



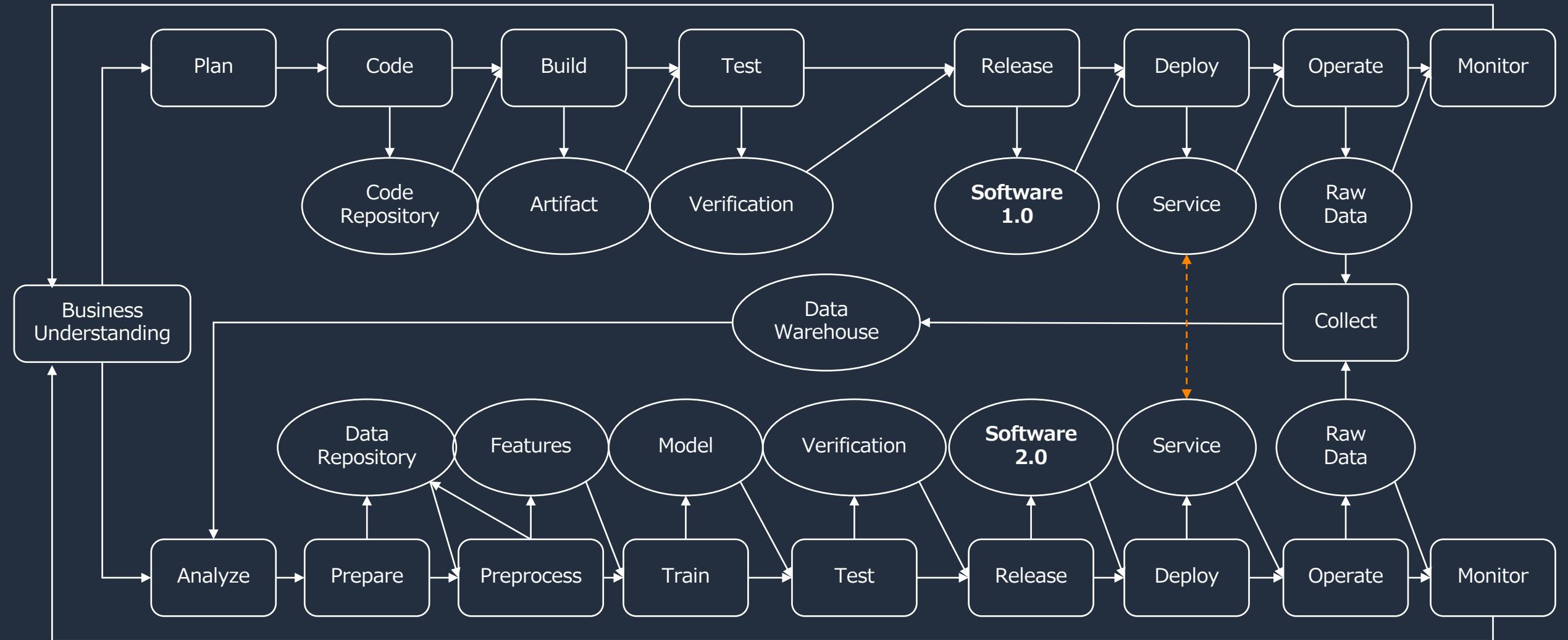
ML Enablement Series 【ML-Dark-02】

Amazon SageMaker による実験管理

機械学習ソリューションアーキテクト
伊藤 芳幸

DevOps & MLOps フロー

DevOps



MLOps

DevOps & MLOps を実現するロールマップ[®]

