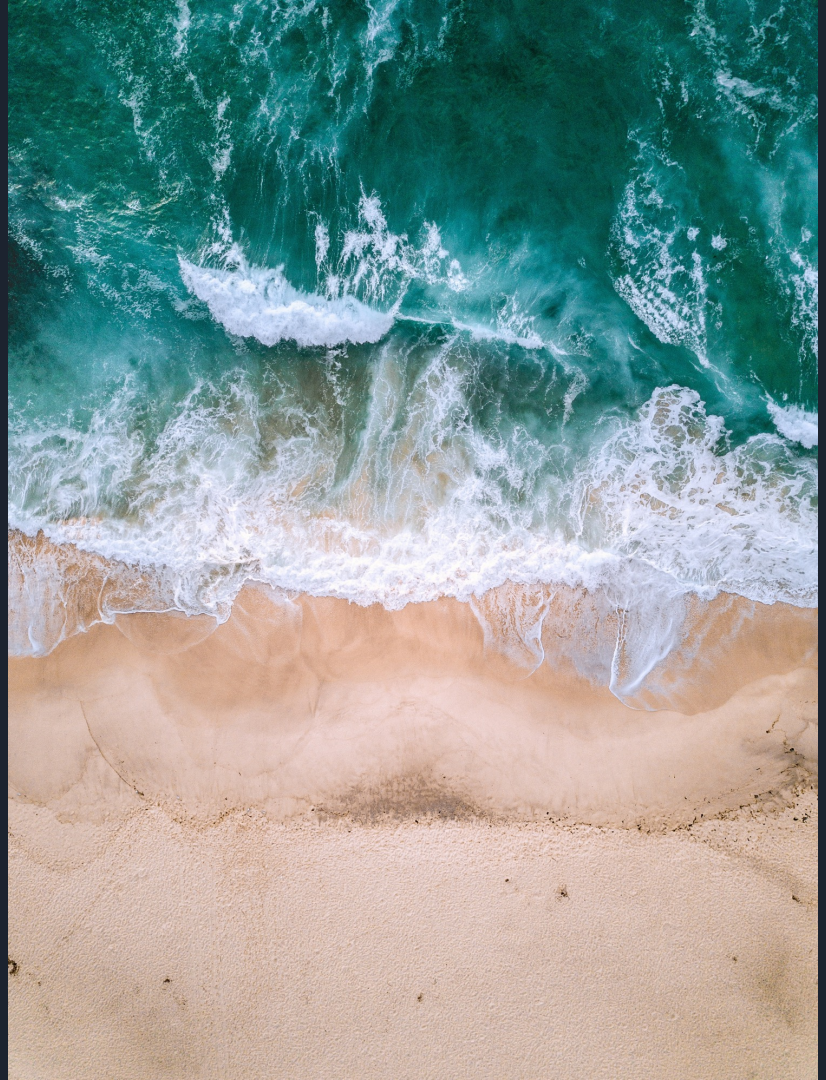
aws machine learning

Machine Learning on AWS with SageMaker

Joe Wilson
Solutions Architecture, AWS
Worldwide Public Sector

Agenda

- Introduction + Recap
- AI Services
- SageMaker Deep Dive
    - Build / train / tune
- SageMaker Deep Dive
    - Deploy
    - Prepare
- Q/A



aws machine learning

The AWS ML Stack

Broadest and most complete set of machine learning capabilities

AI SERVICES

HEALTH AI

- NEW Amazon HealthLake
- Amazon Transcribe for Medical
- Amazon Comprehend for Medical

INDUSTRIAL AI

- NEW AWS Panorama + Appliance
- NEW Amazon Monitron
- NEW Amazon Lookout for Equipment
- NEW Amazon Lookout for Vision

