

Amazon SageMaker Studio のご紹介

アマゾン ウェブ サービス ジャパン株式会社
機械学習ソリューションアーキテクト
上総 虎智



AWS の ML スタック

AI サービス: 機械学習の深い知識なしに利用可能

VISION Amazon Rekognitions <small>+Custom labels NEW</small>	SPEECH Amazon Polly Amazon Transcribe <small>+Medical NEW</small>	TEXT Amazon Comprehend <small>+Medical</small> Amazon Translate Amazon Textract	SEARCH <small>NEW</small> Amazon Kendra	CHATBOTS Amazon Lex	PERSONALIZATION Amazon Personalize	FORECASTING Amazon Forecast	FRAUD <small>NEW</small> Amazon Fraud Detector	DEVELOPMENT <small>NEW</small> Amazon CodeGuru	CONTACT CENTERS <small>NEW</small> Contact Lens <small>For Amazon Connect</small>
---	---	--	---	-----------------------------------	--	---	--	--	--

ML サービス: 機械学習のプロセス全体を効率化するマネージドサービス

Amazon SageMaker	Ground Truth	Augmented AI <small>NEW</small>	SageMaker Studio IDE <small>NEW</small>							SageMaker Neo
			Built-in algorithms	SageMaker Notebooks <small>NEW</small>	SageMaker Experiments <small>NEW</small>	Model tuning	SageMaker Debugger <small>NEW</small>	SageMaker Autopilot <small>NEW</small>	Model hosting	

ML フレームワークとインフラストラクチャ: 機械学習の環境を自在に構築して利用

TensorFlow Chainer	mxnet PYTORCH	GLUON Keras	Deep Learning AMLs & Containers	GPUs & CPUs	Elastic Inference	Inferentia	FPGA
-----------------------	------------------	----------------	---------------------------------	-------------	-------------------	------------	------

AWS の ML スタック

AI サービス: 機械学習の深い知識なしに利用可能

VISION Amazon Rekognitions <small>+Custom labels NEW</small>	SPEECH Amazon Polly Amazon Transcribe <small>+Medical NEW</small>	TEXT Amazon Comprehend <small>+Medical</small> Amazon Translate Amazon Textract	SEARCH <small>NEW</small> Amazon Kendra	CHATBOTS Amazon Lex	PERSONALIZATION Amazon Personalize	FORECASTING Amazon Forecast	FRAUD <small>NEW</small> Amazon Fraud Detector	DEVELOPMENT <small>NEW</small> Amazon CodeGuru	CONTACT CENTERS <small>NEW</small> Contact Lens <small>For Amazon Connect</small>
---	---	--	---	-----------------------------------	--	---	--	--	--

ML サービス: 機械学習のプロセス全体を効率化するマネージドサービス

Amazon SageMaker	Ground Truth	Augmented AI <small>NEW</small>	SageMaker Studio IDE <small>NEW</small>							SageMaker Neo
			Built-in algorithms	SageMaker Notebooks <small>NEW</small>	SageMaker Experiments <small>NEW</small>	Model tuning	SageMaker Debugger <small>NEW</small>	SageMaker Autopilot <small>NEW</small>	Model hosting	

ML フレームワークとインフラストラクチャ: 機械学習の環境を自在に構築して利用

TensorFlow	mxnet	GLUON	Keras	Deep Learning AMIs & Containers	GPUs & CPUs	Elastic Inference	Inferentia	FPGA
Chainer	PYTORCH	PySVM						

AWS の ML スタック

AI サービス: 機械学習の深い知識なしに利用可能

A grid of AWS AI services categorized by function: VISION (Amazon Rekognition, +Custom labels), SPEECH (Amazon Polly, Amazon Transcribe, +Medical), TEXT (Amazon Comprehend, Amazon Translate, Amazon Textract, +Medical), SEARCH (Amazon Kendra, NEW), CHATBOTS (Amazon Lex), PERSONALIZATION (Amazon Personalize), FORECASTING (Amazon Forecast), FRAUD (Amazon Fraud Detector, NEW), DEVELOPMENT (Amazon CodeGuru, NEW), CONTACT CENTERS (Contact Lens, For Amazon Connect, NEW). The 'Amazon SageMaker' text is overlaid in large yellow font in the center.

ML サービス: 機械学習のプロセス全体を効率化するマネージドサービス

A flow diagram of AWS ML services. It starts with 'Amazon SageMaker' on the left, followed by 'Ground Truth' and 'Augmented AI' (NEW). The main flow includes 'Built-in algorithms', 'SageMaker Notebooks' (NEW), 'SageMaker Experiments' (NEW), 'SageMaker Studio IDE' (NEW), 'Model tuning', 'SageMaker Debugger' (NEW), 'SageMaker Autopilot' (NEW), 'Model hosting', and 'SageMaker Model Monitor' (NEW). The flow ends with 'SageMaker Neo' on the right. Several components are highlighted with yellow boxes.

ML フレームワークとインフラストラクチャ: 機械学習の環境を自在に構築して利用



Deep Learning
AMIs & Containers

GPUs &
CPUs

Elastic
Inference

Inferentia

FPGA

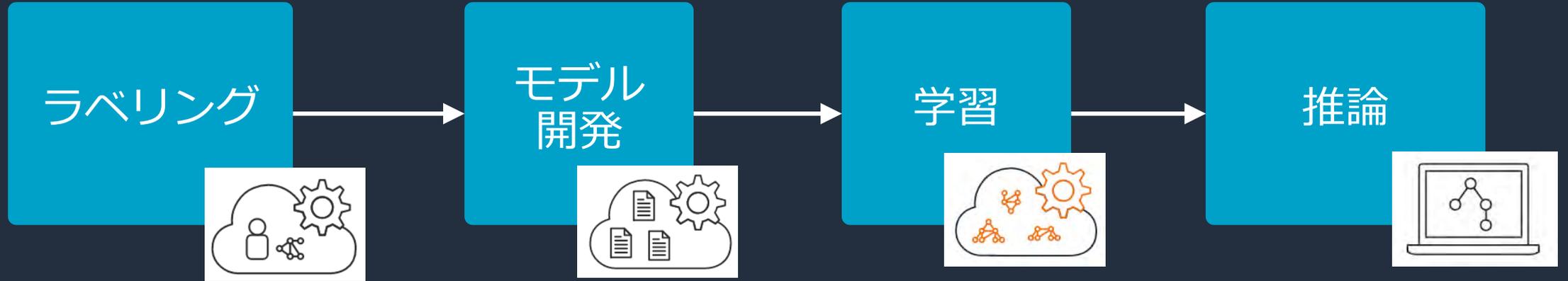
Amazon SageMaker における 基本的な ML 開発運用の流れ





Amazon SageMaker

機械学習フルマネージドサービス



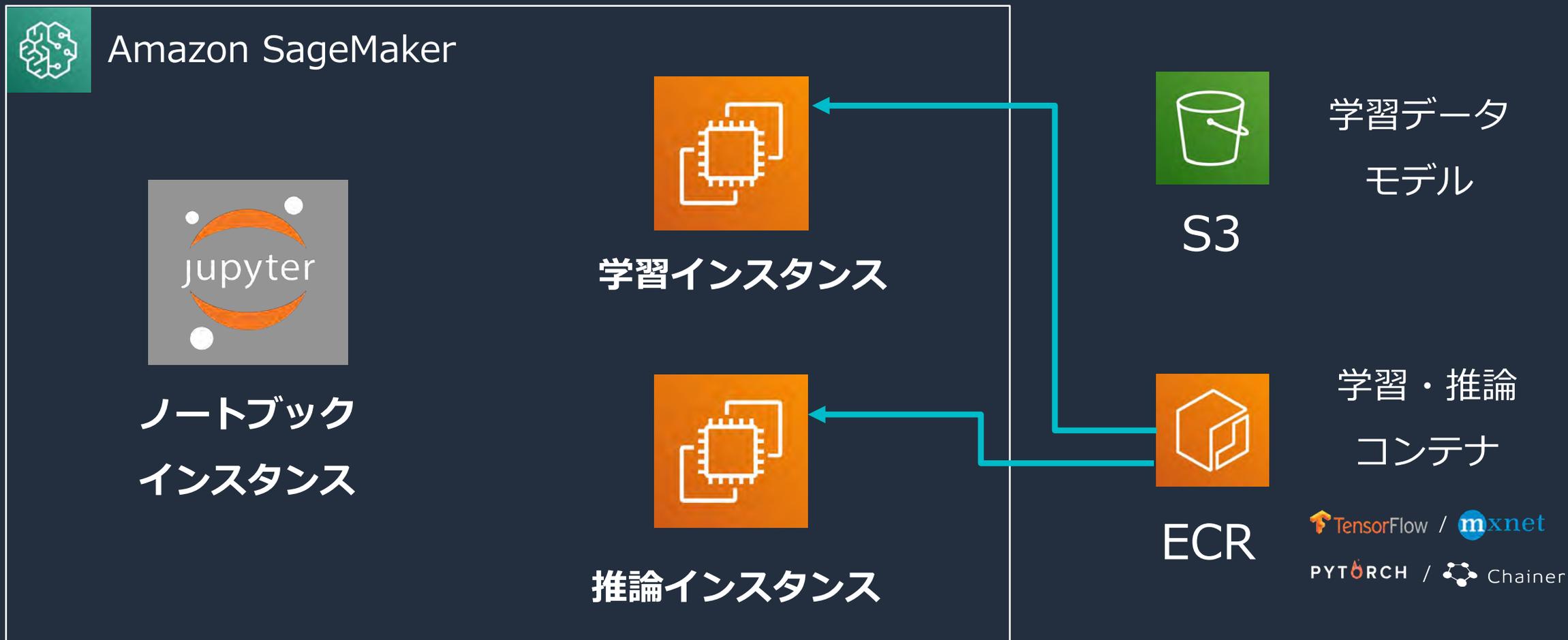
【ラベリング】 Mechanical Turk, クラウドソーシング, プライベートワーカー等へジョブ依頼

【開発】 Jupyter 環境を簡単に起動. 前処理・モデル開発・データ入出力から学習～推論と一連の SageMaker 上の ML ワークフローを API で記述

【学習】 SageMaker コンテナで学習環境を素早く展開, 学習ジョブを実行. 複数インスタンスでの分散学習, **Spot学習**, **大量学習データの高速読み込み**, **ハイパーパラメーターチューニング**

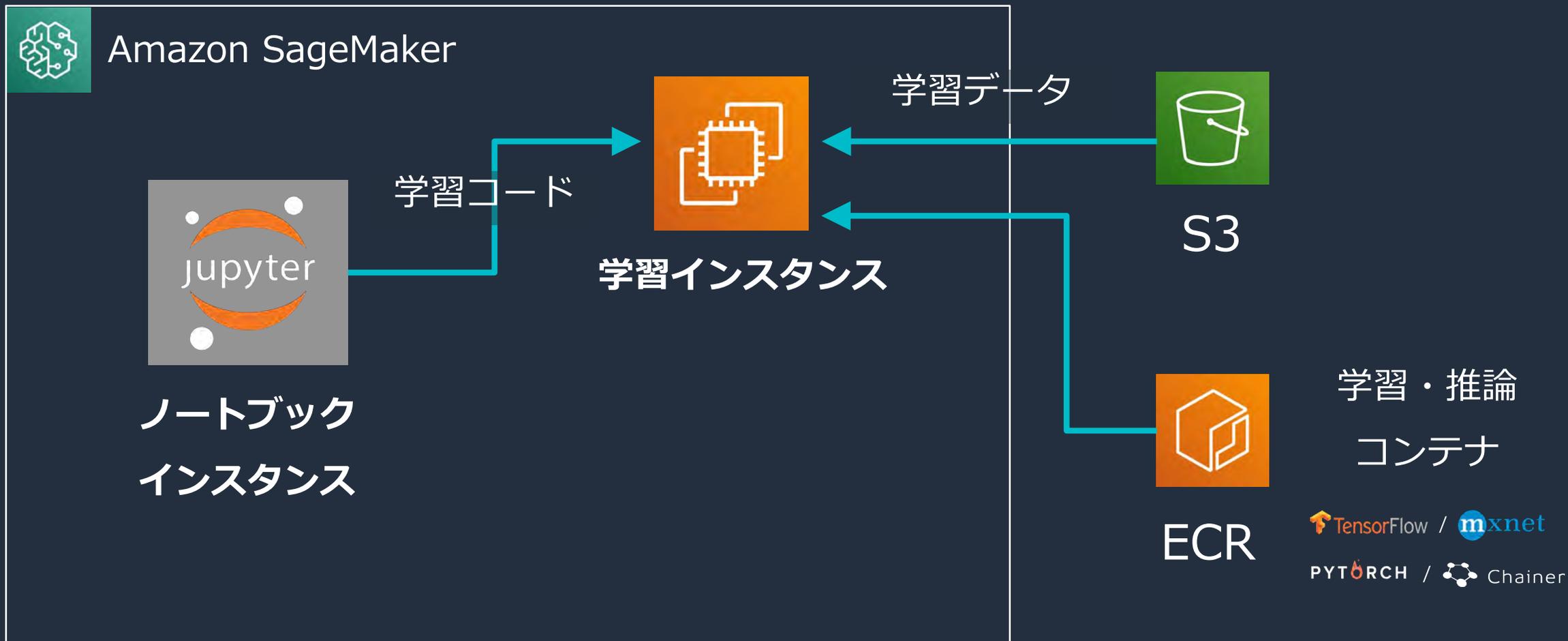
【推論】 推論用エンドポイントを簡単にデプロイ. **リアルタイム・バッチ推論**. エッジ環境への**モデル変換**, GPUアタッチなど**コスト効率よく推論**するための機能が充実.