Architect

ソフトウェア開発に必要なソフトウェアアーキテクチャ全体を設計する。

Product Manager

実装すべきソフトウェア機能を定義する。
Plan

Business Analyst

解決すべきビジネス上の問題を定義する。

Analyze

Data Analyst

データの可視化と分析で問題を定量的に特定する。

DevOps Engineer

ソフトウェアの開発・運用プロセスを自動化する。

Software Engineer

ソフトウェアの開発を行う。

Code Repository

Build

Test

Release

Deploy

Operator

Operate

System Admin

Monitor

IT Auditor

システム

Data architect

データを管理する基盤を構築する。

Domain Expert

あるべき挙動をデータを用いて定義する。評価尺度を定義する。

Data Engineer

機械学習モデルに入力可能なデータと特徴を作成する。

Preprocess

Data Scientist

機械学習モデルを構築する

Train

Warehouse

ML Engineer

機械学習モデルを本番環境にデプロイ可能な形式に変換する。

Test

Service

Release

ML Operator

推論結果に基づき業務を行いつつ、推論結果にフィードバックを与える。

Deploy

Collect

Model risk Manager

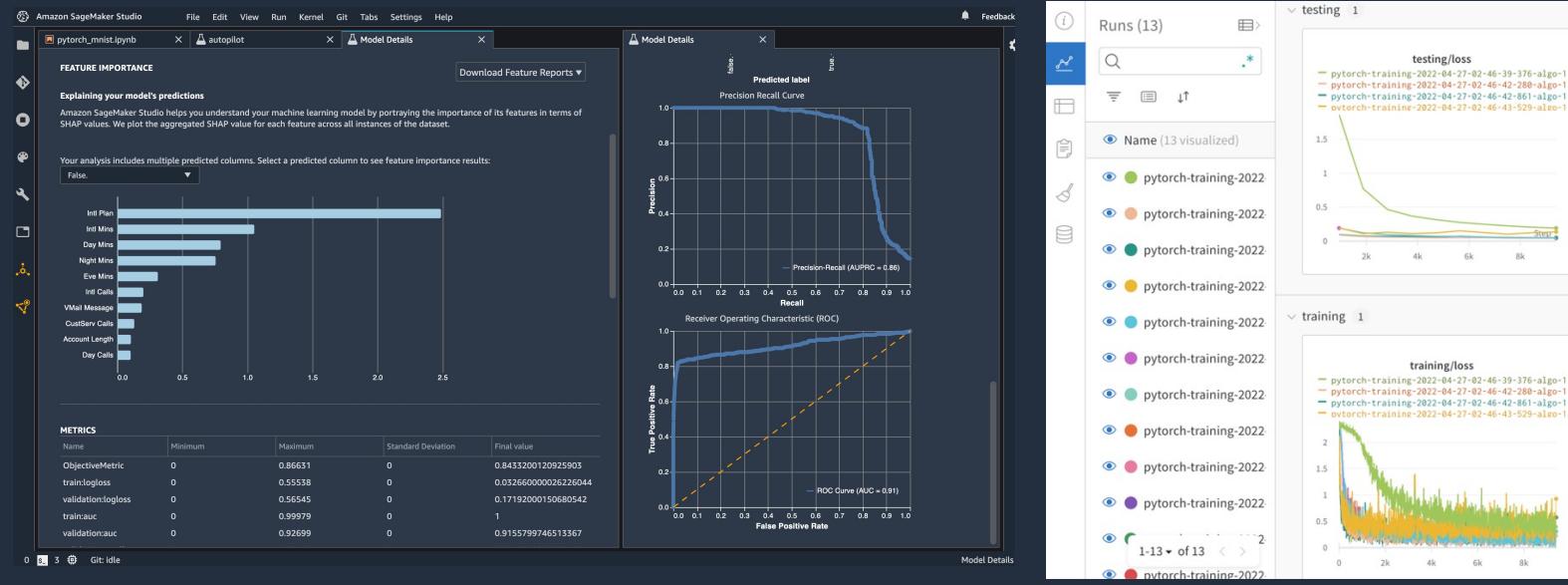
機械学習のサービスの挙動を監視する。

Monitor

MLOps Engineer 機械学習モデルの開発・運用プロセスを自動化する。

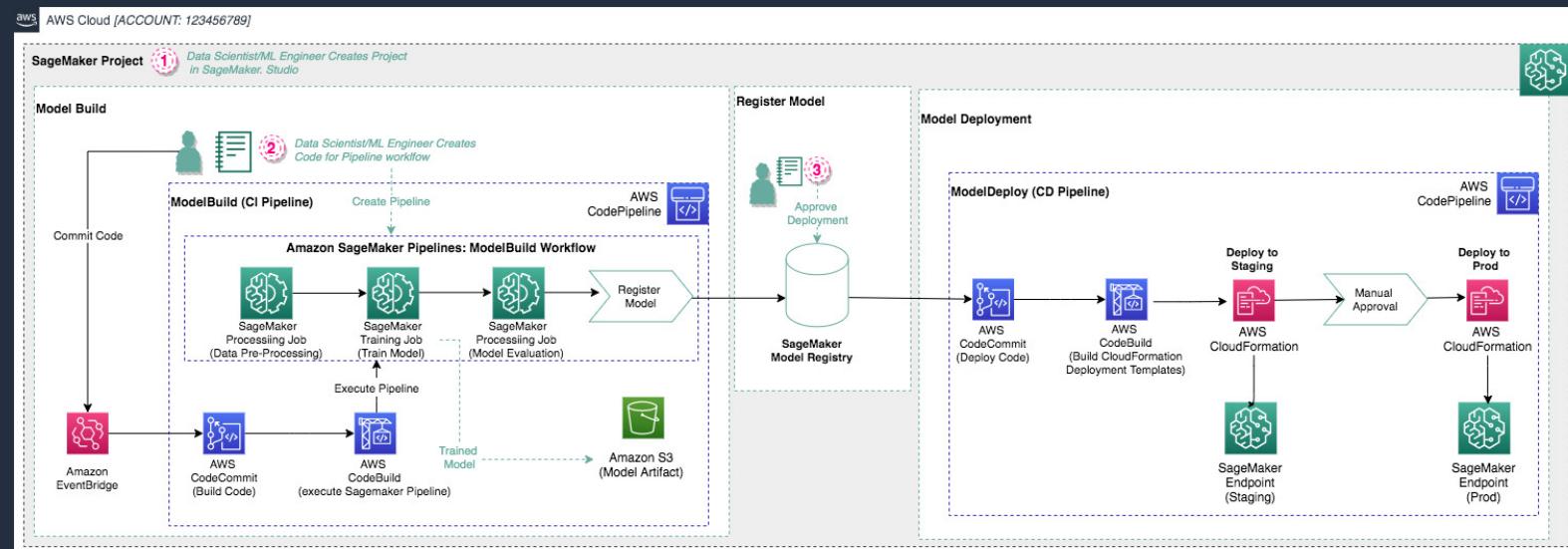
AI/ML Architect 機械学習に必要なアーキテクチャ全体を設計する。

この動画の対象者と得られること



データサイエンティスト向け

- 実験管理の必要性おさらい
- SageMakerでできる実験管理
- SageMakerと使い慣れた実験管理ツールとの連携



MLOpsエンジニア向け

- 実験管理の必要性おさらい
- パイプラインによるガバナンスの効いた実験管理の仕組み

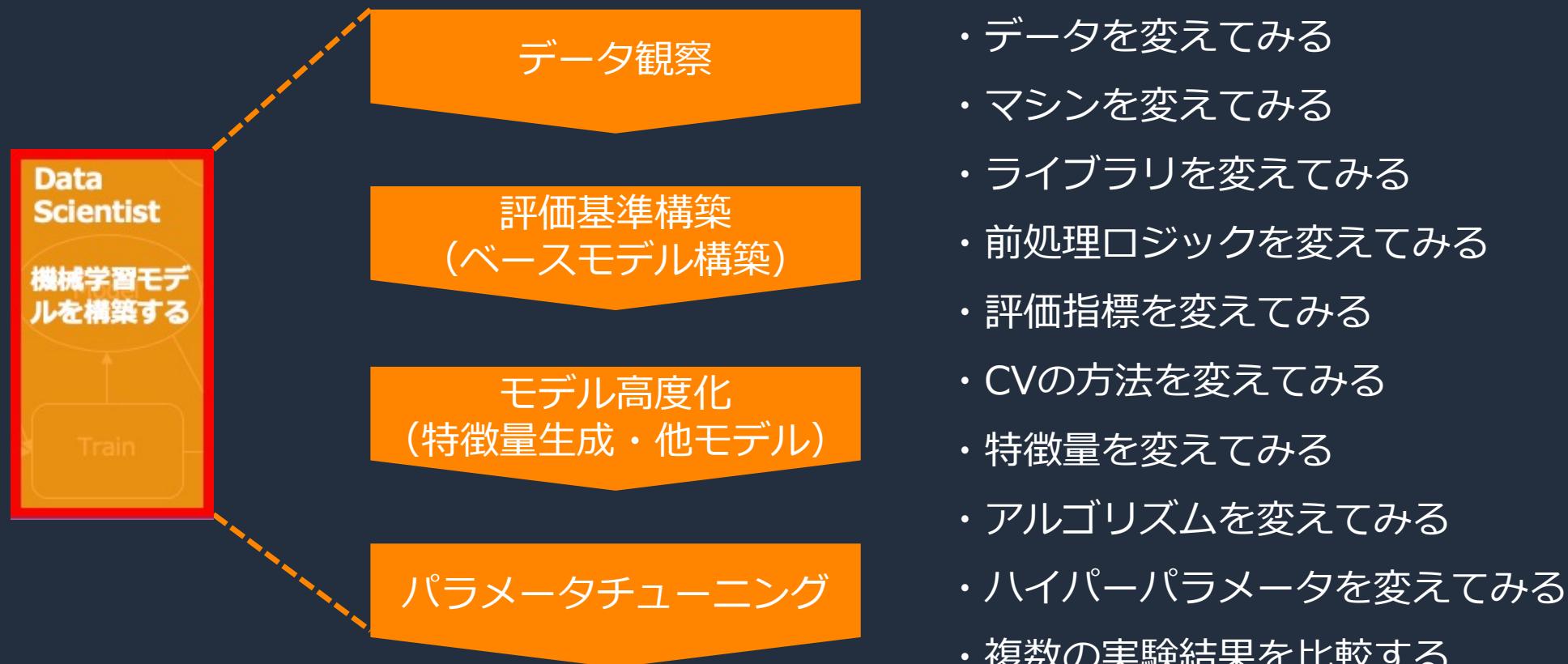
この動画で解説すること3つ

1. なぜ実験管理が必要なのか
2. 再現性のためのソリューション
3. 振り返りのためのソリューション

補足：ソリューション構築手順紹介

なぜ実験管理が必要なのか

モデル開発には試行錯誤が伴う



数百の実験をすることも！

ついやってしまう管理（管理ではない）

ローカルPCにノートブック形式で保存

exp091_train_lgbm_hpo1.ipynb
exp092_train_lgbm_hpo2.ipynb
exp093_train_lgbm_hpo3.ipynb
exp094_train_catb_feat1-Copy1.ipynb
exp094_train_catb_feat1.ipynb
exp095_train_catb_feat2.ipynb
exp096_train_catb_feat3.ipynb
exp097_train_mlp1.ipynb
exp100_train_mlp3.ipynb
exp101_eda_recall_check1.ipynb
exp102_eda_cv_check1.ipynb

✖ データ保存

- ・PCが壊れたり、担当データサイエンティストが退職して全てを失うケースも…

✖ 振り返り

- ・コード差分がわかりずらく、実験の把握に時間がかかる
 - ・セル出力を保存していない場合、ノートブックを再実行する必要がある

✖ 再現性

- ・実行順に記載されておらず、エラーが発生
 - ・異なる環境でエラーが発生

ローカルPCに表計算ソフトで記録

	A	B	C
1	notebook	Cvscore	TESTscore
2	exp035	0.66421478	0.12398777
3	exp037	0.4996275	0.40424265
4	exp040	0.58709581	0.36761254
5	exp041	0.88862392	0.96609412
6	exp042	0.73541102	0.36604913
7	exp043	0.76872504	0.46775784
8	exp044	0.81449647	0.7869291
9	exp045	0.74482335	0.01155465
10	exp048	0.97364383	0.54052217
11	exp049	0.54851112	0.79621083