ANOMALY DETECTION

- NEW Amazon Lookout for Metrics

CODE AND DEVOPS

- NEW Amazon DevOps Guru
- Amazon CodeGuru

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

CONTACT CENTERS

- Contact Lens
- Voice ID *For Amazon Connect*

ML SERVICES

Amazon SageMaker

Label data

SAGEMAKER STUDIO IDE

- NEW Aggregate & prepare data
- NEW Store & share features
- Auto ML
- Spark/R
- NEW Detect bias
- Visualize in notebooks
- Pick algorithm
- Train models
- Tune parameters
- NEW Debug & profile
- Deploy in production
- Manage & monitor
- NEW CI/CD

Human review

NEW: SageMaker JumpStart

NEW: Model management for edge devices

FRAMEWORKS & INFRASTRUCTURE

TensorFlow, mxnet, PyTorch, GLUON, Keras, scikit learn, Horovod, DeepGraphLibrary

- Deep Learning AMIs & Containers
- GPUs & CPUs
- Elastic Inference
- Habana Gaudi
- Trainium
- Inferentia
- FPGA

aws

Amazon SageMaker overview

aws machine learning

Developing on Amazon SageMaker

The “build” part of build, train, deploy

aws machine learning

Amazon SageMaker Studio

Amazon SageMaker Studio | File | Edit | View | Run | Kernel | Git | Tabs | Settings | Help | Feedback

SageMakerAutopilotCandidat | autopilot_customer_churn.ipy

Markdown | git | 2 vCPU + 4 GiB | Python 3 (Data Sc

minutes.

[12]:
```
from time import gmtime, strftime, sleep
timestamp_suffix = strftime('%d-%H-%M-%S', gmtime())

auto_ml_job_name = 'automl-churn-' + timestamp_suffix
print('AutoMLJobName: ' + auto_ml_job_name)

sm.create_auto_ml_job(AutoMLJobName=auto_ml_job_name,
                      InputDataConfig=input_data_config,
                      OutputDataConfig=output_data_config,
                      AutoMLJobConfig={'CompletionCriteria':
                                       {'MaxCandidates': 20}
                                      },
                      RoleArn=role)
```

```
AutoMLJobName: automl-churn-16-18-04-54
```

[12]: {'AutoMLJobArn': 'arn:aws:sagemaker:us-east-1:865926247462:automl-

AbalonePipeline

less than 10 seconds ago

AbalonePipeline

Executions | Graph | Parameters | Settings

Search for step...

Start an execution

AbaloneMSECond

true | true

untitled.flow

Import | Prepare | Analyze | Export

Data flow

Choose the plus sign to add a step to the flow. Select a step to modify.

Source - sampled | Data types | Custom Pandas

S3: emissions_encoded.csv | Transform: emissions_encoded.csv | Transform: emissions_encoded.csv

Source - sampled | Data types | Custom Pandas

S3: train_with_header.csv | Transform: train_with_header.csv | Transform: train_with_header.csv

Describe Trial Component | automl-churn-16-18-04-54

less than 20 seconds ago

EXPERIMENT: AUTOML-CHURN-16-18-04-54

Pre-processing | Candidate Definitions Generated | Feature Engineering | Model Tuning

Amazon SageMaker Autopilot is analyzing the input data.

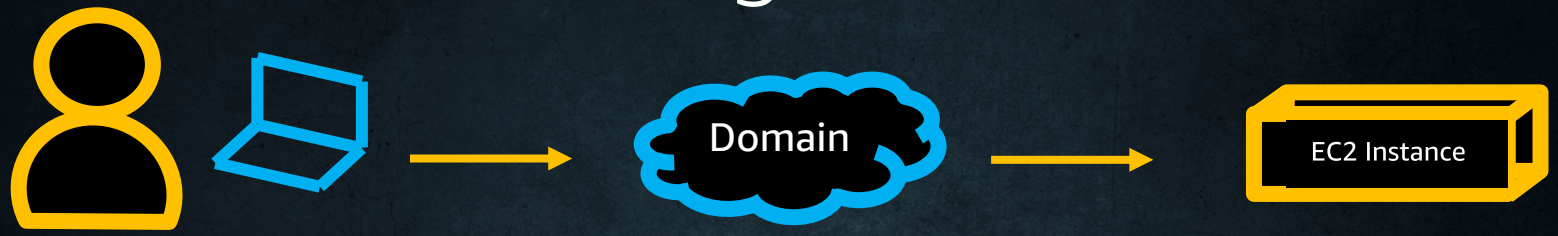
At the default setting of creating 250 trials, this process can take hours to complete.