学習・推論環境をコンテナで素早く構築



The Jupyter Trademark is registered with the U.S. Patent & Trademark Office.

学習環境をコンテナで素早く構築



The Jupyter Trademark is registered with the U.S. Patent & Trademark Office.

学習環境をコンテナで素早く構築



Amazon SageMaker



ノートブック
インスタンス

学習が終わると削除
学習時間のみ秒単位で課金

学習コード・
学習済モデル



S3



ECR

学習・推論
コンテナ

TensorFlow / mxnet

PYTORCH / Chainer

推論環境をコンテナで素早く構築



The Jupyter Trademark is registered with the U.S. Patent & Trademark Office.

Amazon SageMaker 各機能の詳細



AWS の ML スタック

AI サービス: 機械学習の深い知識なしに利用可能

Amazon SageMaker

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

ML サービス: 機械学習のプロセス全体を効率化するマネージドサービス

Amazon SageMaker

Ground Truth	Augmented AI NEW	Built-in algorithms	SageMaker Notebooks NEW	SageMaker Experiments NEW	Model tuning	SageMaker Debugger NEW	SageMaker Autopilot NEW	Model hosting	SageMaker Model Monitor NEW	SageMaker Neo
--------------	-------------------------	---------------------	--------------------------------	----------------------------------	--------------	-------------------------------	--------------------------------	---------------	------------------------------------	---------------

SageMaker Studio IDE NEW

ML フレームワークとインフラストラクチャ: 機械学習の環境を自在に構築して利用



Deep Learning
AMIs & Containers

GPUs &
CPUs

Elastic
Inference

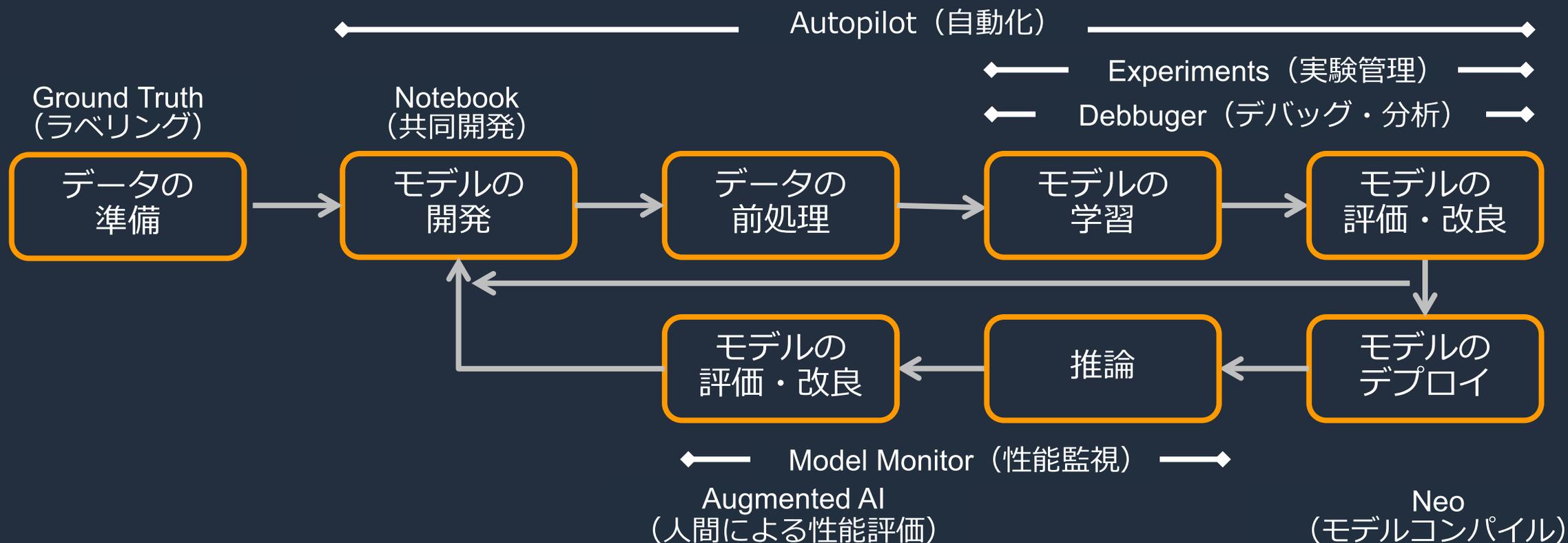
Inferentia

FPGA

機械学習のワークフローと各機能の対応

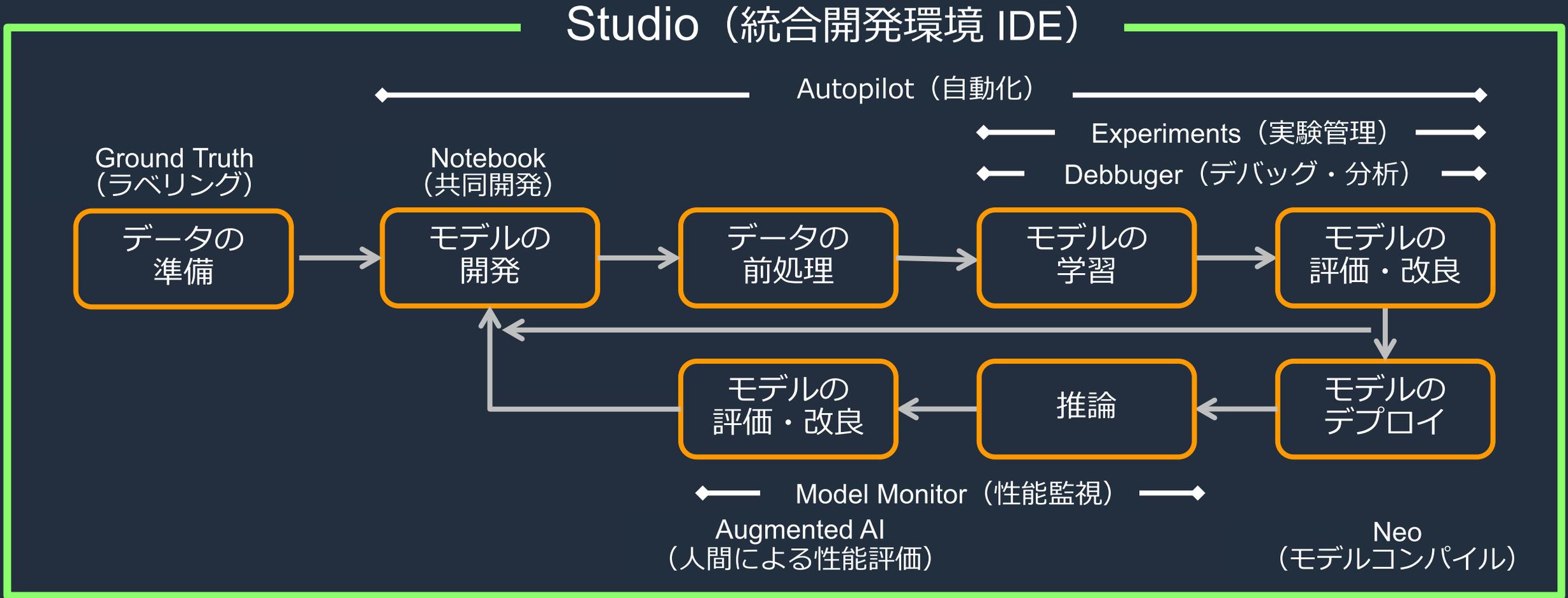


機械学習のワークフローと各機能の対応



コンソールや Python SDK を通して、
ノートブックインスタンスやローカルの開発環境から各機能を活用可能

機械学習のワークフローと各機能の対応



コンソールや Python SDK を通して、
ノートブックインスタンスやローカルの開発環境から各機能を活用可能

Amazon SageMaker Studio

- SageMaker の各機能呼び出すことができるweb上の統合開発環境
- AWS Single Sign-On (AWS SSO) , AWS IAM でアクセス可能
- 対応リージョン: オハイオ



Amazon SageMaker Studio

Available in preview

Loading the JupyterServer application default...

Amazon SageMaker



Amazon SageMaker Studio

Dashboard

Search

Ground Truth

Labeling jobs

Labeling datasets

Labeling workforces

Notebook

Notebook instances

Lifecycle configurations

Git repositories

Training

Algorithms

Training jobs

Hyperparameter tuning jobs

Inference

Compilation jobs

Model packages

Models

MACHINE LEARNING

Amazon SageMaker

Build, train, and deploy machine learning models at scale

The quickest and easiest way to get ML models from idea to production.

Get started

Explore Amazon SageMaker Studio, a machine learning Integrated Development Environment (IDE) to build, train and debug models, track experiments, deploy models in production, and monitor drifts.

[Launch Amazon SageMaker Studio](#)

Pricing (US)

With Amazon SageMaker, you pay only for what you use. Authoring, training and hosting is billed by the second, with no minimum fees and no upfront commitments.

[Learn more](#)

How it works



Label

Set up and manage labeling jobs for highly accurate



Build

Connect to other AWS services and transform data in



Train

Use Amazon SageMaker's algorithms and

Related services

[AWS Glue](#)



Amazon SageMaker Studio

Dashboard

Search

Ground Truth

Labeling jobs

Labeling datasets

Labeling workforces

Notebook

Notebook instances

Lifecycle configurations

Git repositories

Training

Algorithms

Training jobs

Hyperparameter tuning jobs

Amazon SageMaker Studio (Available in Preview)

Summary

Amazon SageMaker Studio address

[https://\[Redacted\].studio.us-east-2.sagemaker.aws](https://[Redacted].studio.us-east-2.sagemaker.aws)

Status

✔ InService

Users

Remove users

Assign users

Search users

< 1 > ⚙

<input type="checkbox"/>	Username	Email
<input type="checkbox"/>	[Redacted]-sagemaker	[Redacted]@amazon.co.jp

Amazon SageMaker Studio

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

xgboost_customer_churn.ipynr

- 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(['Churn?_True:'], axis=1)])
```

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), [int(model_data.shape[0]*0.7), int(model_data.shape[0]*0.85)])
      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, 'train.csv')).upload_file(train_data.to_csv(index=False).read().decode('utf8'), overwrite=True)
      boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'validation.csv')).upload_file(validation_data.to_csv(index=False).read().decode('utf8'), overwrite=True)
```

Trial Component Chart

period	Trial-0	Trial-1	Trial-2	Trial-3
0	0.35	0.30	0.38	0.35
1	0.10	0.15	0.12	0.18
2	0.08	0.10	0.15	0.15
3	0.12	0.10	0.10	0.32
4	0.12	0.05	0.10	0.28
5	0.10	0.08	0.15	0.15
6	0.15	0.10	0.15	0.15

Trial Component List

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

Mode: Command Ln 1, Col 1 xgboost_customer_churn.ipynb

機械学習ワークフローの中での課題感



データ準備時の課題

- 作業を割り当てるワーカーの募集
- 進捗管理・作業割り振り
- 効率の良いラベリングツールの作成
- これらを用意した上で数万個のデータへのラベル付け…

Amazon SageMaker Ground Truth

データにラベル (Ground Truth) を付与するアノテーション作業の支援サービス

- アノテーションの一般的なワークフローをサポート
- 5種類の組み込みラベリングツールを提供
- アノテーション作業を行うワーカーとの連携・管理機能を提供
- 大規模データセットに対しては自動ラベリング機能で最大70%のコスト削減