✖ データ保存

- ・PCが壊れたり、担当データサイエンティストが退職して全てを失うケースも…

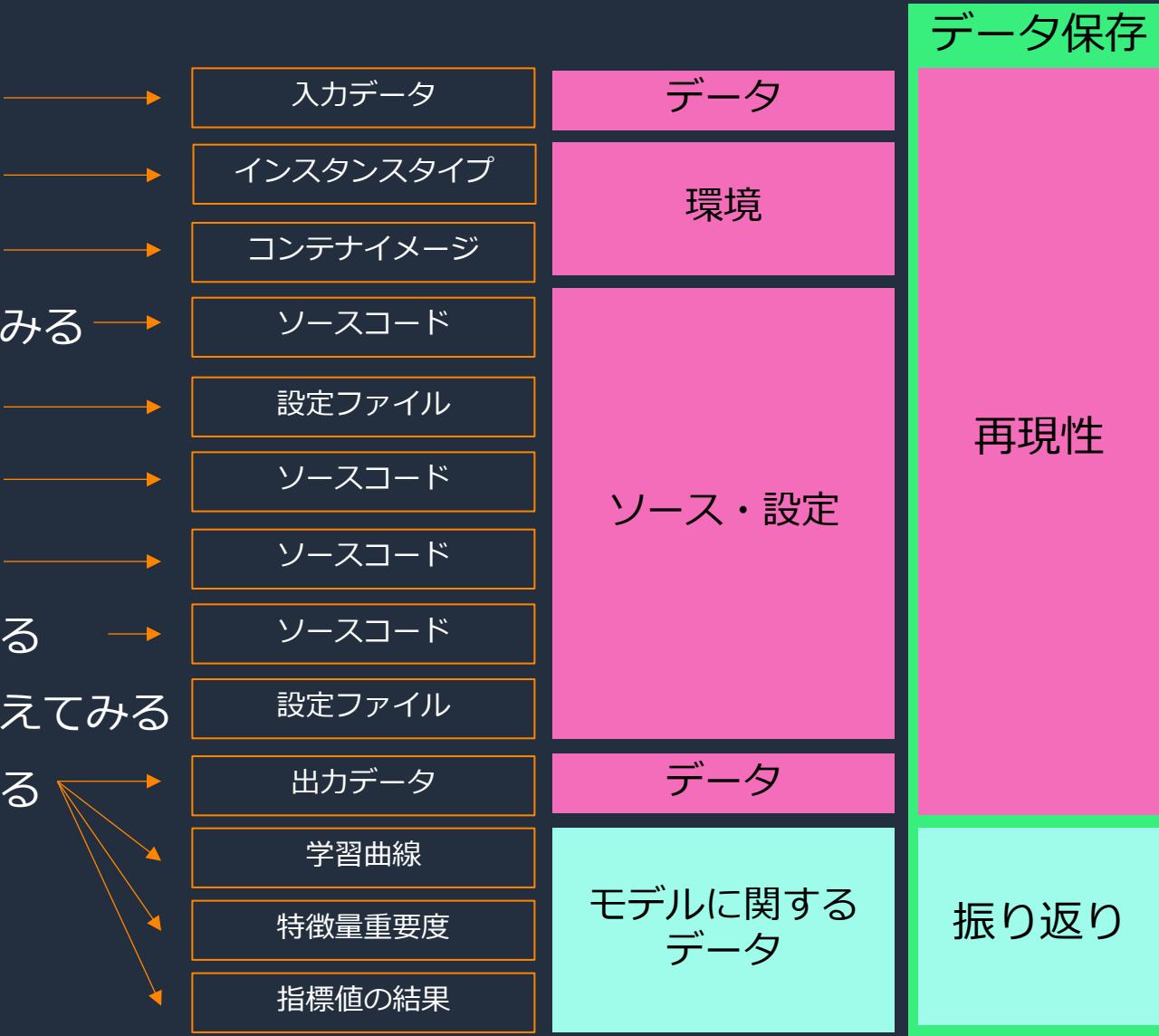
✖ 振り返り

- ・記録の抜け漏れがある
 - ・属人的（人の善意が頼り）

属人性が高く、組織として開発するには向かない仕組み

再現性と振り返りのために保存すべきデータ

- ・データを変えてみる
- ・マシンを変えてみる
- ・ライブラリを変えてみる
- ・前処理ロジックを変えてみる
- ・評価指標を変えてみる
- ・CVの方法を変えてみる
- ・特徴量を変えてみる
- ・アルゴリズムを変えてみる
- ・ハイパーパラメータを変えてみる
- ・複数の実験結果を比較する



ロールごとの、実験管理に求めること

データ保存

振り返り

再現性

管理は最低限にして、多く実験がしたい。
勝手に記録してくれて比較しやすい機能があると嬉しい。



データサイエンティスト

- ・実験記録が“消失しないこと
- ・自動で保存されること

比較しやすい実験管理ツールが
利用できること

他メンバや過去の自分の実験が
再現できること

開発されたモデルがスムーズにプロダクト
に組み込める仕組みを作りたい。
モデル構築にガバナンスを効かせたい。



MLOpsエンジニア

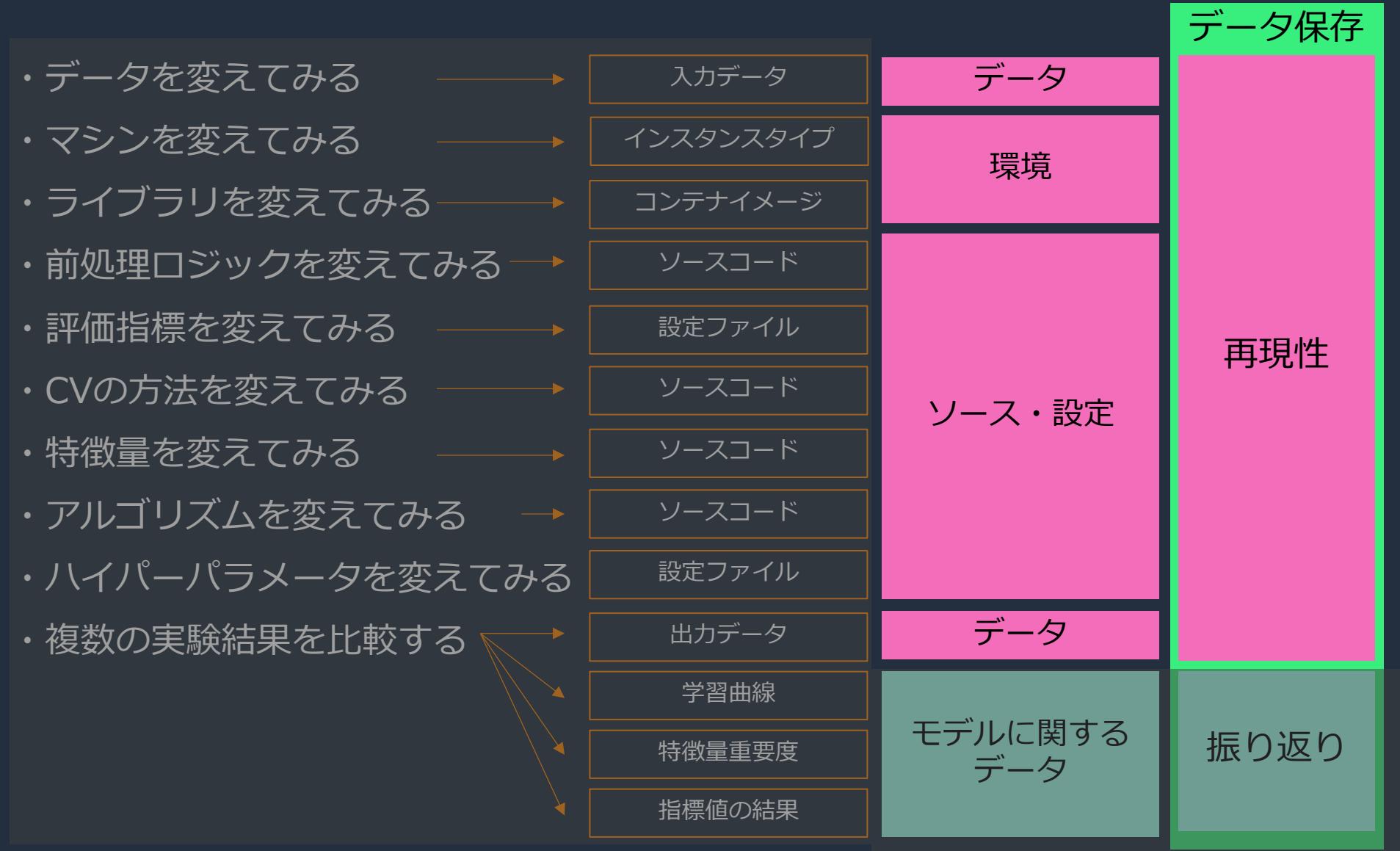
- ・実験記録が“消失しないこと
- ・自動で保存されること

データサイエンティストが効率
的に振り返りができること

データサイエンティストがいな
くても実験を再現できること

再現性のためのソリューション

再現性と振り返りのために保存すべきデータ



とはいえ、ノートブックの柔軟性は捨てがたい…



クラウドで実施しましょう

統合開発環境（IDE）：Amazon SageMaker Studio

Web ブラウザから利用可能なクラウドの Jupyter notebook 環境

The screenshot shows the Amazon SageMaker Studio interface. On the left, a Jupyter notebook cell contains Python code for computing anomaly scores from a taxi dataset. The code includes importing libraries, defining a function `compute_anomaly_scores`, and plotting results. A note at the bottom explains that the notebook uses an eyeball-norm method to identify anomalies. On the right, a "Trial Component Chart" interface displays a scatter plot of test-metric values over time. The chart has a legend indicating different AWS instances (Apple111, etc.) and a dropdown menu for Y-axis selection. Below the chart is a "Trial Component List" showing completed training jobs for the Fruits111 experiment.

```
random_forest.ipynb
Computing Anomaly Scores
Now, let's compute and plot the anomaly scores from the entire taxi dataset.

[ ]: results = rcf_inference.predict(taxi_data_numpy)
scores = [datum['score'] for datum in results['scores']]

# add scores to taxi data frame and print first few values
taxi_data['score'] = pd.Series(scores, index=taxi_data.index)
taxi_data.head()

[ ]: fig, ax1 = plt.subplots()
ax2 = ax1.twinx()

# *Try this out* - change 'start' and 'end' to zoom in on the
# anomaly found earlier in this notebook
#
start, end = 0, len(taxi_data)
#start, end = 5500, 6500
taxi_data_subset = taxi_data[start:end]

ax1.plot(taxi_data_subset['value'], color='C0', alpha=0.8)
ax2.plot(taxi_data_subset['score'], color='C1')

ax1.grid(which='major', axis='both')
ax1.set_ylabel('Taxi Rideship', color='C0')
ax2.set_ylabel('Anomaly Score', color='C1')

ax1.tick_params('y', colors='C0')
ax2.tick_params('y', colors='C1')

ax1.set_ylim(0, 40000)
ax2.set_ylim(min(scores), 1.4*max(scores))
fig.set_figwidth(10)

Note that the anomaly score spikes where our eyeball-norm method suggests there is an
anomalous data point as well as in some places where our eyeballs are not as accurate.

Below we print and plot any data points with scores greater than 3 standard deviations
(approx 99.9th percentile) from the mean score.

[ ]: score_mean = taxi_data['score'].mean()
score_std = taxi_data['score'].std()
score_cutoff = score_mean + 3*score_std

anomalies = taxi_data_subset[taxi_data_subset['score'] > score_cutoff]
anomalies

The following is a list of known anomalous events which occurred in New York City within this
timeframe:
```

データ

Jupyter上のファイルはユーザー毎のEFSに
保管されている

環境

記録されない。実行インスタンスタイプやコ
ンテナはノートブックなどに記載しておく必
要あり

ソース・設定

ノートブックとして記録されているのみ

振り返り

ノートブックを開いて、セル出力結果を都度
確認する

SageMaker-run-notebook

使い方は動画後半で

Papermill で Jupyter ノートブックのままバッチ実行が可能

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for 'Runs' (selected), 'Schedules', 'CURRENT NOTEBOOK' (Model Profiler.ipynb), 'NOTEBOOK EXECUTION' (Parameters: experiment_base_name, version, s3_test_data; Image: 670355705955.dkr.ecr, Role: arn:aws:iam::670355705955, Instance: ml.m4.xlarge), and 'SCHEDULE RULE' (Rule Name, Schedule, Event Pattern). The main area shows a Jupyter notebook cell with Python code to plot an ROC curve:

```
plt.title('ROC curve')
plt.legend(loc='best')
plt.show()
```

The resulting plot is an ROC curve titled "ROC curve" comparing "XGBoost" and "Linear Learner". The x-axis is "False positive rate" and the y-axis is "True positive rate", both ranging from 0.0 to 1.0. A dashed diagonal line represents a random classifier, while the model curves are above it, indicating better performance.