You can always return to this page later by choosing this experiment on the Experiments tab in the navigation panel.

If experiment is taking too long to run, you can stop the experiment.

0 | 4 | Git: idle | automl-churn-16-18-04-54

Amazon SageMaker Studio



You!

- Can enable IAM or single sign on
- Fast notebook start-up
- Consolidated experience

Your IDE

Low-lift / automatic feature access

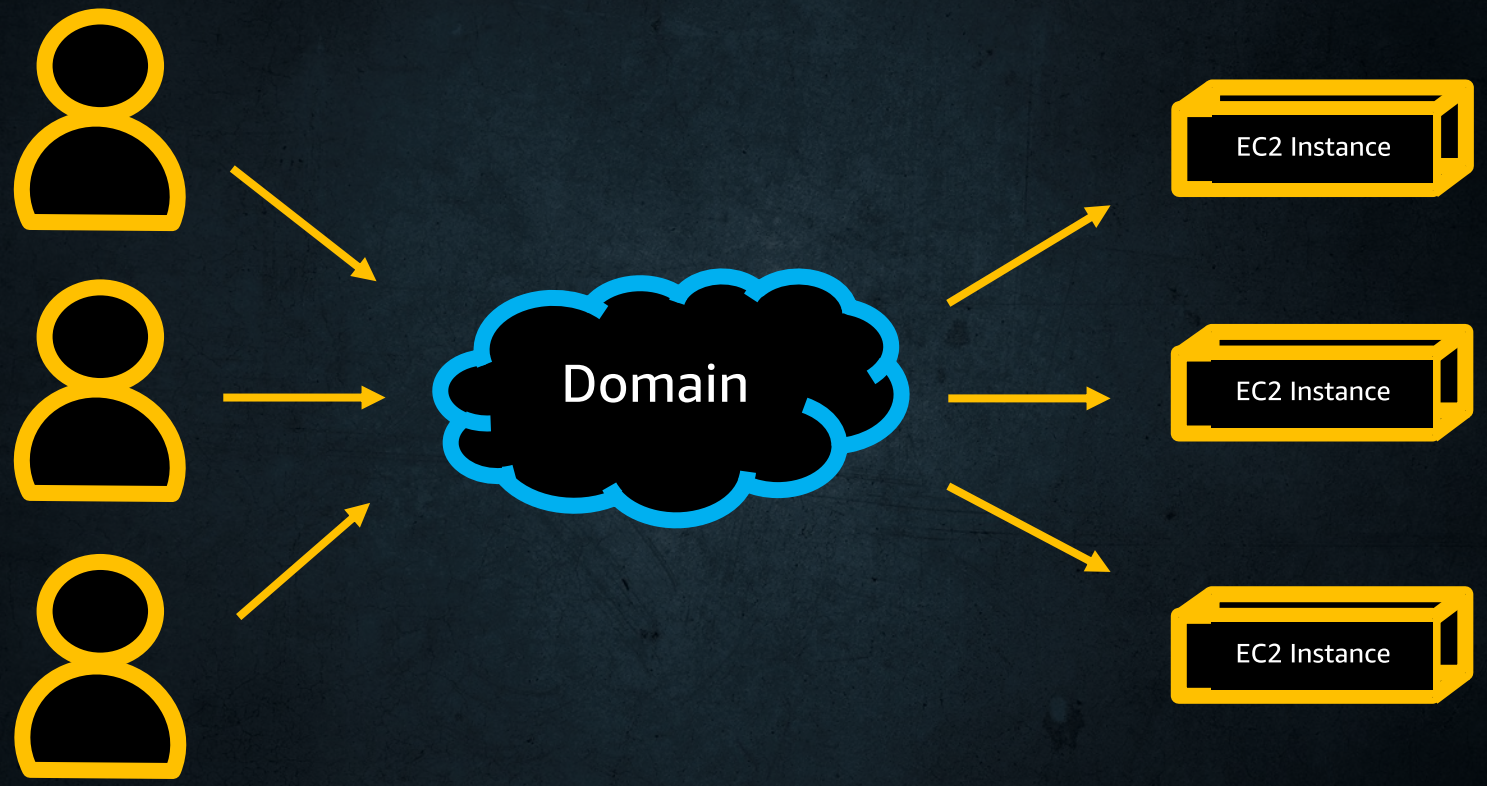
- Autopilot
- Experiments
- Debugger
- Model Monitoring
- Versioning
- Endpoint management