迅速なラベル付け



ワーカーとの連携が容易



高精度

SageMaker Ground Truth 利用のワークフロー

1. アノテーション対象のデータをアップロード



Amazon S3

3. タスクはワーカーに自動で割り振られる

4. ラベリングツールでワーカーがアノテーションを行う



5. アノテーション結果が S3に格納される

6. 学習・推論に利用



複数人の結果をマージ

S3

ビルトインアルゴリズムをそのまま適応可能

2. ラベリングジョブの作成

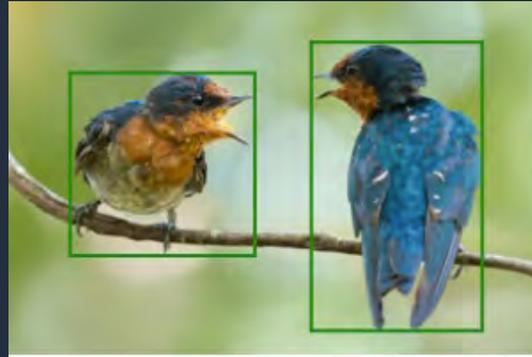
Amazon SageMaker Ground Truth

Amazon SageMaker

組み込みラベリングツールと独自実装が利用可能



画像分類



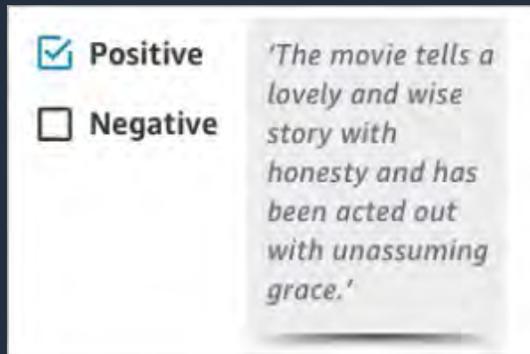
物体検出



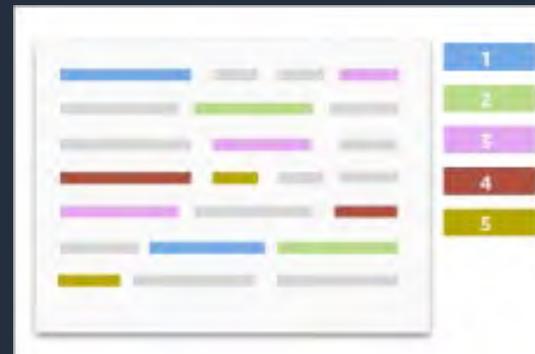
セマンティック
セグメンテーション



カスタム



文章分類



固有表現抽出

カスタムテンプレートの仕組み



前処理用 AWS Lambda

入力データの記述されたマニフェストファイルの各項目を読み込み、それをテンプレートエンジンに返す処理を記述



HTMLテンプレート

テンプレートエンジンのLiquidを採用
簡単なサンプルも多数用意されている

後処理用 AWS Lambda

ワーカーがタスクを終了した際の注釈統合を処理



※ AWS Lambdaはドキュメントのサンプルコードをベースに作成する必要あり
HTMLテンプレートは多数のサンプルの中から選択できる

ワーカーは以下の3種類から選択可能



パブリック

- クラウドソーシングサービスの Amazon Mechanical Turk を利用
- 非言語依存で機密性の低いタスク向き

プライベート



- 友人や社員をワーカーとして登録出来る
- 機密性の高いタスク向き
- ワーカーの管理にCognitoを利用（SAMLでの連携も可）

ベンダー



- SageMaker Ground Truthに[登録済みのアノテーション専門ベンダー](#)に依頼
- 現時点では日本のベンダーは登録されていない

機械学習ワークフローの中での課題感

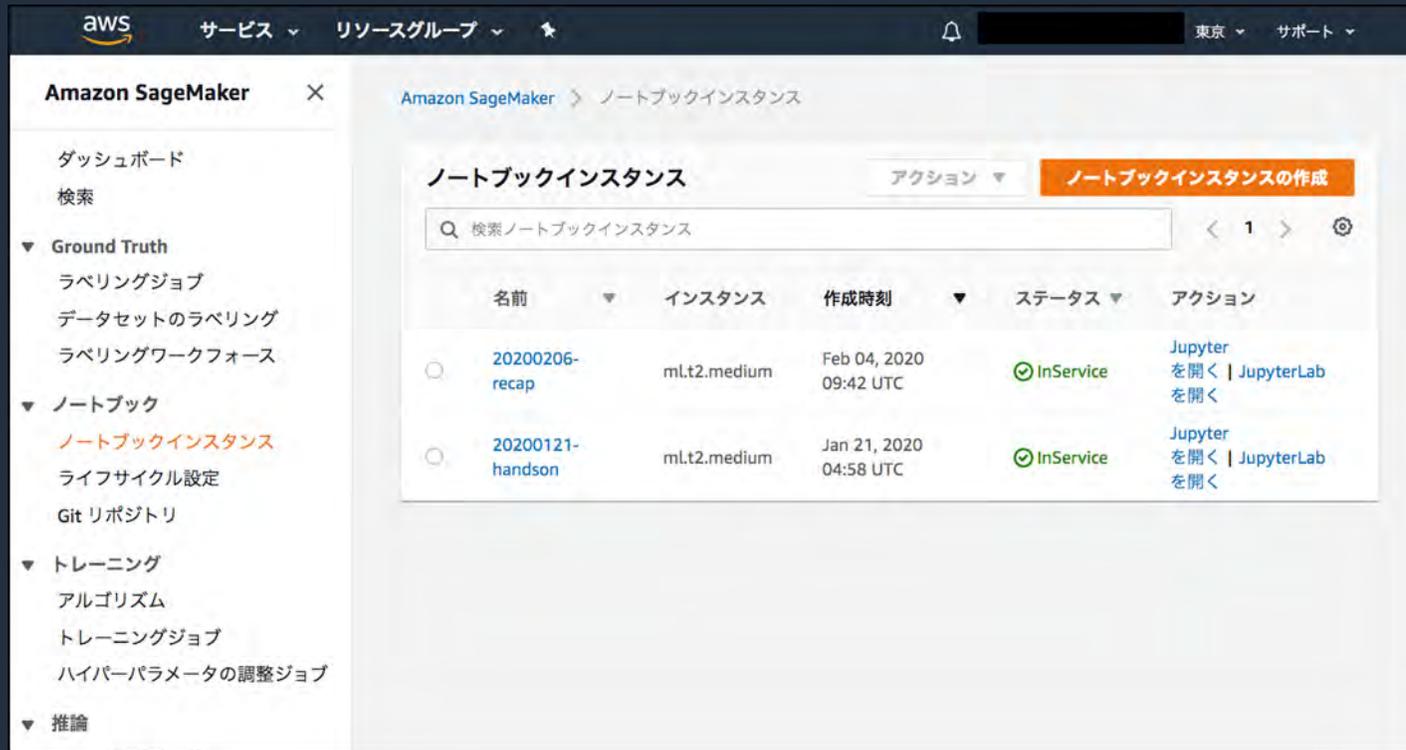


モデルの開発時の課題

- 共同開発が可能な環境を提供したい
- 柔軟に効率的に開発を行いたい

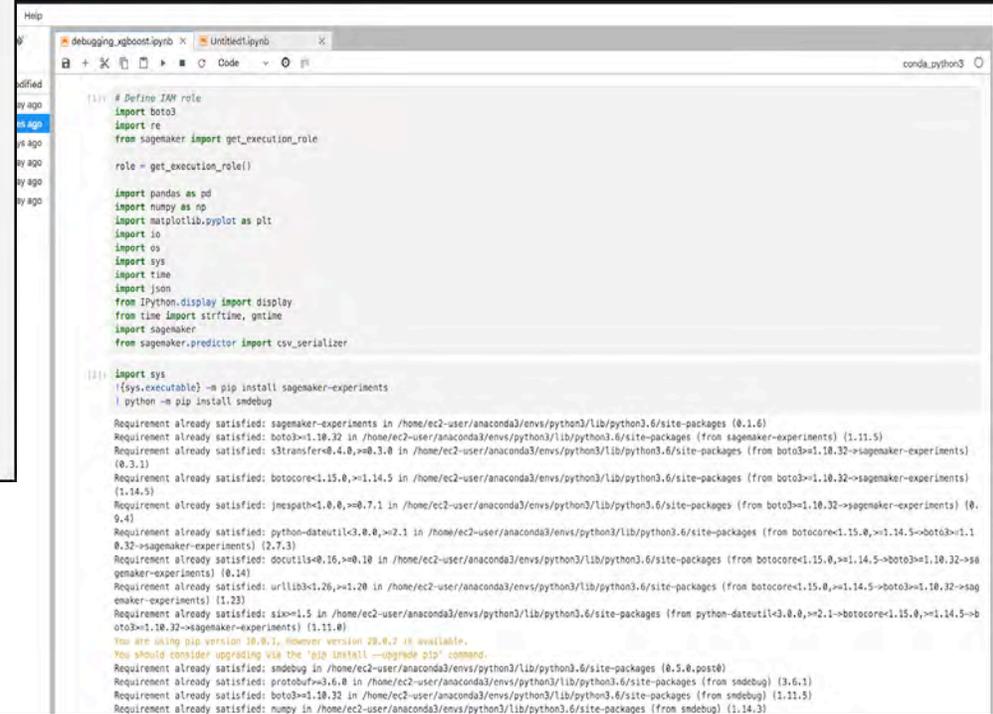
Amazon SageMaker ノートブックインスタンス

- マネージドな Jupyter 環境を提供



The screenshot shows the Amazon SageMaker console interface. The left sidebar contains navigation options such as 'ダッシュボード', 'ラベリングジョブ', 'ノートブック', and 'トレーニング'. The main content area displays a table of notebook instances.

名前	インスタンス	作成時刻	ステータス	アクション
20200206-recap	ml.t2.medium	Feb 04, 2020 09:42 UTC	InService	Jupyter を開く JupyterLab を開く
20200121-handson	ml.t2.medium	Jan 21, 2020 04:58 UTC	InService	Jupyter を開く JupyterLab を開く



The screenshot shows a Jupyter Notebook code cell with Python code and its output. The code imports various libraries and performs a pip install command.

```
from IPython.display import display
from time import strftime, localtime
import datetime
import sys
import os
import io
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

role = get_execution_role()

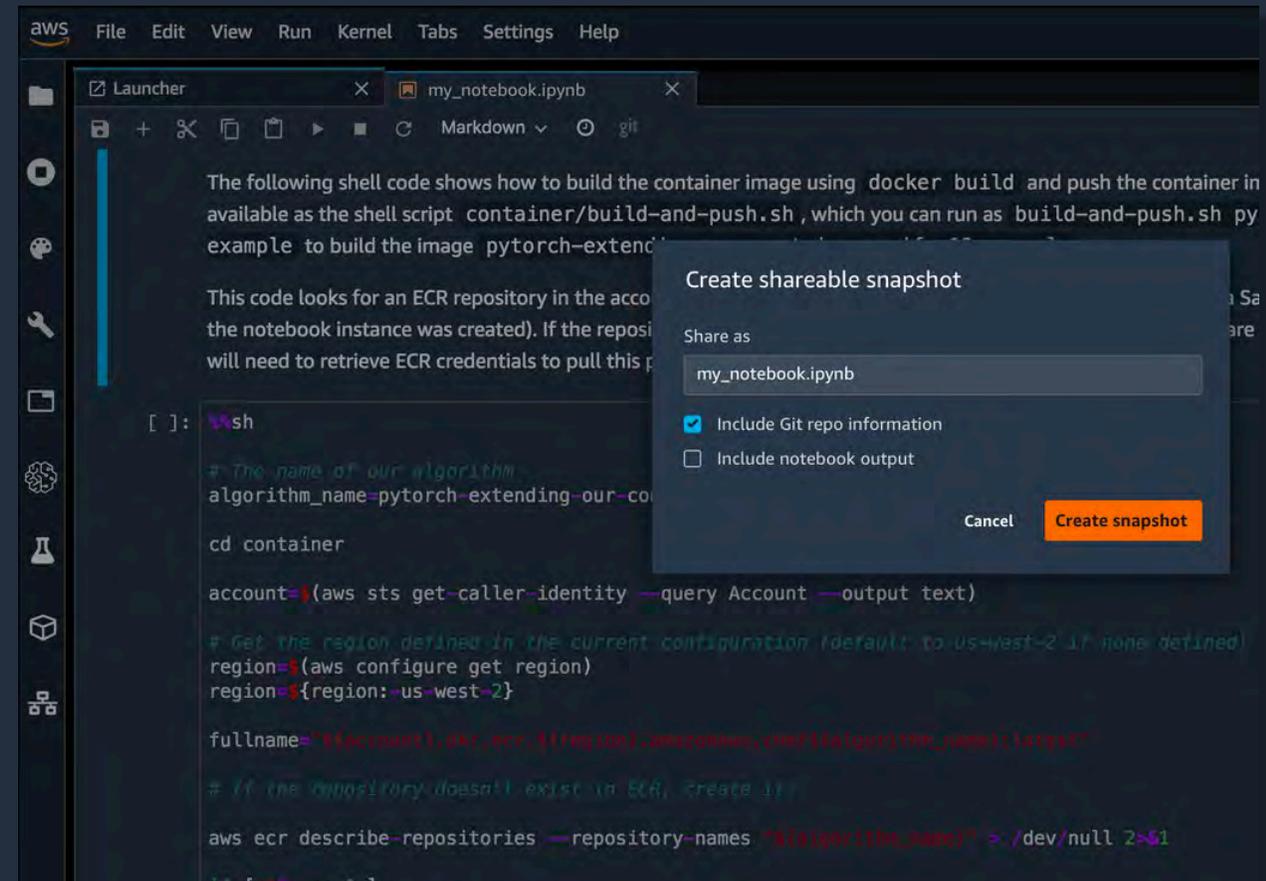
import sys
!{sys.executable} -m pip install sagemaker-experiments
! python -m pip install sagemaker-experiments
```

The output shows the installation progress for various packages, including sagemaker-experiments, boto3, s3transfer, botocore, jmespath, python-dateutil, docutils, urllib3, six, and numpy.