Below the plot, another cell shows some cleanup code:

```
[344]: # for e in endpoints:
#     e.delete_endpoint()
```

At the bottom, the status bar indicates "Mode: Edit" and "Ln 1, Col 1 Model Profiler.ipynb".

データ

事前にS3に保存する。ノートブックではS3のファイルをロードする。

環境

実行インスタンスタイプとコンテナURLが記録される

ソース・設定

実行前ノートブックと、実行後ノートブックがS3に保存される

振り返り

実行後ノートブックを確認

AWSブログ：<https://aws.amazon.com/jp/blogs/news/scheduling-jupyter-notebooks-on-sagemaker-ephemeral-instances/>
GitHub：<https://github.com/aws-samples/sagemaker-run-notebook>



SageMaker ジョブをノートブックから実行

再現に必要なデータを自動で記録

```
# トレーニングジョブの実行
from sagemaker.tensorflow import TensorFlow
estimator = TensorFlow(
    entry_point='./src/1-2-1/calc.py',
    py_version='py38',
    framework_version='2.7.1',
    instance_count=1,
    instance_type='ml.m5.xlarge',
    role=sagemaker.get_execution_role()
)
estimator.fit(input_s3_uri)
```

Algorithm	Hyperparameters
Algorithm ARN	
-	
Training image	コンテナイメージ
763104351884.dkr.ecr.ap-northeast-1.amazonaws.com/tensorflow-training:2.7.1-cpu-py38	
Input mode	
File	
Instance type	インスタンスタイプ
ml.m5.xlarge	
Instance count	
1	
その他	入力データ 出力データ
なども記録	

データ

- ・S3に事前に配置する
- ・実行ではS3の場所を指定

環境

コンテナイメージの場所、インスタンスタイプが記録される

ソース・設定

- ・利用したソースコードをS3に保存
- ・入力パラメータが記録される

振り返り

- ・マネジメントコンソールで記録を確認
- ・metric_definitionsで指標値を記録
(SageMaker 学習ジョブ)



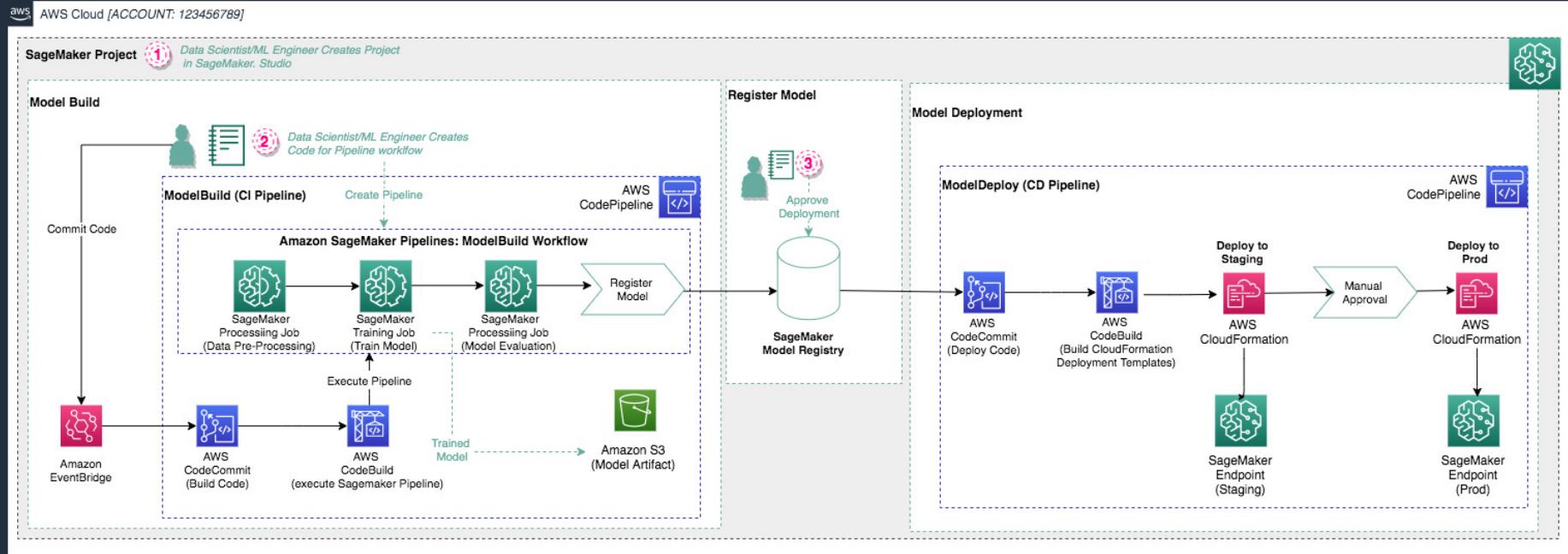
https://github.com/aws-samples/aws-ml-jp/blob/main/sagemaker/sagemaker-traning/tutorial/1_hello_sagemaker_training.ipynb

© 2022, Amazon Web Services, Inc. or its affiliates.

SageMaker Pipelines

使い方は動画後半で

SageMaker Studio から数クリックで構築できるパイプライン



- ・ MLOpsエンジニアはパイプラインを構築・管理する
- ・ 成果物をリポジトリに提出するルールとすることで、ガバナンスを確立
- ・ データサイエンティストはコーディング、リポジトリへのpushを担当し、モデル構築に注力する

データ

SageMakerジョブと同等

環境

SageMakerジョブと同等

ソース・設定

リポジトリに保存される

振り返り

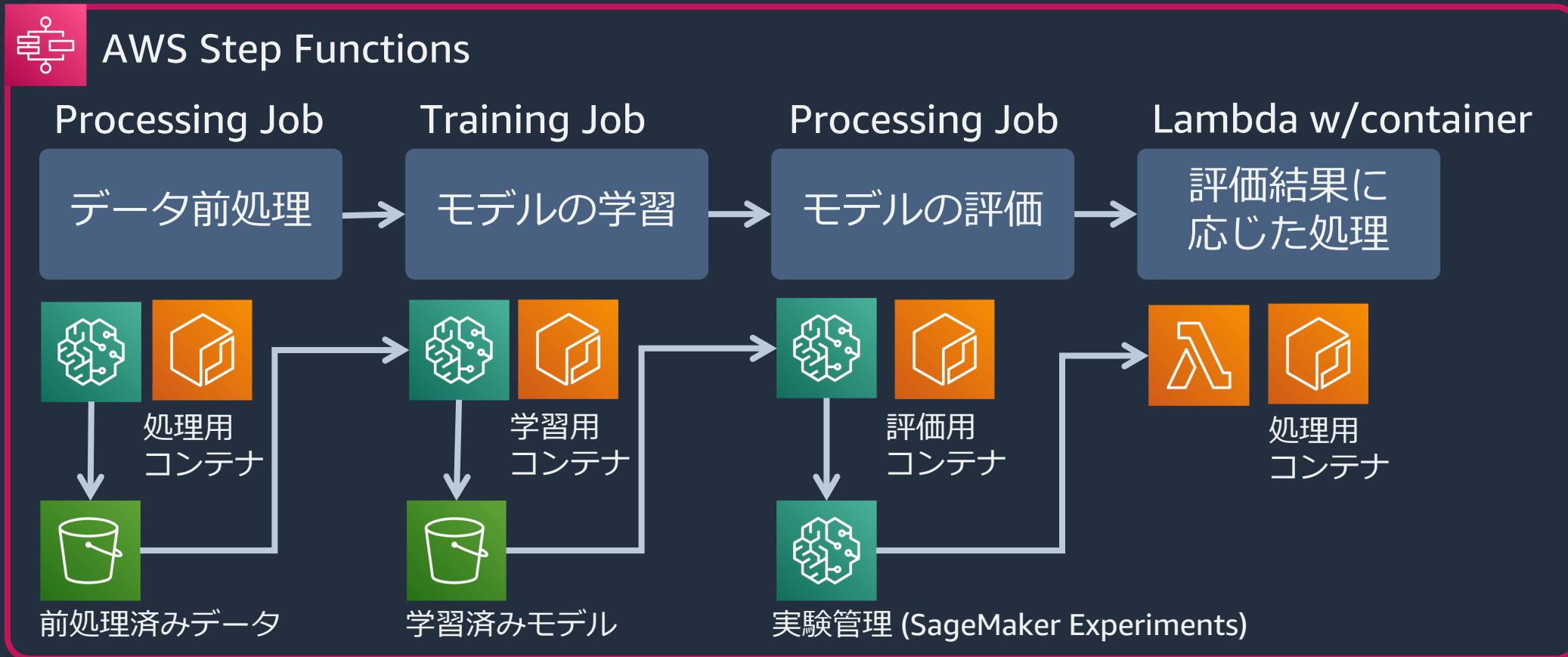
SageMakerジョブと同等



<https://aws.amazon.com/blogs/machine-learning/building-automating-managing-and-scaling-ml-workflows-using-amazon-sagemaker-pipelines/>