Your Compute

- Can create a shareable link with package dependences in single click
- Billing is per usage, not up time.
- Change environments per project

Remember! Features are backwards compatible, but differ in ease-of-use across instances vs Studio.

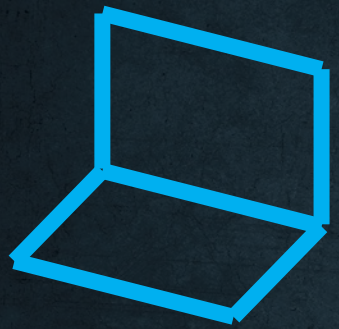
1-Click Notebook Sharing & Instance Upgrade



Studio defaults to 1:1 user to instance mapping, and you can share content on these via snapshots.

Amazon SageMaker brings elastic, dedicated compute

Person



Project

Dataset

Step in your
ML lifecycle

Allowing you to scale and manage each independently

Training on Amazon SageMaker

The “train” part of build, train, deploy

aws machine learning

Many Ways to Train Models on Amazon SageMaker

1. Built-in Algorithms
2. Script Mode
3. Docker container
4. Locally*
5. In a pipeline

aws

Train with your own XGBoost script

```python
# Open Source distributed script mode
from sagemaker.session import Session
from sagemaker.inputs import TrainingInput
from sagemaker.xgboost.estimator import XGBoost

est = XGBoost(entry_point='abalone.py',
            framework_version='1.2-1', # Note: framework_version is mandatory
            hyperparameters=hyperparams,
            role=role,
            instance_count=2,
            instance_type=instance_type,
            output_path=output_path)

train_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix, "train"), content_type=content_type)
validation_input = TrainingInput("s3://{}/{}/{}/".format(bucket, prefix, "validation"),content_type=content_type)

est.fit({'train': train_input, 'validation': validation_input})
```

```
2021-01-28 19:58:46 Starting - Starting the training job...
2021-01-28 19:59:10 Starting - Launching requested ML instancesProfilerReport-1611863926: InProgress
.........
```

dmlc XGBoost

Bring custom Python packages in requirements.txt

aws

Train with your own deep learning model

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(entry_point='mnist.py',
                    role=role,
                    framework_version='1.8.0',
                    instance_count=2,
                    instance_type='ml.c4.xlarge',
                    hyperparameters={
                        'epochs': 6,
                        'backend': 'gloo'
                    })

inputs = sagemaker_session.upload_data(path='data', bucket=bucket, key_prefix=prefix)

estimator.fit({'training': inputs})
```

mxnet PyTorch scikit learn APACHE Spark ML TensorFlow Chainer RAY

Train using a Docker File

```
FROM ubuntu:20.04

ARG DEBIAN_FRONTEND=noninteractive

# Don't prompt for tzdata on new versions of Ubuntu:
ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get -y update && apt-get install -y --no-install-recommends \
    wget \
    libcurl4-openssl-dev\
    libsodium-dev \
    r-base \
    r-base-dev \
    ca-certificates