Amazon SageMaker Notebooks

Preview

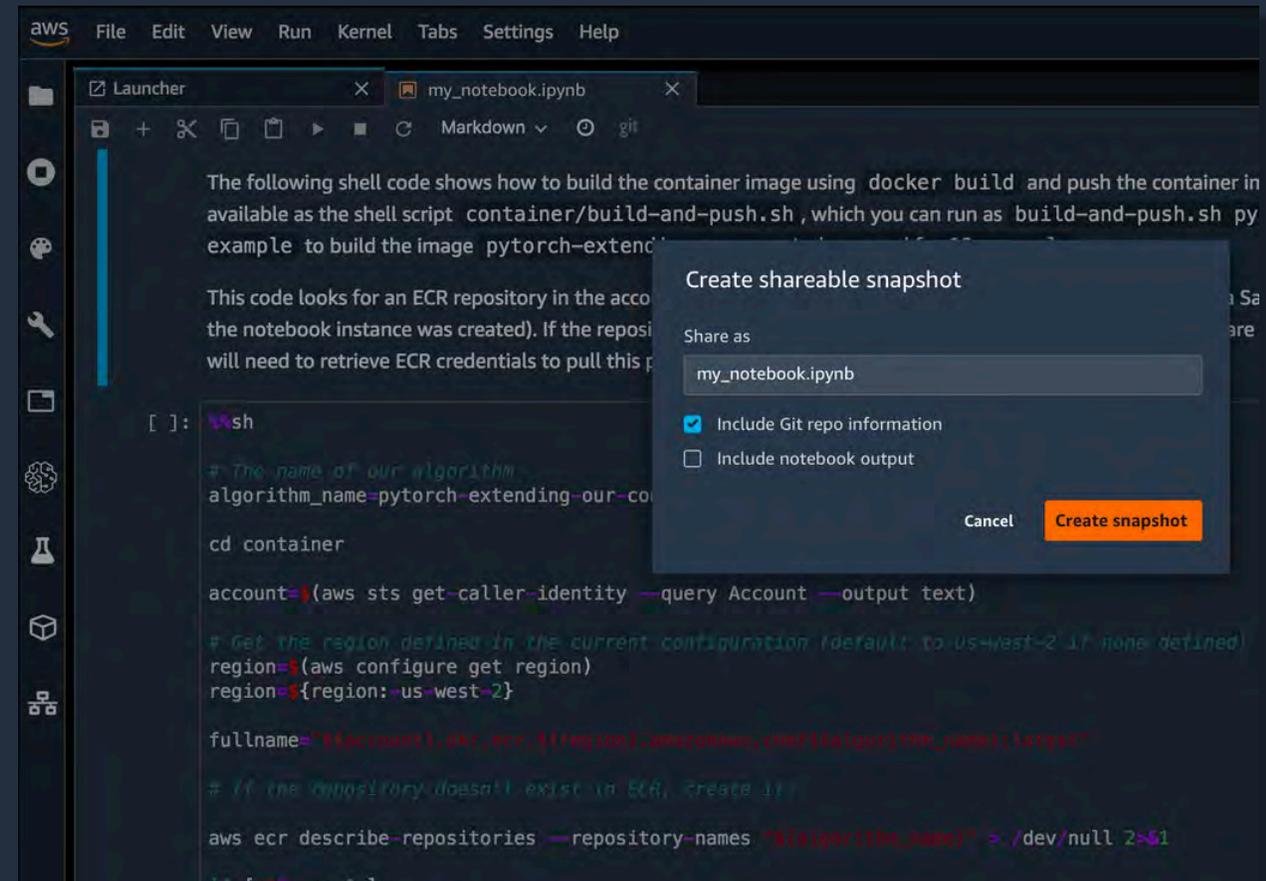
- AWS SSO でのアクセスが可能なノートブック
- 待ち時間なくワンクリックでノートブックを起動
- リソースの拡大・縮小はバックエンドで自動的に行われる
- SSOのアプリケーションにユーザーを登録し共同作業が可能に
- オハイオリージョンで preview



Amazon SageMaker Notebooks

Preview

- AWS SSO でのアクセスが可能なノートブック
- 待ち時間なくワンクリックでノートブックを起動
- リソースの拡大・縮小はバックエンドで自動的に行われる
- SSOのアプリケーションにユーザーを登録し共同作業が可能に
- オハイオリージョンで preview



Amazon SageMakerでの ML アルゴリズム開発

1. SageMaker の組み込みアルゴリズムを使用する
2. SageMaker で準備しているDL/ML用コンテナを使用する
3. 使用するライブラリに応じて独自のコンテナを使用する

ビルトインアルゴリズム

画像

- Image Classification
- Object Detection
- Semantic Segmentation

自然言語

- seq2seq
- LDA
- Neural Topic Model
- BlazingText

回帰・分類ほか

- Linear Learner
- XGBoost
- k-means
- k-NN
- PCA

数値・系列解析

- DeepAR Forecasting
- Random Cut Forest
- IP Insights

レコメンデーション

- Object2Vec
- Factorization Machines

SageMaker でサポートする DL/ML コンテナ一覧

	フレームワーク	SageMaker container サポートバージョン
Deep learning	TensorFlow	<u>Legacy mode</u> : 1.4.1, 1.5.0, 1.6.0, 1.7.0, 1.8.0, 1.9.0, 1.10.0 <u>Script mode</u> : 1.11.0, 1.12.0, 1.13.0, 1.14.0, 1.15.0, 2.0.0
	MXNet	0.12.1, 1.0.0, 1.1.0, 1.2.1, 1.3.0, 1.4.0, 1.4.1
	Chainer	4.0.0, 4.1.0, 5.0.0
	PyTorch	0.4.0, 1.0.0, 1.1.0, 1.2.0, 1.2.0, 1.3.1
ML	scikit-learn	0.20.0

<https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker>

TensorFlow: <https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker/tensorflow>

Chainer: <https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker/chainer>

PyTorch: <https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker/pytorch>

MXNet: <https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker/mxnet>

Sklearn: <https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker/sklearn>

独自で開発されたコンテナも活用可能です

※ 2020年1月時点

機械学習ワークフローの中での課題感

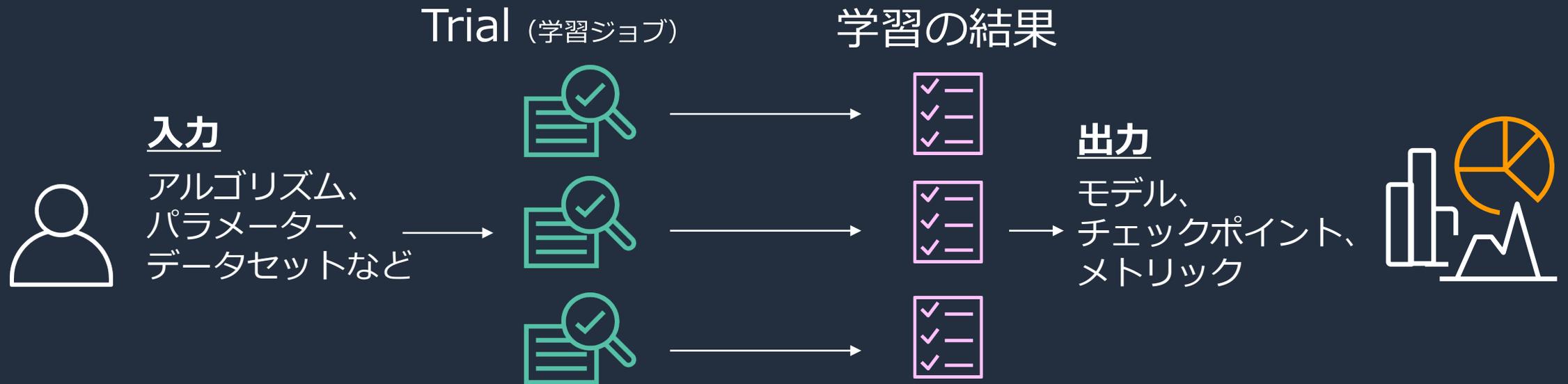


データ前処理/学習/評価・改良時の課題

- 大量の試行錯誤における実験管理を楽にしたい
- 機械学習モデルのデバックや分析を行いたい
- 前処理/後処理に必要なバッチ処理を効率的に行いたい
- 適切な学習リソースを活用したい
- 分散学習を手軽に行いたい
- 大規模なデータの学習へ対応したい
- パラメーター最適化を自動化したい

Amazon SageMaker Experiments

- SageMaker Experiments SDK
- 学習を改善するための試行錯誤 (実験) の結果を管理する機能
- 複数の異なる学習を1つの実験と紐付け、メトリクスなどを一元管理
- 実験の結果を、pandas や matplotlib を使って確認できる



サンプルノートブックからの抜粋

一連の Trial の結果を DataFrame にして一覧表示

TrialComponentName	DisplayName	SourceArn	dropout	epochs	hidden_channels	optimizer	test:accuracy - Min	test:accuracy - Max	test:accuracy - Avg	test:acc - S
0	cnn-training-job-1576391108-aws-training-job	Training	0.2	2.0	32.0	"sgd"	95.0	97.0	96.0	1.4
1	cnn-training-job-1576390823-aws-training-job	Training	0.2	2.0	20.0	"sgd"	96.0	97.0	96.5	0.7
2	cnn-training-job-1576390538-aws-training-job	Training	0.2	2.0	10.0	"sgd"	95.0	95.0	95.0	0.0
3	cnn-training-job-1576390029-aws-training-job	Training	0.2	2.0	2.0	"sgd"	95.0	95.0	95.0	0.0
4	cnn-training-job-1576390254-aws-training-job	Training	0.2	2.0	5.0	"sgd"	94.0	94.0	94.0	0.0

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

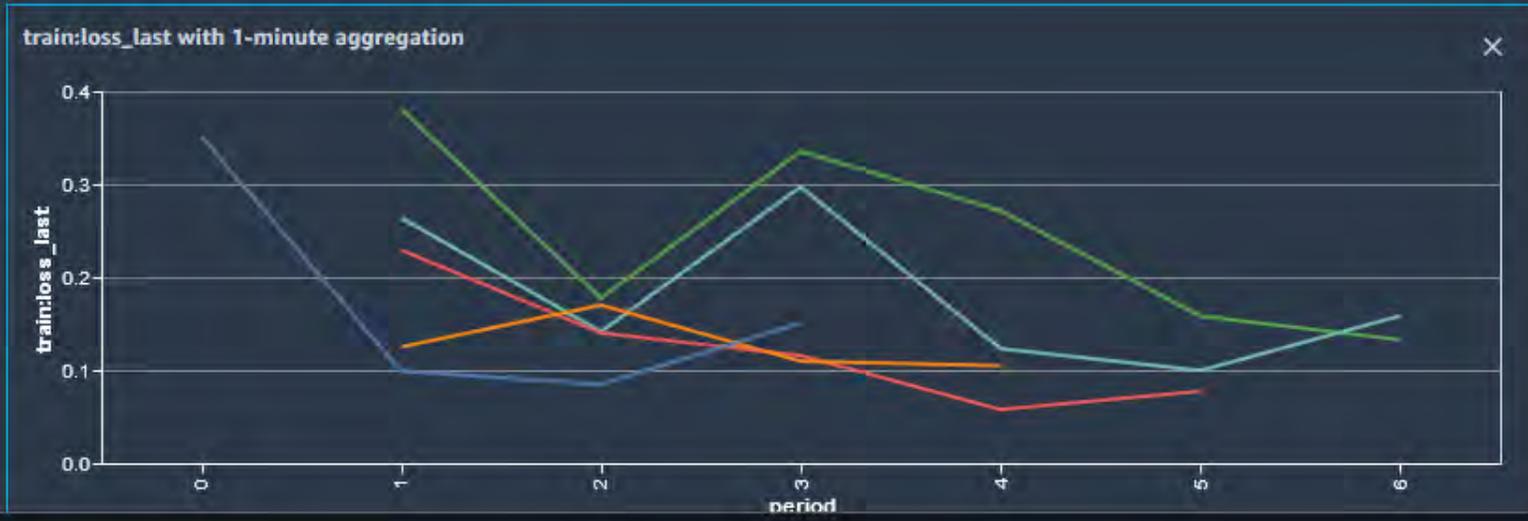


CHART PROPERTIES

- 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



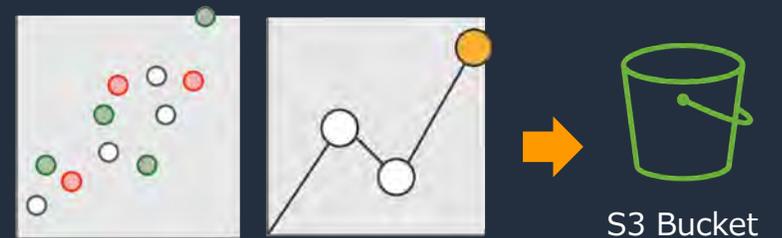
Amazon SageMaker Experiments

Track and organize thousands of ML experiments.