AWS StepFunctions パイプライン

共通部分をパイプラインにして自動化し、
使用するデータや学習スクリプトを変えながらより良いモデルを効率的に探索



https://aws.amazon.com/jp/builders-flash/202111/nyantech-ml-ops/?awsf.filter-name=*all

再現性のためのソリューションまとめ

- ・自社のユースケースや組織、どこまで再現性を要求するかをもとにソリューションを選択する
- ・SageMakerジョブで前処理や学習を行なっておけば、高い再現性を確保できる

ソリューション	再現性	データサイエンティストの学習コスト	MLOpsエンジニアが実施すること
ローカルノートブック	DS退職やPC壊れたら終了	特になし	特になし
SageMaker Studio	ノートブックをEFSに保存	特になし	SageMaker Studio ドメイン構築
SageMaker-run-notebook	ノートブック・実行環境をS3に保存（実行前・後）	S3でのデータの扱い方	Run-notebookコンテナ構築
SageMakerジョブ (ノートブックから)	データ・環境・ソースコードを保存（S3）	(上記に加え) SageMaker SDKの使い方	特になし
パイプライン (SageMaker Pipelines / AWS StepFunctions など)	ソースコードをリポジトリ（CodeCommit / GitHubなど）に保存	(上記に加え) ・リポジトリの使い方 ・パイプラインの理解	パイプライン構築

振り返りのためのソリューション

実験の振り返り、比較に手間がかかってしまう

ローカルPCにノートブック形式で保存

[exp091_train_lgbm_hpo1.ipynb](#)
[exp092_train_lgbm_hpo2.ipynb](#)
[exp093_train_lgbm_hpo3.ipynb](#)
[exp094_train_catb_feat1-Copy1.ipynb](#)
[exp094_train_catb_feat1.ipynb](#)
[exp095_train_catb_feat2.ipynb](#)
[exp096_train_catb_feat3.ipynb](#)
[exp097_train_mlp1.ipynb](#)
[exp100_train_mlp3.ipynb](#)
[exp101_eda_recall_check1.ipynb](#)
[exp102_eda_cv_check1.ipynb](#)

✖ データ保存

- ・PCが壊れたり、担当データサイエンティストが退職して全てを失うケースも…

✖ 振り返り

- ・コード差分がわかりずらく、実験の把握に時間がかかる
- ・セル出力を保存していない場合、ノートブックを再実行する必要がある

✖ 再現性

- ・実行順に記載されておらず、エラーが発生
- ・異なる環境でエラーが発生

ローカルPCに表計算ソフトで記録

	A	B	C
1	notebook	Cvscore	TESTscore
2	exp035	0.66421478	0.12398777
3	exp037	0.4996275	0.40424265
4	exp040	0.58709581	0.36761254
5	exp041	0.88862392	0.96609412
6	exp042	0.73541102	0.36604913
7	exp043	0.76872504	0.46775784
8	exp044	0.81449647	0.7869291
9	exp045	0.74482335	0.01155465
10	exp048	0.97364383	0.54052217
11	exp049	0.54851112	0.79621083

✖ データ保存

- ・PCが壊れたり、担当データサイエンティストが退職して全てを失うケースも…

✖ 振り返り

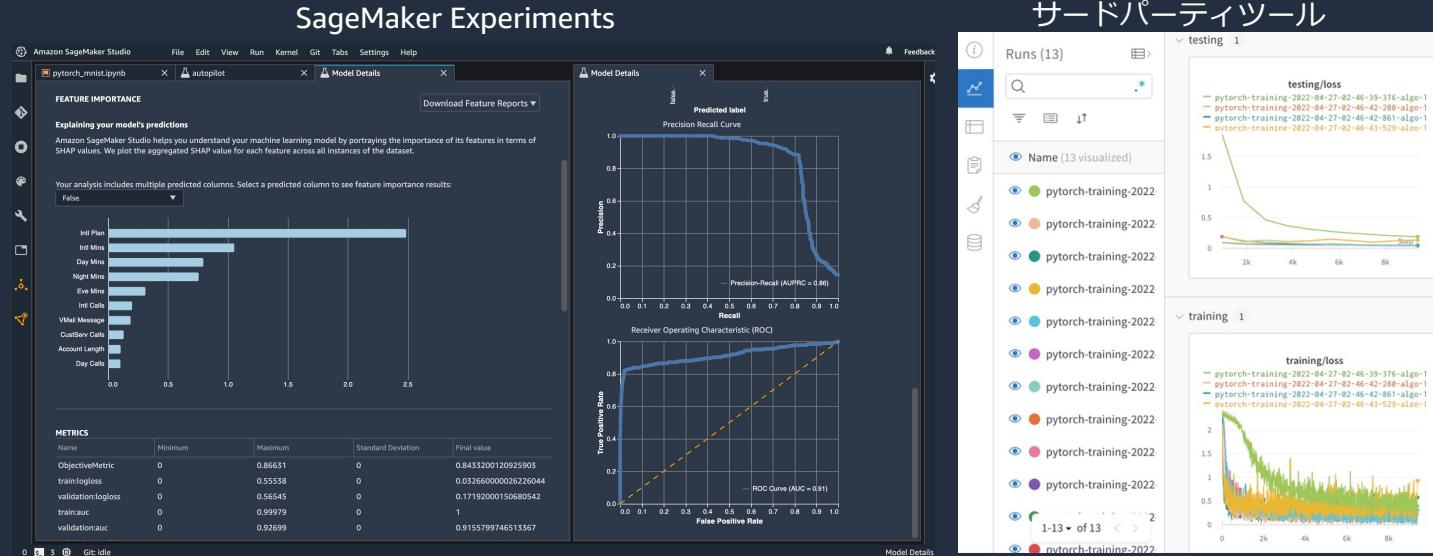
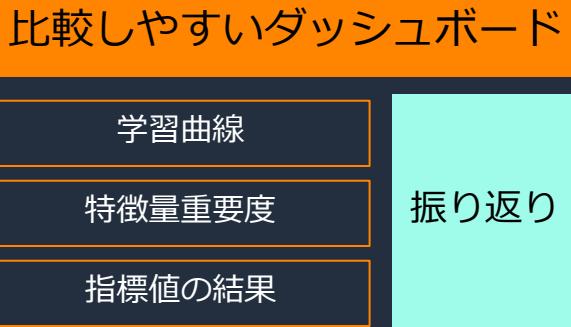
- ・記録の抜け漏れがある
- ・属人的（人の善意が頼り）