RUN R -e "install.packages(c('mda', 'plumber'), repos='https://cloud.r-project.org')"

COPY mars.R /opt/ml/mars.R
COPY plumber.R /opt/ml/plumber.R

ENTRYPOINT ["/usr/bin/Rscript", "/opt/ml/mars.R", "--no-save"]

```

```
"AlgorithmSpecification": {
        "TrainingImage": '{}.dkr.ecr.{}.amazonaws.com/sagemaker-rmars:latest'.format(account,
region)
},
```

Store, Search, Version, Track Model Artifacts by Default

Amazon SageMaker > Search

Search

Property	Operator	Value	
InputDataConfig.DataSource.S3DataS	Contains	sagemaker	Remove

Add row

Search

Results: Training jobs

1 ...

Name	Algorithm	Data sources
tuning-job-1-086396f1dda841258d-019-6c6367d0	683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.0-1-cpu-py3	train : s3://sagemaker-us-east-1-865926247462/automl-churn-sdk-18-19-41-09/transformed-data/dpp0/rpb/train validation : s3://sagemaker-us-east-1-865926247462/automl-churn-sdk-18-19-41-09/transformed-data/dpp0/rpb/validation
tuning-job-1-086396f1dda841258d-018-fbb3e0c3	683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.0-1-cpu-py3	train : s3://sagemaker-us-east-1-865926247462/automl-churn-sdk-18-19-41-09/transformed-data/dpp8/rpb/train validation : s3://sagemaker-us-east-1-865926247462/automl-churn-sdk-18-19-41-09/transformed-data/dpp8/rpb/validation
tuning-job-1-086396f1dda841258d-017-5c14c90a	683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.0-1-cpu-py3	train : s3://sagemaker-us-east-1-865926247462/automl-churn-sdk-18-19-41-09/transformed-data/dpp6/rpb/train validation : s3://sagemaker-us-east-1-865926247462/automl-churn-sdk-18-19-41-09/transformed-data/dpp6/rpb/validation

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

View as text | Actions | Create Metric Filter

Filter events | Clear | 1m | 30m | 1h | 12h | Custom

Timestamp	Message
	No older events at this moment. *Retry*
2021-02-18T14:49:15.656-06:00	Docker entrypoint called with argument(s): train
2021-02-18T14:49:15.656-06:00	Running default environment configuration script
2021-02-18T14:49:16.656-06:00	[02/18/2021 20:49:16 INFO 139632547493632] Input channel configuration: {'validation'…
	[02/18/2021 20:49:16 INFO 139632547493632] Input channel configuration: {'validation': {'ContentType': 'text/csv', 'TrainingInputMode': 'Pipe', 'S3DistributionType': 'FullyReplicated', 'RecordWrapperType': 'None'}, 'train': {'ContentType': 'text/csv', 'TrainingInputMode': 'Pipe', 'S3DistributionType': 'FullyReplicated', 'RecordWrapperType': 'None'}}. Copy
2021-02-18T14:49:18.657-06:00	[02/18/2021 20:49:18 INFO 139632547493632] nvidia-smi took: 0.025197267532348633 secs…
2021-02-18T14:49:18.657-06:00	[02/18/2021 20:49:18 INFO 139632547493632] Number of GPUs being used: 0
2021-02-18T14:49:18.657-06:00	[02/18/2021 20:49:18 INFO 139632547493632] Creating network in context cpu(0).
2021-02-18T14:49:18.657-06:00	[02/18/2021 20:49:18 INFO 139632547493632] Starting model trainer.
2021-02-18T14:49:20.658-06:00	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=1, s…
	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=1, speed=224.68171381600214 samples/sec Copy
2021-02-18T14:49:20.658-06:00	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=2, s…
	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=2, speed=435.57591410342366 samples/sec Copy
2021-02-18T14:49:20.658-06:00	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=3, s…
	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=3, speed=629.347907622412 samples/sec Copy
2021-02-18T14:49:20.658-06:00	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=4, s…
	[02/18/2021 20:49:19 INFO 139632547493632] #progress_notice: epoch=0, iterations=4, speed=816.3276545798652 samples/sec Copy

aws machine learning

Reduce training costs with fully-managed spot instances