[Learn more >>](#)

Amazon SageMaker Debugger

- Amazon SageMaker Debugger SDK
- 学習時の異常出力等の問題を検出
- 問題検出のルールがあらかじめ与えられ（追加も可能）、S3 に出力される学習の途中計算結果（テンソル）からデバッグ
- TensorFlow, Keras, Apache MXNet, PyTorch, XGBoost に対応
- SageMakerが使える全てのリージョンで提供



```
Python
description = client.describe_training_job(TrainingJobName=training_job_name)
print(description['DebugRuleEvaluationStatuses'][0]['RuleEvaluationStatus'])
print(description['DebugRuleEvaluationStatuses'][0]['RuleEvaluationStatus'])

ExplodingTensor
IssuesFound
```

```
Python
tensor = 'gradients/multiply/MatMul_1_gradients'
for s in list(trial.tensor(tensor).steps()):
    print("Value: ", trial.tensor(tensor).steps()[s].value)

Value: [[1.1508383e+23] [1.0809098e+23]]
Value: [[1.0278440e+23] [1.1347468e+23]]
Value: [[nan] [nan]]
Value: [[nan] [nan]]
Value: [[nan] [nan]]
Value: [[nan] [nan]]
```

SMDEbugger-CloudWatch-Lo

conda_tensorflow_p36

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, decrease 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='...', # 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 file
                           rule_to_invoke='CustomGradientRule', # name of the class
                           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 initialize
                           } ) ],
    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 you
# The rule evaluation status(es) will be visible in the training logs
# at regular intervals

estimator.fit(wait=False)
```

Result

Describe Trial Component

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

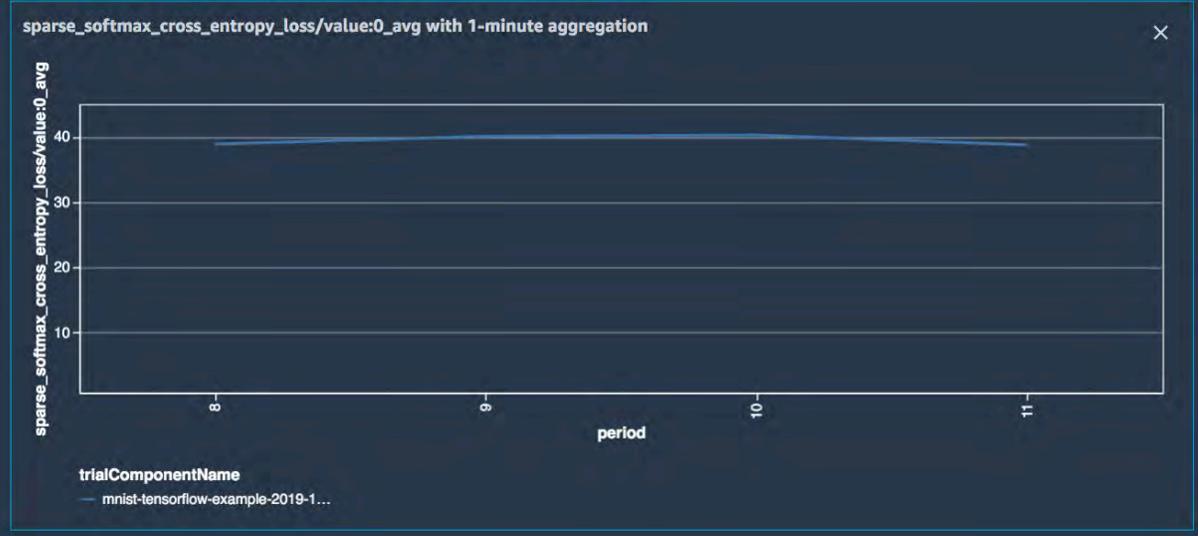
Status	Last modified	Rule name	Job ARN
Issues Found	4 minutes ago	VanishingGradient	arn:aws:sagemaker:us-west-2:3
Issues Found	4 minutes ago	Overfitting	arn:aws:sagemaker:us-west-2:3

Trial Component Chart

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



Amazon SageMaker Processing

- SageMaker Processing SDK
- 機械学習で必要となるデータの前処理・後処理をバッチで行う
- 学習・テストデータの一括整形、推論結果からのフィルタリングなど
- 組み込み scikit-learn コンテナでデータ処理 (コンテナ持ち込みも可能)
- 処理開始時にインスタンスが起動、処理終了時にインスタンスも終了



<https://aws.amazon.com/jp/blogs/news/amazon-sagemaker-processin-fully-managed-data-processing-and-model-evaluation/>

Processing.py

Python

```
import pandas as pd
from sklearn.model_selection import train_test_split
# Read data locally
df = pd.read_csv('/opt/ml/processing/input/dataset.csv')
# Preprocess the data set
downsampled = apply_mad_data_science_skills(df)
# Split data set into training, validation, and test
train, test = train_test_split(downsampled, test_size=0.2)
train, validation = train_test_split(train, test_size=0.2)
# Create local output directories
try:
    os.makedirs('/opt/ml/processing/output/train')
    os.makedirs('/opt/ml/processing/output/validation')
    os.makedirs('/opt/ml/processing/output/test')
except:
    pass
# Save data locally
train.to_csv("/opt/ml/processing/output/train/train.csv")
validation.to_csv("/opt/ml/processing/output/validation/validation.csv")
```

オンプレミス環境からの SageMaker 活用

- オンプレミス環境で SageMaker を動かすことで、Docker, AWS SDK, SageMaker Python SDK をインストール SageMakerの学習・推論ジョブを実行可能
- `train_instance_type='local_gpu'` でローカルGPUで学習を実行
- オンプレミスとのハイブリッドなSageMaker環境が作成できる

```
container = get_image_uri(  
    boto3.Session().region_name,  
    'xgboost'  
)  
  
xgb = sagemaker.estimator.Estimator(  
    container,  
    role,  
    train_instance_count=1,  
    train_instance_type='local_gpu',  
    sagemaker_session=sess,  
    output_path=config['s3_output_location']  
)  
  
xgb.fit(inputs=  
    {'train': s3_input_train, 'validation': s3_input_validation},  
    job_name=job_name  
)
```

学習のコスト低減のための Managed Spot Training

- これまでの学習コストを最大で90%削減
- すべてのフレームワーク, モデル, 学習構成で利用可能
- Checkpointing により Spot instance が落ちても途中から学習を再開

```
from sagemaker.tensorflow import TensorFlow

mnist_estimator = TensorFlow(entry_point='mnist.py',
                             role=role,
                             train_instance_count=2,

                             train_instance_type='ml.p2.xlarge',
                             train_max_run = 5000,

                             train_use_spot_instances=True,
                             train_max_wait = 10000)
```

```
In [8]: mnist_estimator.fit(training_data_uri)
```

```
Placeholder is deprecated. Please use tf.compat.v1.placeholder instead.
W0828 14:00:47.032024 140424720971520 training.py:181] Model not saved.
Model. Your training job will not save any model files to S3.
For details of how to construct your training script see:
https://github.com/aws/sagemaker-python-sdk/tree/master/src/sagemaker
tensorflow-script
```

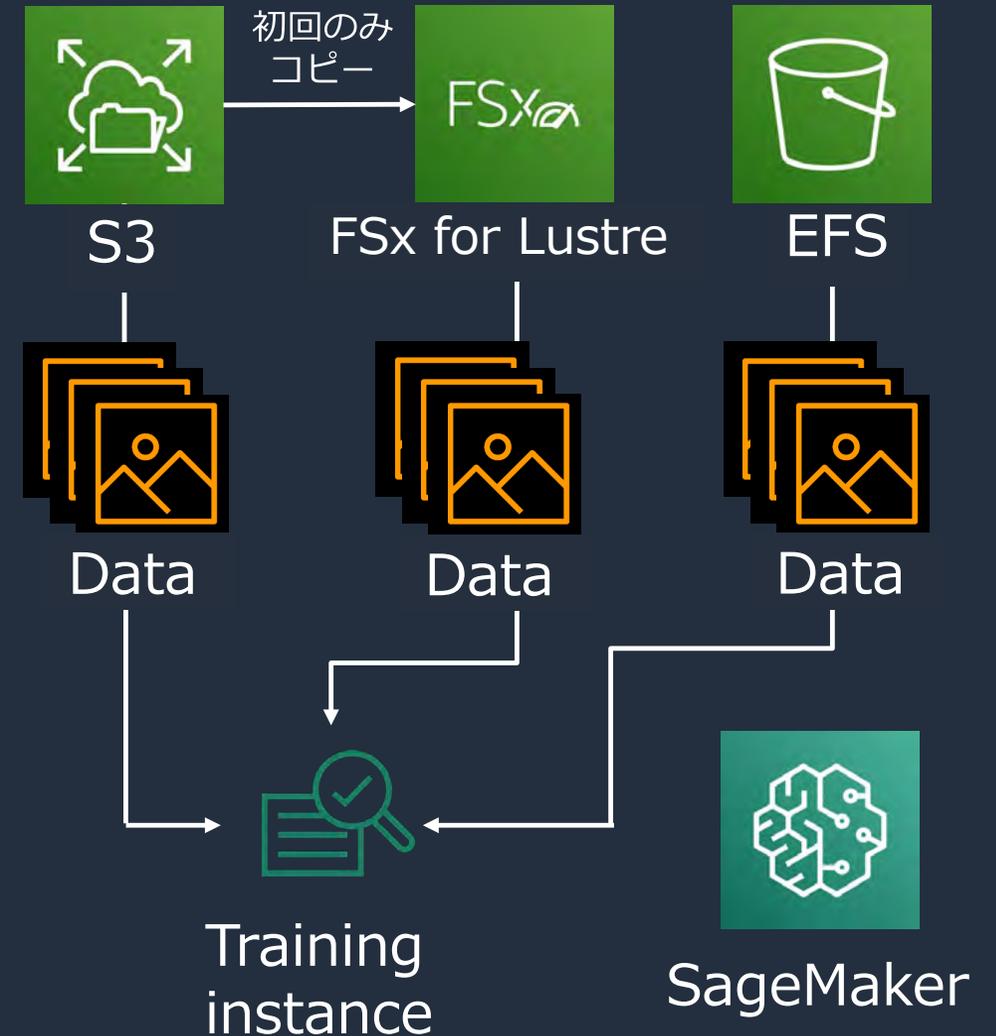
```
2019-08-28 14:00:53 Uploading - Uploading generated training m
2019-08-28 14:00:53 Completed - Training job completed
Training seconds: 1544
Billable seconds: 620
Managed Spot Training savings: 59.8%
```

<https://aws.amazon.com/jp/blogs/aws/managed-spot-training-save-up-to-90-on-your-amazon-sagemaker-training-jobs/>

大量学習データの高速な読み出し

- 学習用のファイルシステムとして, FSx for Lustre, EFS を直接指定可能に
- 学習ジョブの開始時に S3から学習インスタンスのEBSへのデータ転送が不要, 学習を高速化できる
- 同一ファイルを用いた反復の学習ジョブでは, 高速キャッシュとしてファイルシステムを利用できる

<https://aws.amazon.com/jp/blogs/machine-learning/speed-up-training-on-amazon-sagemaker-using-amazon-efs-or-amazon-fsx-for-lustre-file-systems/>



ハイパーパラメータ最適化

- Deep Learning では、多くのパラメータを調整する必要がある
 - 学習率
 - バッチサイズ
 - エポック数...
- 学習ジョブの並列実行を活用し、最適なパラメータを高速に探索
- デフォルトは**ベイズ最適化**
- **ランダム探索**も選択可能に
- HPO ウォームスタートも利用可能

```
from sagemaker.chainer import Chainer

estimator = Chainer(entry_point="mnist.py",
                    role=role,
                    framework_version='5.0.0',
                    train_instance_count=1,
                    train_instance_type='ml.m4.xlarge',
                    hyperparameters={'epochs': 30, 'batch-size': 256})

hyperparameter_ranges = {'lr': ContinuousParameter(0.001, 0.1),
                          'batch-size': CategoricalParameter([32,64,128,256,512])}

tuner = HyperparameterTuner(estimator,
                             objective_metric_name,
                             hyperparameter_ranges,
                             metric_definitions,
                             max_jobs=9,
                             max_parallel_jobs=3,
                             objective_type=objective_type)

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

機械学習ワークフローの中での課題感



データ前処理/学習/評価・改良時の課題

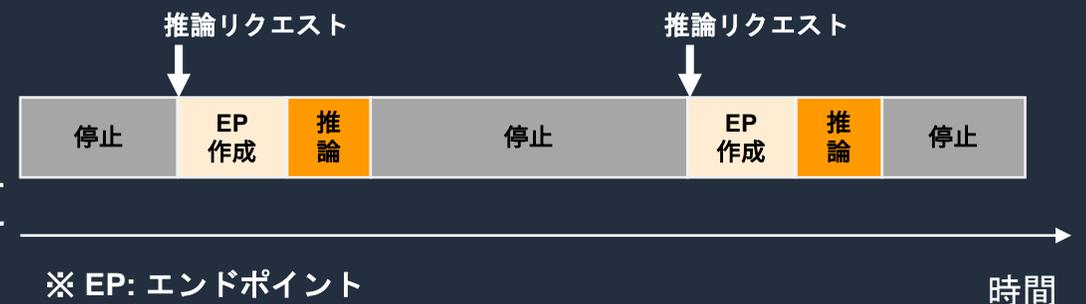
- 機械学習エンジニアやデータサイエンティストへのデプロイの負荷を低減したい
- 機械学習モデルの推論コストを低減したい
- 推論を行っている機械学習モデルをモニタリングしたい
- 推論を行っている機械学習モデルの性能を継続的に向上させたい