ひとつずつ開いて確認

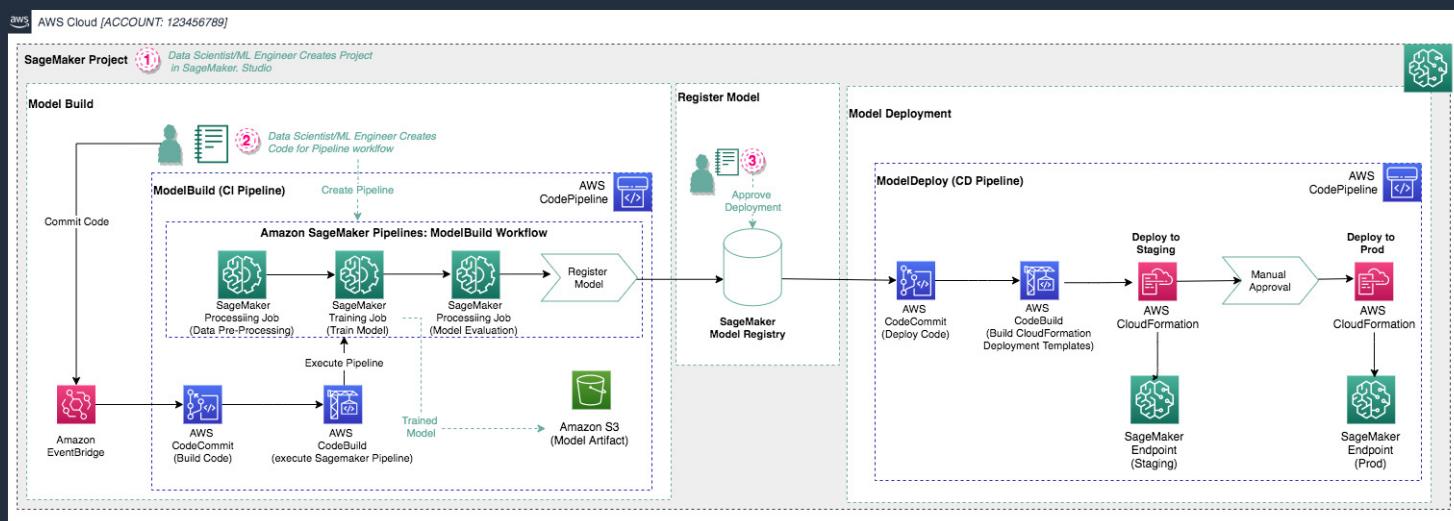
記録に手間がかかる

ダッシュボードで振り返り、パイプラインで再現性担保

ひとつずつ開いて確認

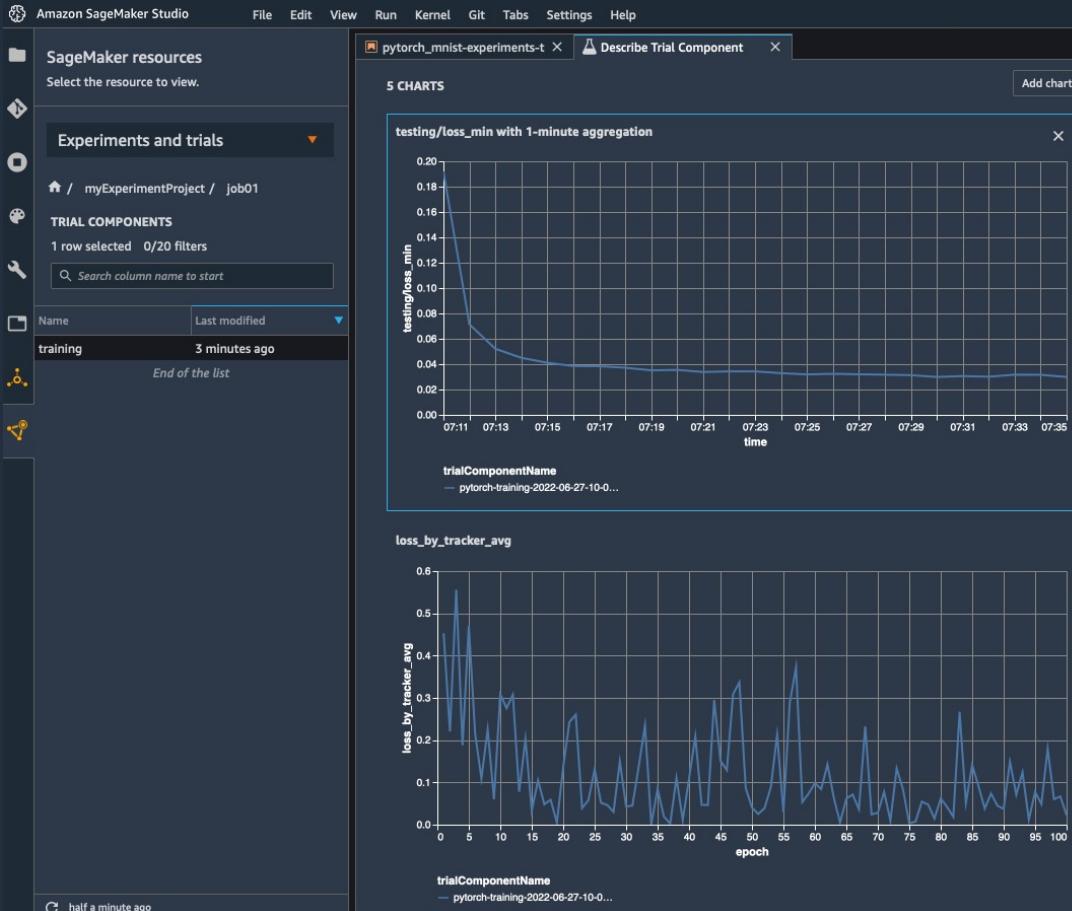


記録に手間がかかる



SageMaker Experiments で比較する

例 : SageMaker Experiments



ジョブ発行前に記載

```
from smexperiments import experiment, tracker

my_experiment = experiment.Experiment.create(experiment_name='myExperimentProject')
my_trial = my_experiment.create_trial(trial_name='job01')

with tracker.Tracker.create(display_name='training') as my_tracker:
    my_tracker.log_input(name="input-dataset-dir", media_type="s3/uri", value=inputs)

estimator.fit({'training': inputs},
              experiment_config={
                  "TrialName": my_trial.trial_name,
                  "TrialComponentDisplayName": my_tracker.trial_component.display_name,
              })
```

ソースコードに記載

```
from smexperiments import tracker
# load tracker from already existing trial component
my_tracker = tracker.Tracker.load()
for epoch in range(1, args.epochs + 1):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader, 1):

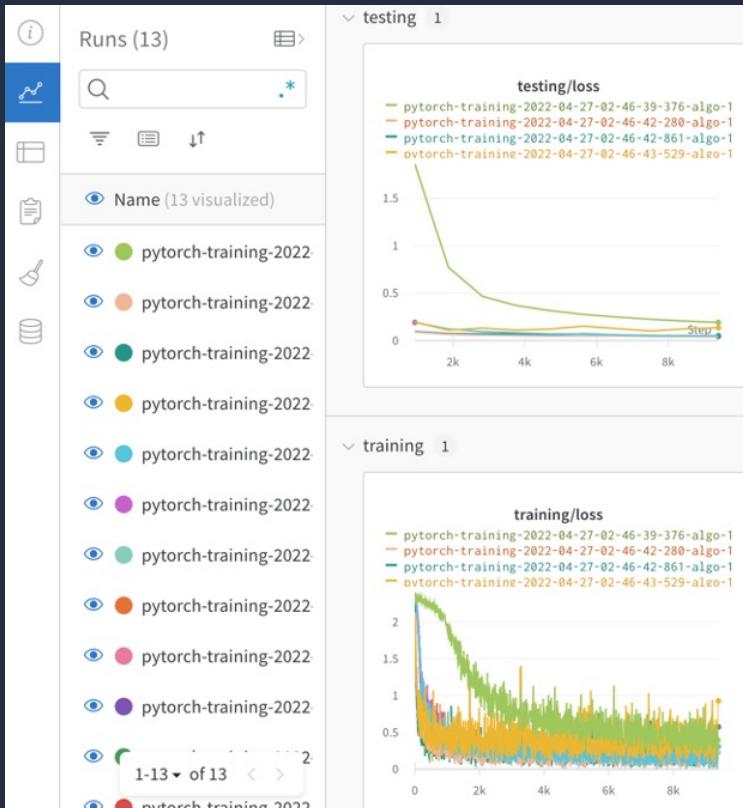
        # epochの終わりにlossを記録
        my_tracker.log_metric(metric_name='loss_by_tracker', value=loss.item(), iteration_number=epoch)
```

<https://github.com/aws/amazon-sagemaker-examples/blob/main/sagemaker-experiments/mnist-handwritten-digits-classification-experiment/mnist-handwritten-digits-classification-experiment.ipynb>



サードパーティツールで比較する

例 : Weights & Biases



ジョブ発行前に記載

```
import wandb
wandb.login()
settings = wandb.setup().settings
current_api_key =
wandb.wandb_lib.apikey.api_key(settings=settings)
```

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(entry_point='mnist.py',
                    source_dir="src",
                    role=role,
                    py_version='py3',
                    framework_version='1.8.0',
                    instance_count=1,
                    instance_type='ml.c5.2xlarge',
                    hyperparameters={
                        'epochs': 1,
                        'backend': 'gloo'
                    },
                    environment={"WANDB_API_KEY": current_api_key})
```

ソースコードに記載

```
import wandb

wandb.init(project="sm-pytorch-mnist-studio",
config=vars(args))

wandb.watch(model)

wandb.log({"training/loss": loss.item()})
```

<https://github.com/wandb/examples/tree/master/examples/pytorch/pytorch-cifar10-sagemaker>



SageMakerは多くのサードパーティツールと連携可能

Weights & Biases

<https://docs.wandb.ai/guides/integrations/other/sagemaker>

MLflow Tracking

<https://aws.amazon.com/jp/blogs/news/machine-learning-managing-your-machine-learning-lifecycle-with-mlflow-and-amazon-sagemaker/>

Neptune.ai

<https://docs.neptune.ai/integrations-and-supported-tools/ide-and-notebooks/amazon-sagemaker>

Comet.ml

<https://www.comet.ml/site/building-reliable-machine-learning-pipelines-with-aws-sagemaker-and-comet-ml/>



まとめ

- ・組織で機械学習モデルの開発をするには、実験の再現性を確保することが重要。
- ・SageMaker ジョブを利用すれば、実験の再現に必要な情報は自動で記録することが可能。パイプラインを構築することで、ソースコードのリポジトリ管理や再現性確保のための情報の自動記録など、さらにガバナンスの効いた仕組みを構築できる。
- ・SageMaker はサードパーティの実験管理ツールの多くと連携でき、データサイエンティストが使いやすいツールを利用して振り返りを行うこともできる。

補足：ソリューション構築

Run-notebook
SageMaker Pipelines



SageMaker-run-notebook

使い方は動画後半で

Papermill で Jupyter ノートブックのままバッチ実行が可能

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with sections for 'VIEW' (Runs, Schedules), 'CURRENT NOTEBOOK' (Model Profiler.ipynb), 'NOTEBOOK EXECUTION' (Parameters: experiment_base_name, version, s3_test_data; Image: 670355705955.dkr.ecr, Role: arn:aws:iam::670355705955, Instance: ml.m4.xlarge), and 'SCHEDULE RULE' (Rule Name, Schedule, Event Pattern). The main area shows a Jupyter notebook cell with Python code to plot an ROC curve:

```
plt.title('ROC curve')
plt.legend(loc='best')
plt.show()
```

The resulting plot is an ROC curve titled "ROC curve" comparing "XGBoost" and "Linear Learner". The x-axis is "False positive rate" and the y-axis is "True positive rate", both ranging from 0.0 to 1.0. A dashed diagonal line represents a random classifier, while the model curves are above it, indicating better performance.

Below the plot, another cell shows some cleanup code:

```
[344]: # for e in endpoints:
#     e.delete_endpoint()
```

At the bottom, the status bar indicates "Mode: Edit" and "Ln 1, Col 1 Model Profiler.ipynb".

データ

事前にS3に保存する。ノートブックではS3のファイルをロードする。

環境

実行インスタンスタイプとコンテナURLが記録される

ソース・設定

実行前ノートブックと、実行後ノートブックがS3に保存される

振り返り

実行後ノートブックを確認

AWSブログ：<https://aws.amazon.com/jp/blogs/news/scheduling-jupyter-notebooks-on-sagemaker-ephemeral-instances/>
GitHub：<https://github.com/aws-samples/sagemaker-run-notebook>



ノートブックのまま実行する環境を構築する

SageMaker Studio システムコンソールなどで以下のコマンドを実行

コマンド1 `pip install https://github.com/aws-samples/sagemaker-run-notebook/releases/download/v0.20.0/sagemaker_run_notebook-0.20.0.tar.gz`

`run-notebook` コマンドのインストール

コマンド2 `run-notebook create-infrastructure`

AWS CloudFormationで、AWS Lambda や IAM role が作成される

コマンド3 `run-notebook create-container`

AWS CodeBuild が実行され、ノートブックを SageMaker Processing で利用する コンテナ (Papermill 入り) が構築される

コマンド4 `bash install-run-notebook.sh`

`install-run-notebook.sh` の内容（自分で作成する）

`version=0.18.0`

```
pip install https://github.com/aws-samples/sagemaker-run-notebook/releases/download/v${version}/sagemaker_run_notebook-${version}.tar.gz
```

```
jlpm config set cache-folder /tmp/yarncache
```

```
jupyter lab build --debug --minimize=False
```

```
nohup supervisorctl -c /etc/supervisor/conf.d/supervisord.conf restart jupyterlabserver
```

事前に付与しておく権限

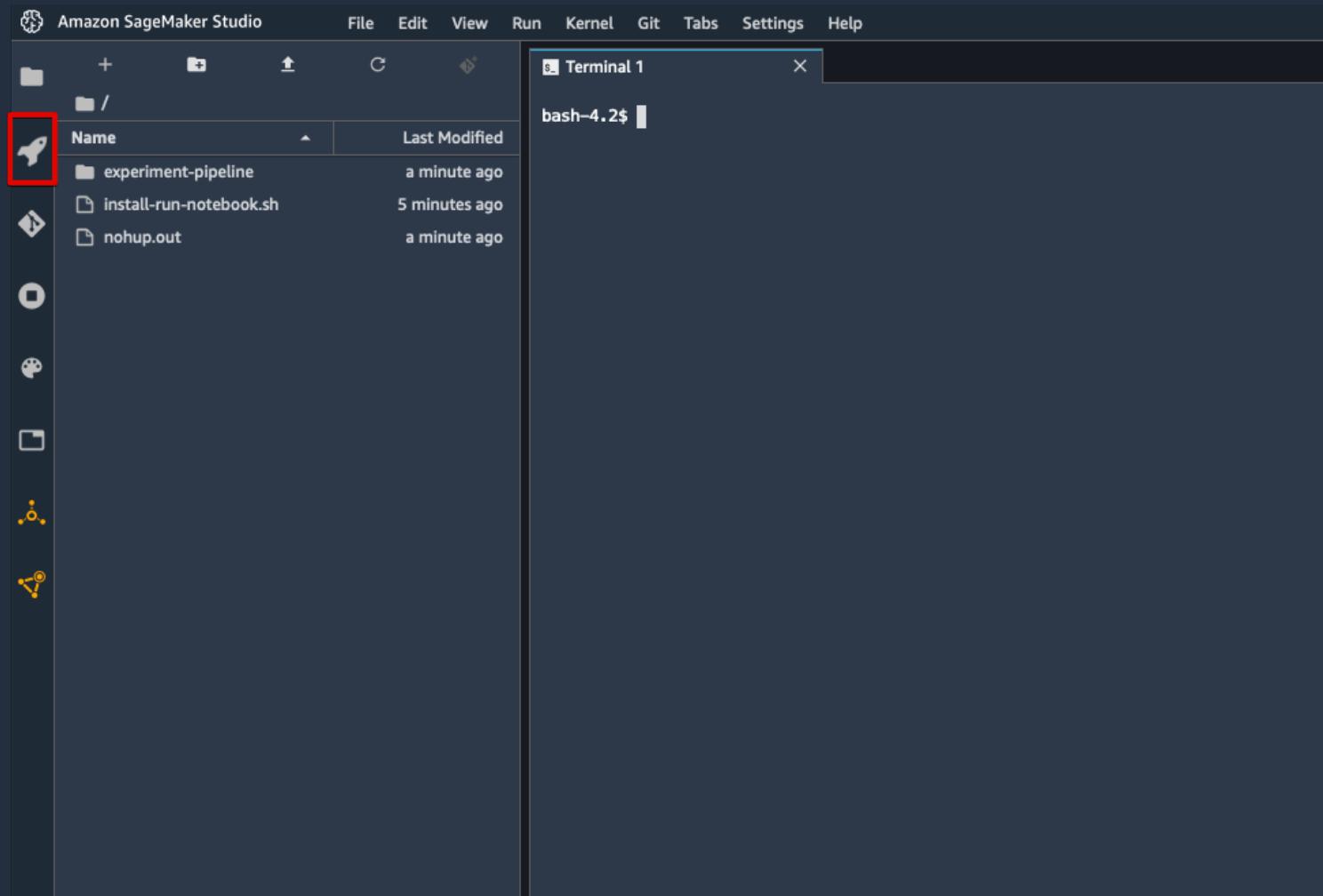
Policy name
+ IAMFullAccess
+ AmazonS3FullAccess
+ AWSCodeBuildAdminAccess
+ AmazonSageMakerFullAccess
+ AWSCloudFormationFullAccess
+ AWSLambda_FullAccess



<https://github.com/aws-samples/sagemaker-run-notebook/blob/master/QuickStart.md>

© 2022, Amazon Web Services, Inc. or its affiliates.

ブラウザ更新で、拡張機能が表示される



ノートブックを選択した状態で、[Run Now]で実行

The screenshot shows the Amazon SageMaker Studio interface. On the left, the 'CURRENT NOTEBOOK' section displays 'exp002-demo-xgboost.ipynb'. Below it, the 'NOTEBOOK EXECUTION' section shows 'Parameters:' with a note: 'No parameters defined. Press "+" to add parameters.' It also lists 'Image', 'Role', and 'Instance' settings, which are highlighted with a red box. The 'Run Now' button is also highlighted with a red box. A green message box at the bottom states 'Started notebook run "papermill-exp002-demo-xgboost-2022-06-19-15-22-05"'.

VIEW

File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 X install-run-notebook X exp001-demo.ipynb X exp002-demo-xgb X Notebook Runs X [Read-only] exp002-d...

Code git 2 vCPU + 4 GiB Cluster Python 3 (Data Science) Share

Runs Schedules

CURRENT NOTEBOOK

exp002-demo-xgboost.ipynb

NOTEBOOK EXECUTION

Parameters:
No parameters defined. Press "+" to add parameters.

Image: 871040346072.dkr.ecr.ap-northeast-1.amazonaws.com/ml-xgboost-0.1.0:latest
Role: arn:aws:iam::871040346072:role/SageMaker-Processing
Instance: ml.m5.2xlarge

SCHEDULE RULE

Rule Name:
Schedule:
Event Pattern:

Run Now Create Schedule

Started notebook run "papermill-exp002-demo-xgboost-2022-06-19-15-22-05"

1 \$ 3 Git: idle Python 3 (Data Science) | Idle Kernel: Idle | Instance MEM Mode: Command ✘ Ln 1, Col 1 exp002-demo-xgboost.ipynb

Image
Role
Instance
を設定して、
[Run Now] を押下すると、
SageMaker Processingジョブが
発行される

[41] train-logloss:0.00920 eval-logloss:0.15507
[42] train-logloss:0.00910 eval-logloss:0.15460
[43] train-logloss:0.00900 eval-logloss:0.15430
[44] train-logloss:0.00891 eval-logloss:0.15383
[45] train-logloss:0.00882 eval-logloss:0.15476
[46] train-logloss:0.00874 eval-logloss:0.15427
[47] train-logloss:0.00865 eval-logloss:0.15402
[48] train-logloss:0.00858 eval-logloss:0.15340
[49] train-logloss:0.00850 eval-logloss:0.15298

[6]: # 予測：検証用データが各クラスに分類される確率を計算する
pred_proba = model.predict(dtest)
しきい値 0.5 で 0, 1 に丸める
pred = np.where(pred_proba > 0.5, 1, 0)
精度 (Accuracy) を検証する
acc = accuracy_score(test_y, pred)
print('Accuracy:', acc)

Accuracy: 0.9385964912280702

実行ジョブは一覧で確認することができる

The screenshot shows the Amazon SageMaker Studio interface with the 'Notebook Runs' tab selected. On the left, there's a sidebar with icons for file operations like Create, Open, Save, and Delete, along with a current notebook selection area. The main content area displays a table titled 'Notebook Execution History' with the following data:

Rule Notebook	Parameters Status	Start	Elapsed	View Details	View Output
exp002-demo-xgboost.ipynb	Completed	6/20/2022, 12:26:22 AM	0:00:17.296000	View Details	View Output
exp001-demo.ipynb	Completed	6/20/2022, 12:08:07 AM	0:00:17.176000	View Details	View Output
exp001-demo.ipynb	Failed			View Details	
exp001-demo.ipynb	Completed	6/19/2022, 11:27:34 PM	0:00:17.141000	View Details	View Output
exp001-demo.ipynb	Completed	6/19/2022, 11:18:23 PM	0:00:17.208000	View Details	View Output
exp001-demo.ipynb	Completed	6/19/2022, 10:43:10 PM	0:00:17.080000	View Details	View Output
exp001-demo.ipynb	Completed	6/19/2022, 10:25:51 PM	0:00:17.246000	View Details	View Output
exp001-demo.ipynb	Completed	6/19/2022, 10:14:14 PM	0:00:16.136000	View Details	View Output
Untitled.ipynb	Completed	6/19/2022, 10:13:50 PM	0:00:17.396000	View Details	View Output
Untitled.ipynb	Completed	6/19/2022, 10:13:22 PM	0:00:16.210000	View Details	View Output

実行条件と、実行結果の確認

実行条件

The screenshot shows the "Notebook Execution History" section of the Amazon SageMaker Studio interface. It displays information for an "On-demand notebook execution" of a notebook named "exp002-demo-xgboost.ipynb" run at 6/20/2022, 12:22:07 AM. The status is completed, and the run time was 0:00:17.296000. The processing job info includes details about the job name, instance type (ml.m5.2xlarge), S3 locations for input and output, container image (notebook-runner), and IAM role. A "Close" button is visible at the bottom right.

実行結果（ノートブックのセル出力）

The screenshot shows the "Runs" tab in the Amazon SageMaker Studio interface. It displays the execution results of a cell in a notebook titled "exp002-demo-xgboost.ipynb". The cell contains Python code for importing xgboost, numpy, and sklearn, loading a breast cancer dataset, and splitting it into training and testing sets. A note indicates that this is a read-only preview, and it shows several warning messages related to pip version and root user permissions. The cell output is labeled [1], [2], and [3].