```
from sagemaker.estimator import Estimator

est = Estimator(container,
                role,
                hyperparameters=hyperparameters,
                instance_count=1,
                instance_type=instance_type,
                volume_size=5,  # 5 GB
                output_path=output_path,
                sagemaker_session=sagemaker.Session(),
                use_spot_instances=True,
                max_run=3600,
                max_wait=7200,
                checkpoint_s3_uri=checkpoint_s3_uri)

est.fit({'train': train_input}, job_name=job_name)
```

- Specify a max wait time
- SageMaker will default to giving you the lowest possible cost
- Store model checkpoints in S3 in case your job is interrupted for BYOM
- Many built-in algorithms automatically revert to a training job
- We have examples
- Save up to 90%!

Amazon SageMaker price reductions: Up to 18% lower prices on ml.p3 and ml.p2 instances

by Urvashi Chowdhary | on 07 OCT 2020 | in Amazon SageMaker, Artificial Intelligence | Permalink | Comments | Share

Deploying on Amazon SageMaker

The “deploy” part of build, train, deploy

 |

aws machine learning

Deploy any open source model with Amazon SageMaker

aws machine learning

Enabling autoscaling on your endpoint is easy

1

```
client = boto3.client('sagemaker')

response = client.describe_endpoint(
    EndpointName=endpoint_name)

variant_name = response['ProductionVariants'][0]['VariantName']
resource_id = 'endpoint/{}/variant/{}'.format(endpoint_name, variant_name)
```

Use boto3's SageMaker client to get the *resource ID*

2

```
scaling_client = boto3.client('application-autoscaling')

response = scaling_client.register_scalable_target(
    ServiceNamespace='sagemaker',
    ResourceId=resource_id,
    ScalableDimension='sagemaker:variant:DesiredInstanceCount',
    MinCapacity=min_capacity,
    MaxCapacity=max_capacity,
    RoleARN=role)
```

Use boto3's application autoscaling client to register a scalable target

Bring your own pretrained models to host on SageMaker using script mode

```
from sagemaker.tensorflow import TensorFlowModel

model = TensorFlowModel(model_data='s3://mybucket/model.tar.gz', role='MySageMakerRole')

predictor = model.deploy(initial_instance_count=1, instance_type='ml.c5.xlarge')
```

You can use the built-in inference script with the deep learning container, *or you can bring your own inference script.*

```
pytorch_model = PyTorchModel(model_data='s3://my-bucket/my-path/model.tar.gz', role=role,
                             entry_point='inference.py')

predictor = pytorch_model.deploy(instance_type='ml.c4.xlarge', initial_instance_count=1)
```


aws machine learning

Deploy *any open source model* on SageMaker with Docker

jupyter Dockerfile.inference 5 hours ago

File Edit View Language