リアルタイム推論

- リアルタイムな推論を必要とする場合に選択
- `deploy()` でエンドポイントを構築できる
- Web API の URL が発行され, URL にリクエストを送り推論
- エンドポイント起動中は料金がかかるので注意

バッチ変換ジョブ (バッチ推論)

- リアルタイム性が求められない推論向き
- `transform()` でエンドポイントを起動
- 推論が終わるとエンドポイントが自動的に削除される
- 推論時のみインスタンスを起動するため, コスト効率高

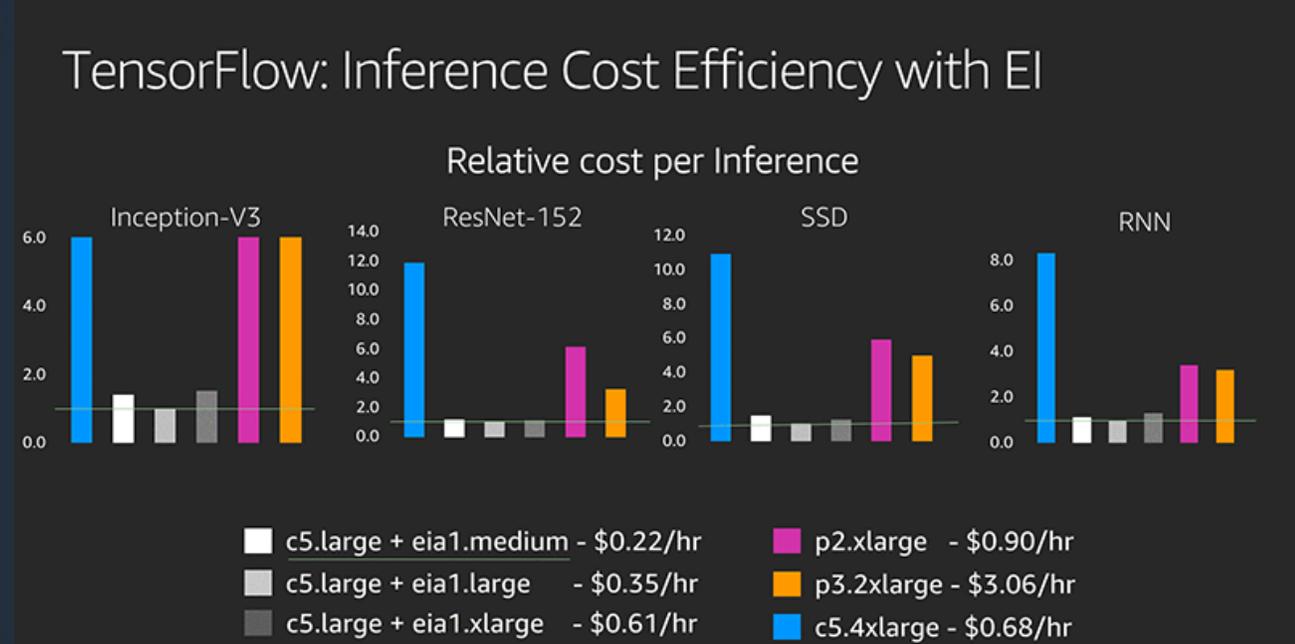


推論環境のコスト最適化

Elastic Inference

- 推論用CPUにスライスGPU をアタッチすることで、コスト効率よく推論
- predictor のアクセレレーター定義により、TF, MXNet, ONNX ですぐに使える

<https://aws.amazon.com/jp/blogs/news/optimizing-costs-in-amazon-elastic-inference-with-amazon-tensorflow/>



SageMaker Neo

- モデルを推論環境に合わせコンパイル、デバイス上のメモリ使用量削減、推論高速化
- フレームワーク依存のないランタイム、TensorFlow, MXNet, PyTorch, ONNX, XGBoost が対応
- EC2, Rasp3b, Jetson に加え、rk3288, rk3299, QC603, QC605, Jetson Nano などターゲットデバイスも追加

<https://neo-ai-dlr.readthedocs.io/en/latest/install.html#table-1-list-of-supported-devices>

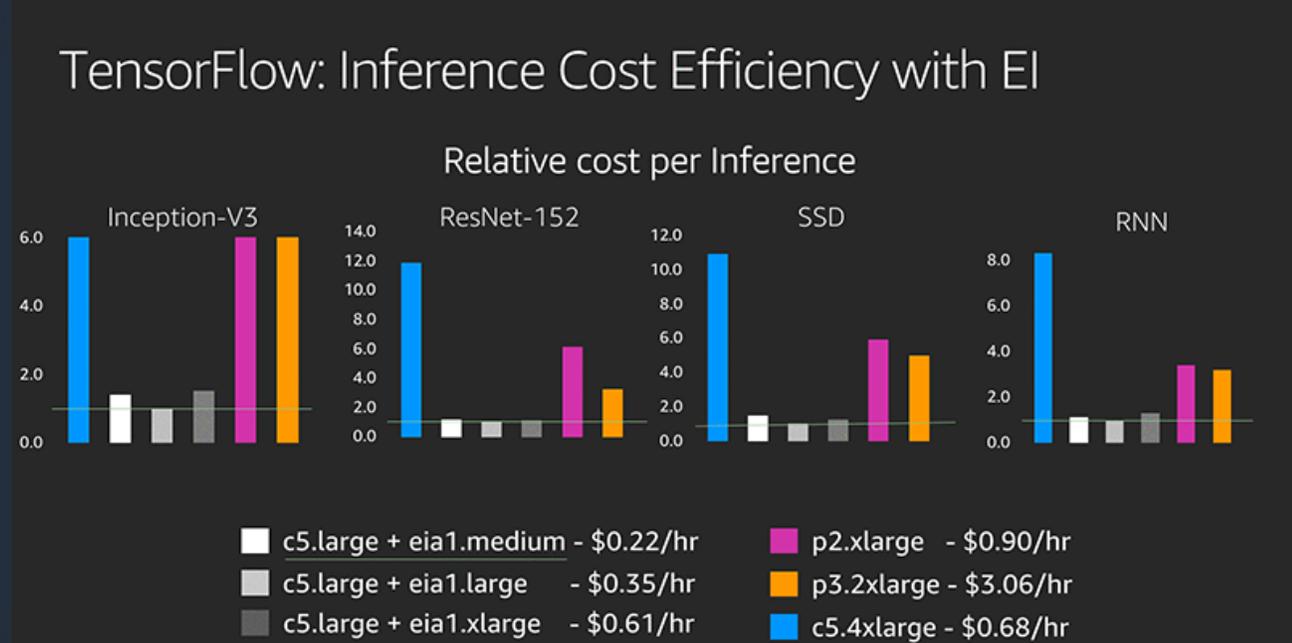
<https://aws.amazon.com/blogs/machine-learning/running-amazon-elastic-inference-workloads-on-amazon-ecs/>

推論環境のコスト最適化

Elastic Inference

- 推論用CPUにスライスGPU をアタッチすることで、コスト効率よく推論
- predictor のアクセレレーター定義により、TF, MXNet, ONNX ですぐに使える

<https://aws.amazon.com/jp/blogs/news/optimizing-costs-in-amazon-elastic-inference-with-amazon-tensorflow/>



SageMaker Neo

- モデルを推論環境に合わせコンパイル、デバイス上のメモリ使用量削減、推論高速化
- フレームワーク依存のないランタイム、TensorFlow, MXNet, PyTorch, ONNX, XGBoost が対応
- EC2, Rasp3b, Jetson に加え、Jetson Nano, rk3288, rk3299, QC603, QC605 などターゲットデバイスも追加

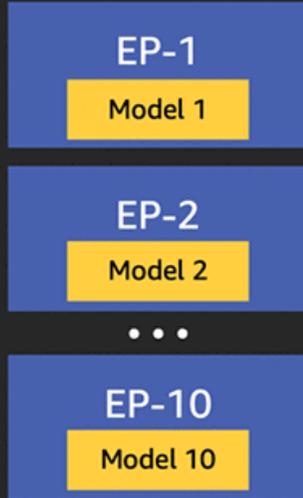
<https://neo-ai-dlr.readthedocs.io/en/latest/install.html#table-1-list-of-supported-devices>

<https://aws.amazon.com/blogs/machine-learning/running-amazon-elastic-inference-workloads-on-amazon-ecs/>

マルチモデルエンドポイント(MME)の活用

- 一つのエンドポイントの裏で複数のモデルを切り替えて推論コストを削減
- 学習済みモデルは S3にストアしてエンドポイントを動的に切り替え可能に

10 separate endpoints
\$3,430/month

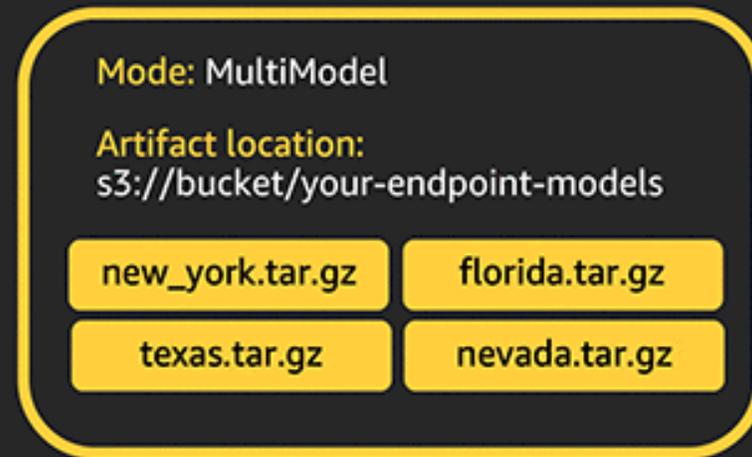


predict

```
predict('nevada.tar.gz',  
features)
```



Amazon SageMaker
Multi-model endpoint



S3 model storage

s3://bucket/your-endpoint-models/

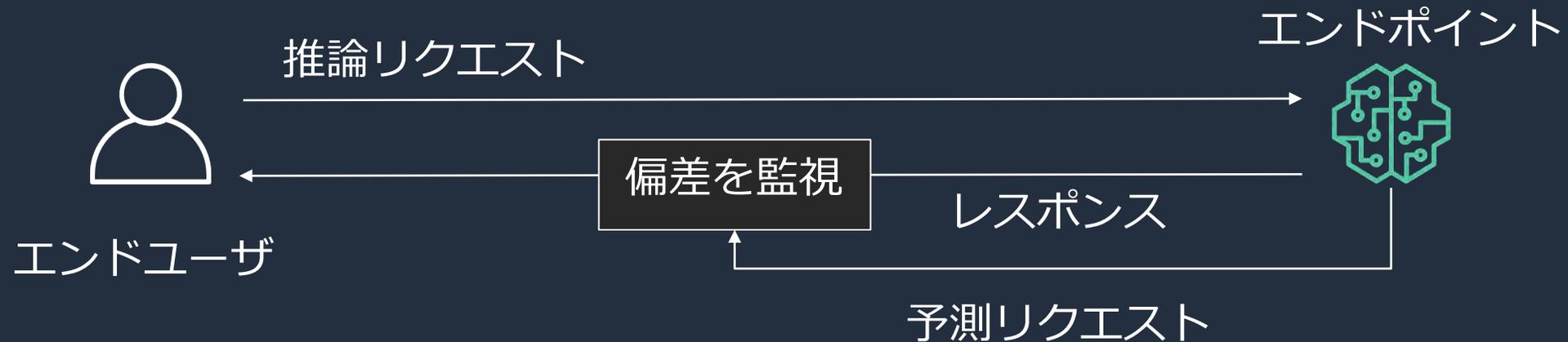
load

new_york.tar.gz
florida.tar.gz
texas.tar.gz
nevada.tar.gz

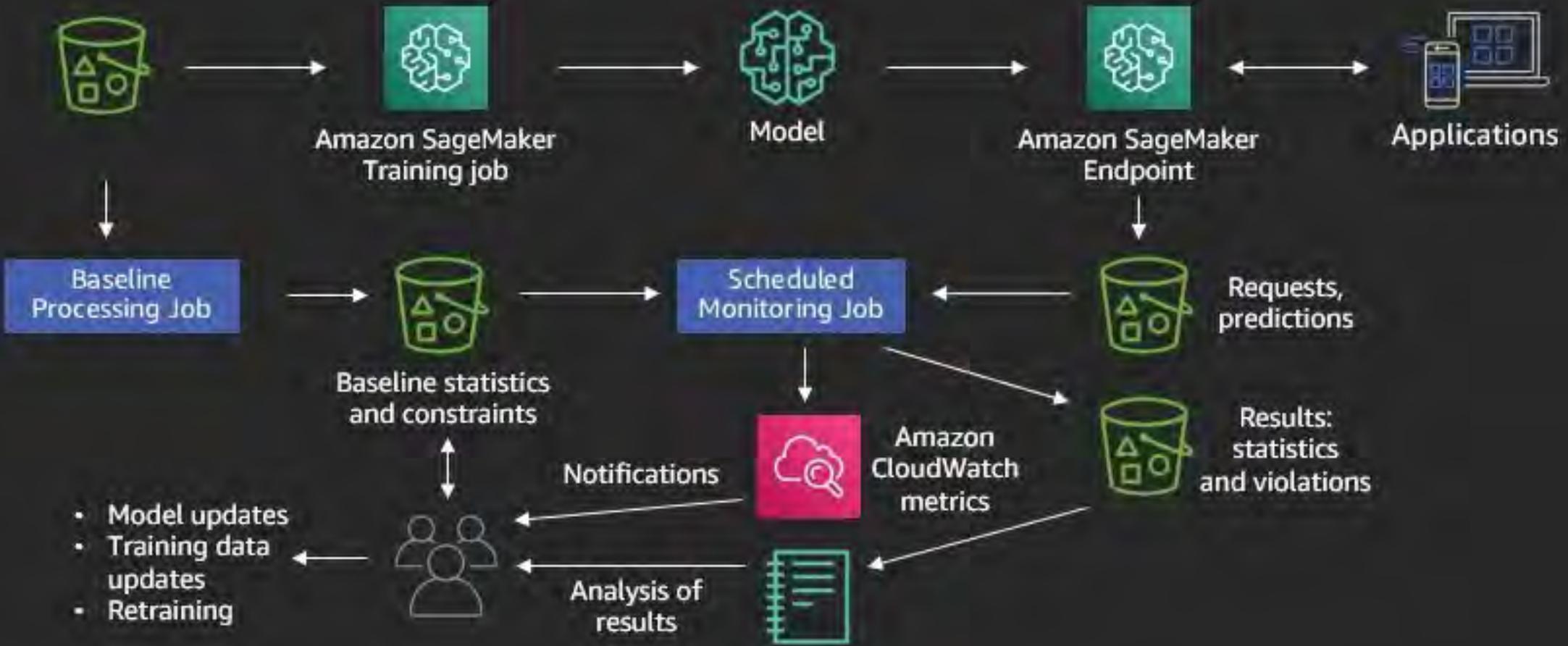
<https://aws.amazon.com/blogs/machine-learning/save-on-inference-costs-by-using-amazon-sagemaker-multi-model-endpoints/>

Amazon SageMaker Model Monitor

- MLモデルの品質を維持する機能
- SageMaker の推論エンドポイントの入出力を継続的に監視し、学習時に用いたデータをベースにした予測リクエストと実際のリクエストに乖離がないかを評価する
- 乖離がある場合は、モデルが正しく推論できていない可能性があり、最新のデータに合わせて再学習を行う必要がある



Amazon SageMaker Model Monitor



ENDPOINTS

Search endpoints

Name	Monitoring status
Endpoint-01	Issues
Endpoint-02	No issues
Endpoint-03	Issues
Endpoint-04	No issues
Endpoint-05	Issues
Endpoint-06	No issues
Endpoint-07	Issues

Showing all rows Last updated seconds ago

Endpoint-05

Monitoring results Monitoring job history Endpoint configuration

MODEL QUALITY MONITORING

Monitor your machine learning models in production to detect data drift, data quality issues, anomalies, and deviation in model quality. [Learn more](#)

Schedule: Schedule_1

Search by features

Timeline 1h 6h 12h 1d 1w

Feature 1: Completeness

Date and time	Baseline	Current
Sept 01 12:00	0.81	0.79
Sept 02 12:00	0.81	0.80
Sept 03 12:00	0.81	0.81

Feature 3: Completeness

Date and time	Baseline	Current
Sept 01 12:00	0.81	0.79
Sept 02 12:00	0.81	0.80
Sept 03 12:00	0.81	0.81

SageMaker Model Monitor – 結果の可視化

Distributions

This section visualizes the distribution and renders the distribution statistics for all features

```
[22]: features = mu.get_features(execution_statistics)
      feature_baselines = mu.get_features(baseline_statistics)
```

```
[*]: mu.show_distributions(features)
```

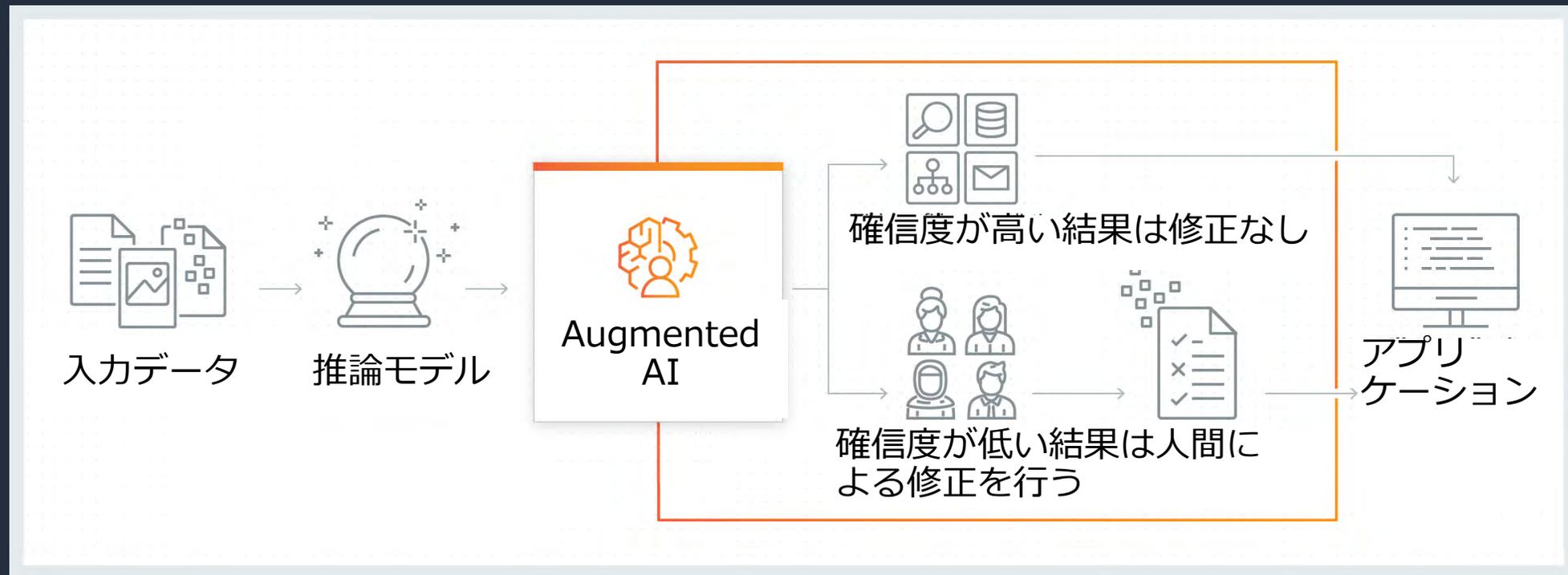
Numerical Features

	num_present	num_missing	mean	sum	std_dev	min	max
Churn	3399	0	0.138887	472.077477	0.261824	0.006144	0.989564
Account Length	3399	0	100.045013	340053.000000	39.352781	10.000000	232.000000
VMail Message	3399	0	7.893498	26830.000000	13.371318	0.000000	44.000000
Day Mins	3399	0	177.529362	603422.300000	55.288781	29.900000	328.100000
Day Calls	3399	0	101.556929	345192.000000	19.375790	36.000000	158.000000
...
Area Code_510	3399	0	0.258900	880.000000	0.438836	0.000000	1.200000
Int'l Plan_no	3399	0	0.923213	3138.000000	0.268400	0.000000	1.300000

Amazon Augmented AI

Preview

- 推論結果を人間が修正するワークフローを構築する機能
- Ground Truth と同様、テンプレートを構築してワーカーにレビュー依頼



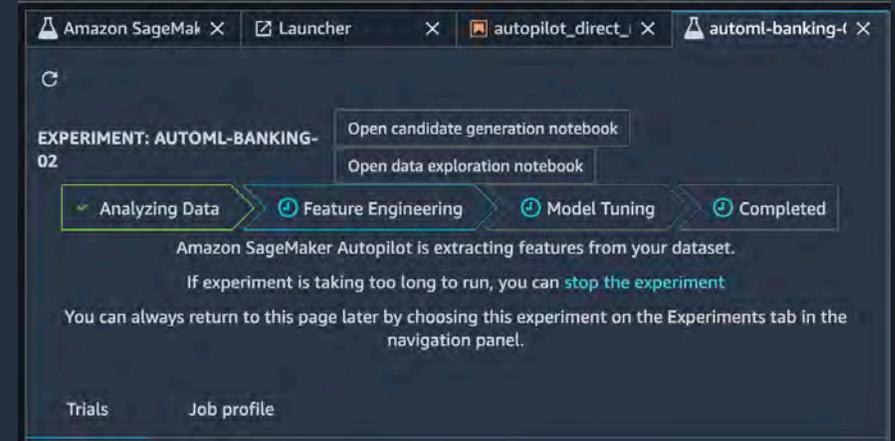
Amazon SageMaker での AutoML



Amazon SageMaker Autopilot

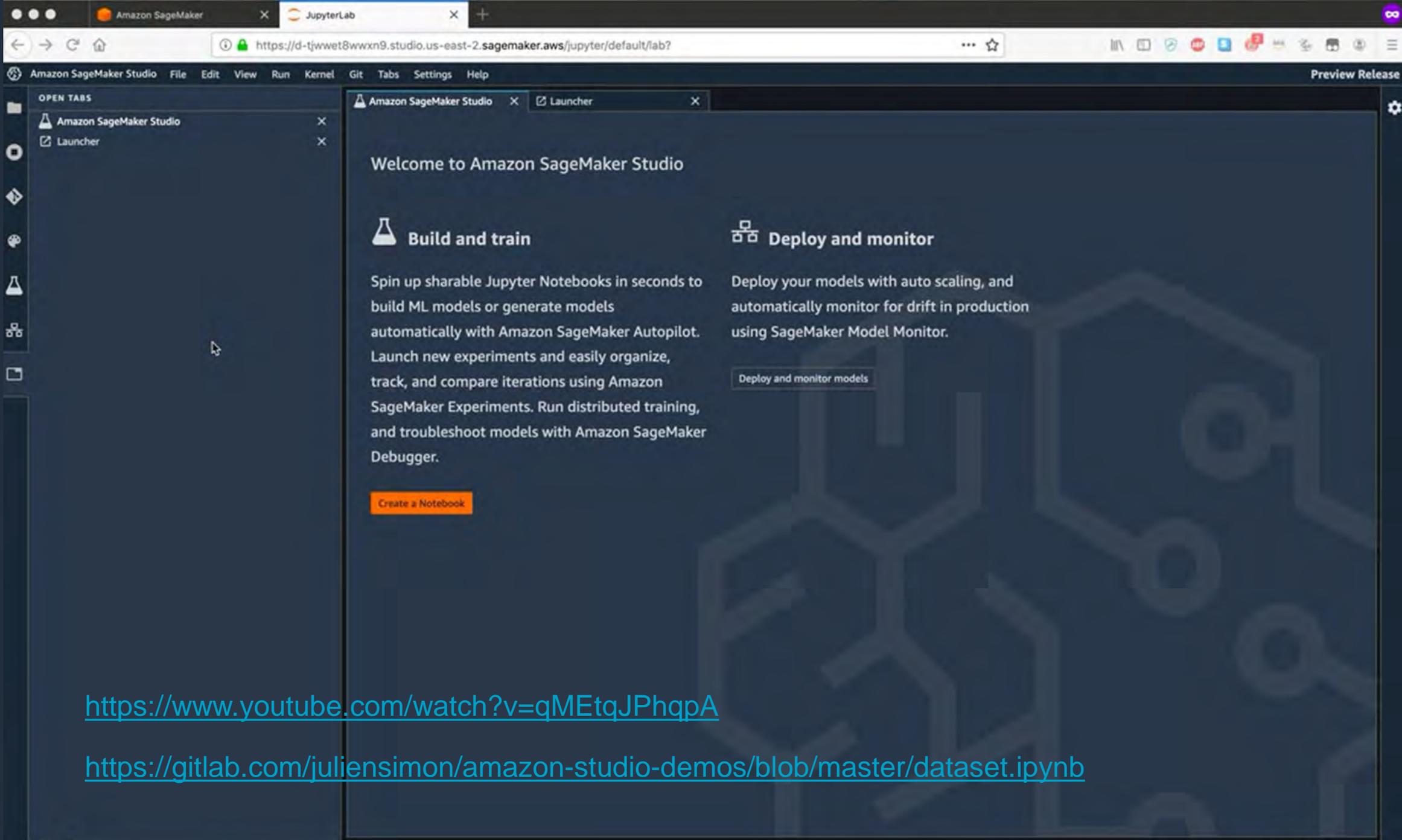
- **テーブルデータ**に対する予測と分類の機械学習のAutoML機能
- 不正検知、ユーザ分類（離反予測）、レコメンデーションなど多数のユースケースに適用できる
- データソースと予測対象を指定
- データの前処理、アルゴリズムの選択、インスタンスタイプ選択なども自動化
- US4リージョン、東京など16リージョンで利用可能

SageMaker 上でAutoMLを簡単に実行



生データの検査や特徴量抽出

	age	job	marital	education	default	housing	loan
0	56	housemaid	married	basic.4y	no	no	no
1	57	services	married	high.school	unknown	no	no
2	37	services	married	high.school	no	yes	no
3	40	admin.	married	basic.6y	no	no	no
4	56	services	married	high.school	no	no	yes
5	45	services	married	basic.9y	unknown	no	no



<https://www.youtube.com/watch?v=qMEtqJPhqA>

<https://gitlab.com/juliensimon/amazon-studio-demos/blob/master/dataset.ipynb>

Navigation sidebar with icons for home, experiments, and tools.