```
ARG REGISTRY_URI
FROM ${REGISTRY_URI}/mxnet-inference:1.6.0-cpu-py3

RUN pip install autogluon
RUN pip install PrettyTable

# Defines inference.py as script entrypoint
ENV SAGEMAKER_PROGRAM inference.py
```

Remember, you need to build this image first, then push to ECR

aws machine learning

More features in SageMaker Studio

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

Use Amazon SageMaker Autopilot to create and review top performing regression and classification models

aws File Edit View Run Kernel Git Tabs Settings Help

/ automl-preview /

Name	Last Modified
bank-additional	12 hours ago
model	12 hours ago
sagemaker_auto...	3 hours ago

sagemaker_automl_direct_m: | my-sagemaker-autopilot | Deploy model

EXPERIMENT: MY-SAGEMAKER-AUTOPILOT

Open candidate generation notebook | Open data exploration notebook

Trials | Job profile

TRIALS

0 row selected

Deploy model

Trial name	Status	Start time	End time	Objective
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9206119775772095
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9202479720115662
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9200050234794617
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9195190072059631
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9191550016403198
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9190340042114258
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9189119935035706
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825

0 1 my-sagemaker-autopilot

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

aws

Use Amazon SageMaker Experiments to track and manage thousands of experiments

Amazon SageMaker Studio File Edit View Run Kernel Git Tabs Settings Help

Trial Component List | Trial Component Chart

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

Add Chart

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

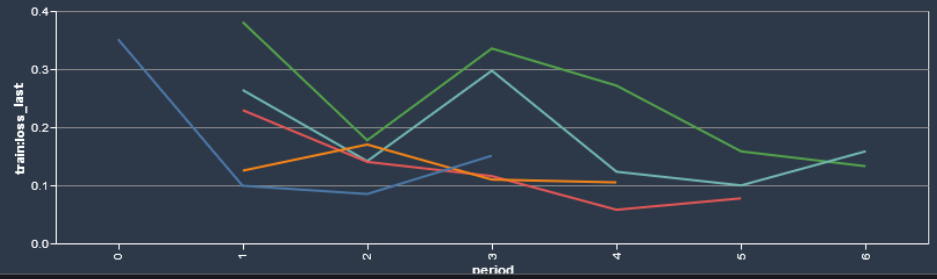


CHART PROPERTIES

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

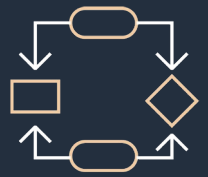
2 0 Trial Component Chart

Amazon SageMaker Debugger

Analysis and debugging, explainability, and alert generation

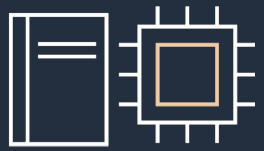
Relevant data capture

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

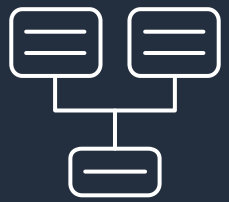
Visually analyze & debug from SageMaker Studio

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

Amazon SageMaker Studio File Edit View Run Kernel Git Tabs Settings Help

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 S. | UC-DEMO | UC-DEMO | model-mo | SageMake | model-mo | DEMO-xgb | model-mo

Monitoring results | Monitoring job history | AWS settings

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

Add chart

Int'l Plan_yes: Sum

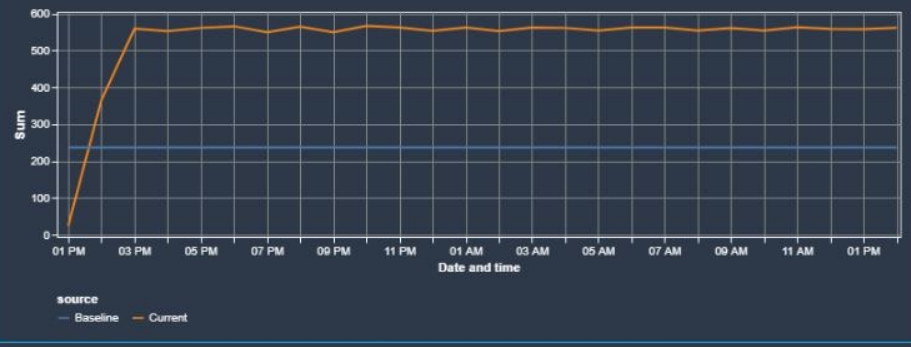


CHART PROPERTIES

Timeline
- 1 week
- 1 day
- 12 hours
- 6 hours
- 1 hour

Statistic
- Average
- SampleCount
- Sum
- Minimum
- Maximum

Feature

feature_data_Int'l Plan_yes

Q&A!

aws machine learning