EXPERIMENTS
0 row selected Create Experiment

Name	Last Modified
Unassigned trial components	9 seconds ago
marketing-automl-demo-aw...	7 minutes ago
automl-demo-reinvent-aws-...	6 hours ago
mnist-hand-written-digits-cl...	11 days ago

Experiment details for **MARKETING-AUTOML-DEMO**.

Open candidate generation notebook Open data exploration notebook

Progress: Analyzing Data (checked) → Feature Engineering → Model Tuning → Completed

Amazon SageMaker Autopilot is extracting features from your dataset.
If experiment is taking too long to run, you can [stop the experiment](#).

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

Trials | Job profile

You don't have any trials running.

The screenshot displays the Amazon SageMaker Studio interface. On the left, a navigation panel shows a list of experiments under the heading "EXPERIMENTS". The main content area is titled "EXPERIMENT: MARKETING-AUTOML-DEMO" and features a progress bar with four steps: "Analyzing Data", "Feature Engineering", "Model Tuning", and "Completed". Below the progress bar, the text reads: "Amazon SageMaker Autopilot is tuning the model. If experiment is taking too long to run, you can [stop the experiment](#). You can always return to this page later by choosing this experiment on the Experiments tab in the navigation panel." There are buttons for "Open candidate generation notebook" and "Open data exploration notebook".

Name	Last Modified
Unassigned trial components	2 minutes ago
marketing-automl-demo-aw...	9 seconds ago
automl-demo-reinvent-aws-...	6 hours ago
mnist-hand-written-digits-cl...	11 days ago

0 1 marketing-automl-demo

https://github.com/aws-labs/amazon-sagemaker-examples/blob/master/introduction_to_applying_machine_learning/xgboost_direct_marketing/xgboost_direct_marketing_sagemaker.ipynb

EXPERIMENTS

1 row selected Create Experiment

Name	Last Modified
Unassigned trial components	22 hours ago
marketing-automl-demo-aw...	20 hours ago
automl-demo-reinvent-aws-...	1 day ago
mnist-hand-written-digits-cl...	12 days ago

Amazon SageMaker Studio dataset.ipynb marketing-automl-demo

EXPERIMENT: MARKETING-AUTOML-DEMO Open candidate generation notebook Open data exploration notebook

Trial profile

TRIALS

0 row selected Deploy model

Trial name	Status	Start time	Objective
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9180830121040344
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9124600291252136
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.11258800327777863
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9176999926567078
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9141209721565247
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9007030129432678
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9012140035629272
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.8985300064086914
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9109269976615906
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.8987860083580017
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9151440262794495
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.892524003982544
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9191049933433533
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9060699939727783
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9130989909172058
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9083709716796875
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.11258800327777863
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.90031898021698
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9166769981384277
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.8985300064086914
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9125880002975464
marketing-au-tuning-job-1-bcbb...	Completed	20 hours ago	0.9076039791107178

No properties to inspect.

Amazon SageMaker と AWS サービス連携



Amazon SageMaker Operators for Kubernetes

- Kubernetes を利用している開発者が、Amazon SageMaker を利用して機械学習モデルの学習・デプロイするための機能
- Kubernetes API や kubectl などのツールを利用して SageMaker を呼び出し、SageMaker でジョブの作成・操作が可能
- 米国東部（オハイオ）、米国東部（バージニア北部）、米国西部（オレゴン）、および EU（アイルランド）AWS リージョンで一般的に利用可能

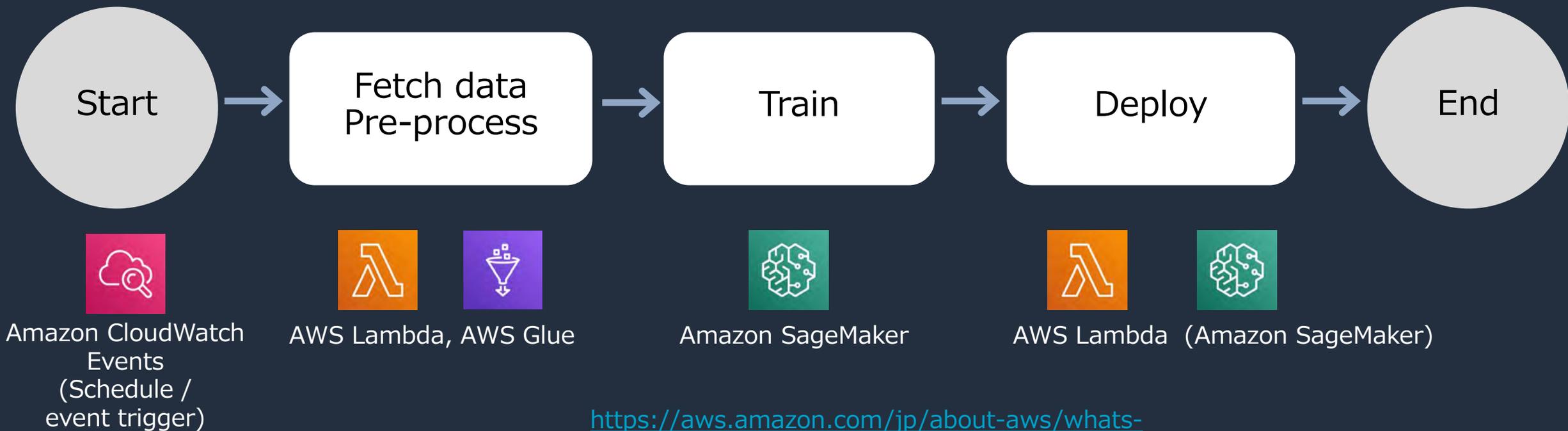


https://sagemaker.readthedocs.io/en/stable/amazon_sagemaker_operators_for_kubernetes.html

<https://aws.amazon.com/jp/blogs/news/introducing-amazon-sagemaker-operators-for-kubernetes/>

AWS Step Functions Data Science SDK

- AWS Lambda, Glue などにも対応したサーバーレスオーケストレーション
- [AWS Step Functions Data Science SDK](https://aws.amazon.com/jp/about-aws/whats-new/2019/11/introducing-aws-step-functions-data-science-sdk-amazon-sagemaker/) が登場、Pythonで前処理～学習～デプロイ～のデータサイエンスワークフローを作成し、視覚化できます



<https://aws.amazon.com/jp/about-aws/whats-new/2019/11/introducing-aws-step-functions-data-science-sdk-amazon-sagemaker/>

Amazon Athena で SQL クエリから ML 実行

- Athena を使って SQL クエリから直接推論を呼び出し可能に
- 機械学習モデルを
コンソール、API、JDBCドライバから呼び出し可能
- SageMaker で学習したモデルや Marketplace で購入した
モデルを SageMaker hosting service としてデプロイ、
AthenaからSQLで呼び出しが可能
- Amazon Athena の ML 機能は、
バージニア北部リージョンで現在 preview 中



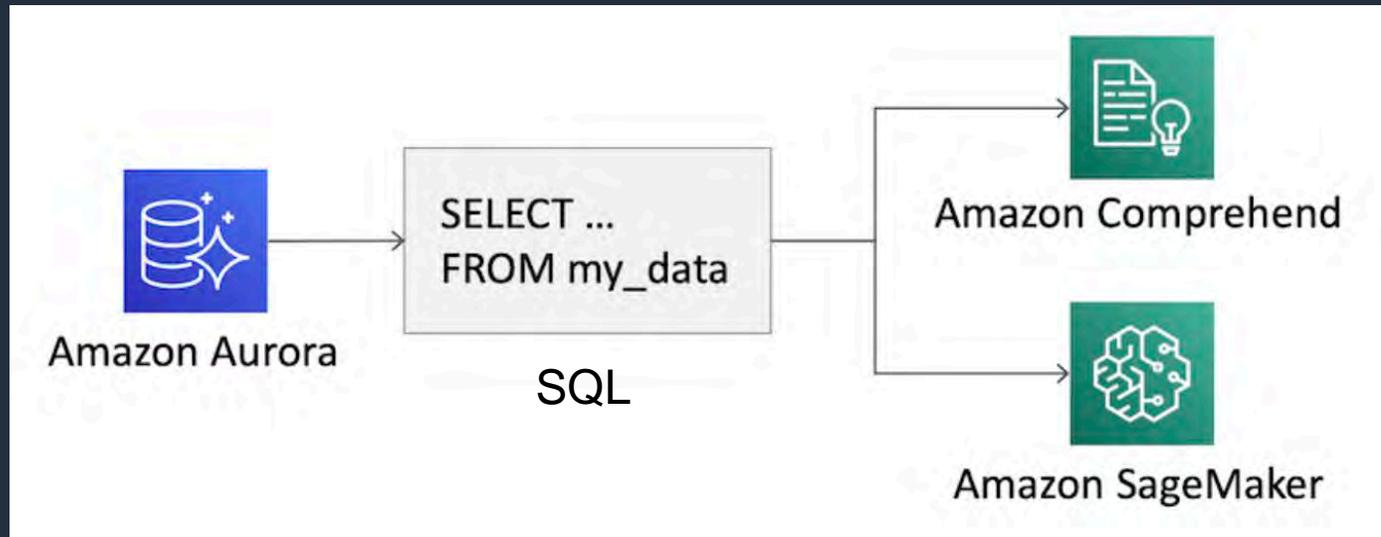
<https://aws.amazon.com/about-aws/whats-new/2019/11/amazon-athena-adds-support-for-invoking-machine-learning-models-in-sql-queries/>

<https://docs.aws.amazon.com/athena/latest/ug/querying-mlmodel.html>

Amazon Aurora Machine Learning



- SQL言語を使用して、データベースに機械学習の予測機能を追加
- Auroraデータベースから、Amazon SageMaker および Amazon Comprehend へ統合
- Amazon SageMaker モデルトレーニングのために、Amazon Aurora からデータを S3 にエクスポート

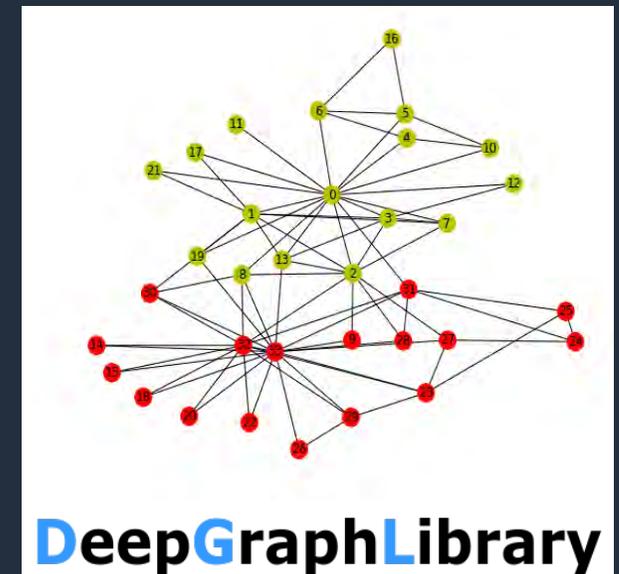


```
SQL
CREATE FUNCTION will_churn (
  state varchar(2048), acc_length bigint(20),
  area_code bigint(20), int_plan varchar(2048),
  vmail_plan varchar(2048), vmail_msg bigint(20),
  day_mins double, day_calls bigint(20),
  eve_mins double, eve_calls bigint(20),
  night_mins double, night_calls bigint(20),
  int_mins double, int_calls bigint(20),
  cust_service_calls bigint(20))
RETURNS varchar(2048) CHARSET latin1
alias aws_sagemaker_invoke_endpoint
endpoint name 'estimate_customer_churn_endpoint_version_123';
```

例) SQLでSageMakerエンドポイントにアクセスして、顧客離反予測

Deep Graph Library (DGL) が SageMaker で利用可能

- [DGL](#): GNNのすばやい開発・学習・評価を補助する Python のオープンソースライブラリ
- PyTorch や Apache MXNetとの連携が可能
- 下記の[ユースケース](#)の予測モデル生成が可能
 - ソーシャルネットワーク: 人同士の関係性を示すグラフ
 - 推薦システム: カスタマーと商品の関係性を示すグラフ
 - 化学構造解析: 化合物が原子やそれらの結合としてモデル化されているグラフ
 - サイバーセキュリティ: ソースとデスティネーション IP アドレスの関係性を示すグラフ



Amazon SageMaker Studio まとめ

- 統合開発環境 SageMaker Studio による開発効率の向上
- 開発・学習・推論に便利な機能をSDKベースで提供
- 開発：サーバレスノートブックで共同開発が容易に
- 学習：大量にできる学習モデルの管理と分析・可視化が
インターフェースを通してより簡単に行えるように
- 推論：定常的なモデル管理と分析が容易に
- AWS の他のサービスとの連携により豊富な機能が追加

ご清聴ありがとうございました