```
[1]: pip install xgboost
Requirement already satisfied: xgboost in /usr/local/lib/python3.7/site-packages (1.6.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/site-packages (from xgboost) (1.7.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/site-packages (from xgboost) (1.21.6)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.

[2]: import xgboost as xgb
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

[3]: """XGBoost で二値分類するサンプルコード"""
# 乳がんデータセットを読み込む
dataset = datasets.load_breast_cancer()
x, y = dataset.data, dataset.target
# データセットを学習用とテスト用に分割する
train_x, test_x, train_y, test_y = train_test_split(x, y,
                                                    test_size=0.2,
```

Tips: S3から直接DataFrameに読み込み

This is a read-only preview
To run and edit the notebook, create a copy to your workspace.

[1]: !pip install s3fs >/dev/null 2>&1

[2]: import pandas as pd

[3]: df = pd.read_csv('s3://demo-sagemaker-autopilot/input/churn.csv')

[4]: df

[4]:

.....

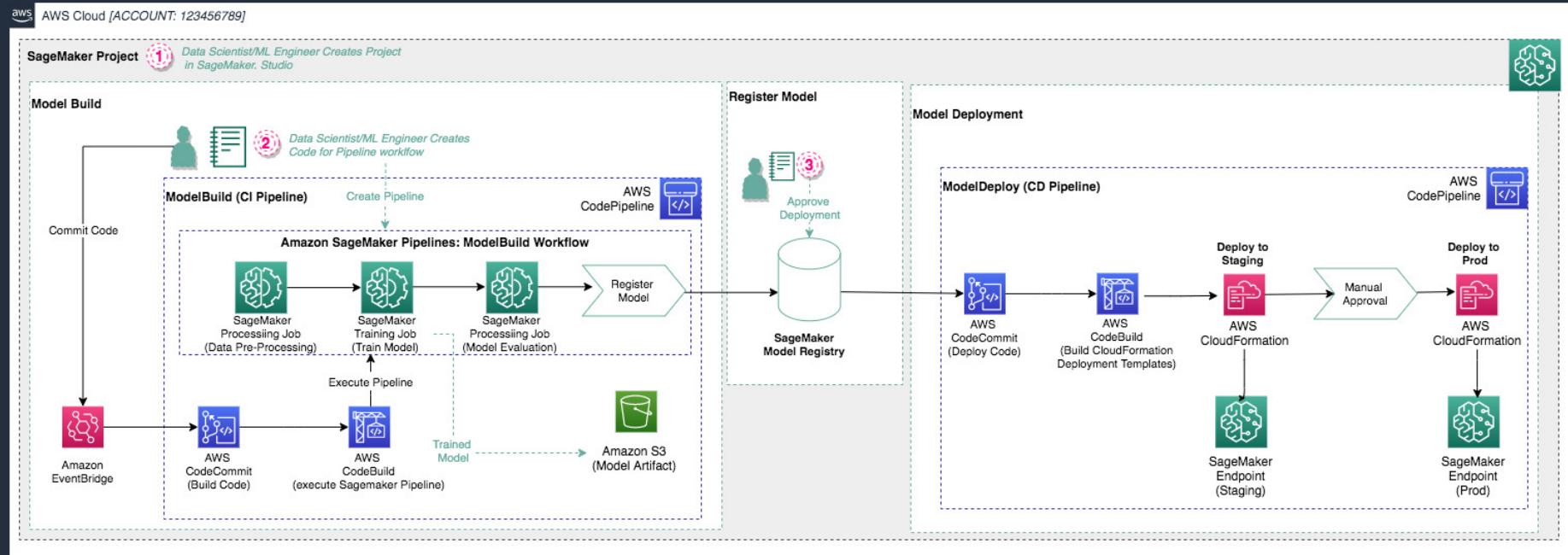
.....

	State	Account Length	Area Code	Phone	Intl Plan	VMail Plan	VMail Message	Day Mins	Day Calls	Day Charge	...	Eve Calls	Eve Charge	Night Mins	Night Calls	Night Charge	Intl Mins
0	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	...	99	16.78	244.7	91	11.01	10.0
1	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	...	103	16.62	254.4	103	11.45	13.7
2	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	...	110	10.30	162.6	104	7.32	12.2
3	OH	84	408	375-9999	yes	no	0	299.4	71	50.90	...	88	5.26	196.9	89	8.86	6.6
4	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	...	122	12.61	186.9	121	8.41	10.1
...	
3328	AZ	192	415	414-4276	no	yes	36	156.2	77	26.55	...	126	18.32	279.1	83	12.56	9.9

SageMaker Pipelines

SageMaker Studio から数クリックで構築可能

使い方は動画後半で



- ・ MLOpsエンジニアはパイプラインを構築・管理する
- ・ 成果物をリポジトリに提出するルールとすることで、ガバナンスを確立
- ・ データサイエンティストはコーディング、リポジトリへのpushを担当し、モデル構築に注力する

データ

SageMakerジョブと同等

環境

SageMakerジョブと同等

ソース・設定

リポジトリに保存される

振り返り

SageMakerジョブと同等



<https://aws.amazon.com/blogs/machine-learning/building-automating-managing-and-scaling-ml-workflows-using-amazon-sagemaker-pipelines/>

SageMaker Studioから、Projects を選択

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with icons for SageMaker resources, Projects, and other components like Pipelines and Experiments. A message says "There are no Projects yet. Create a Project using the SageMaker SDK and track your work automatically." The main area has a "Create project" dialog open. The dialog title is "Create project" and it contains the following text:

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

Below this, there's a section titled "SageMaker project templates" with two tabs: "Organization templates" (selected) and "SageMaker templates". Under "SageMaker templates", there's a table with columns "Name" and "Description". The table lists several MLOps templates:

Name	Description
MLOps template for model building, training, deployment and monit...	Use this template to automate the entire model lifecycle that includ...
MLOps template for image building, model building, and model depl...	Use this template to build an image that is used to train a model and...
MLOps template for model building, training, and deployment with t...	Use this template to automate the entire model lifecycle that includ...
MLOps template for model building, training, and deployment	Use this template to automate the entire model lifecycle that includ...
MLOps template for model deployment	Use this template to automate the deployment of models in the Am...
MLOps template for model building and training	Use this template to automate the model building workflow. Process...
MLOps template for model building, training, and deployment with t...	Use this template to automate the entire model lifecycle that includ...

At the bottom of the dialog, there are buttons for "Select project template" and "Create project".

テンプレートを展開し、構築完了

パイプライン完成(5分程度)

The screenshot shows the 'Repositories' section of the Amazon SageMaker Studio interface. A prominent green success message at the top reads 'Successfully created project-customer-churn-predict project.' Below this, there are tabs for 'Repositories', 'Pipelines', 'Experiments', 'Model groups', 'Endpoints', and 'Settings'. The 'Repositories' tab is selected. A table lists two repositories:

Name	Local path	URI	Last modified
sagemaker-project-customer-churn-p...	No local path clone repo...	https://git-codecommit.ap-northeast-1.amazonaws.com/v1/repos/sagemaker-project-customer-churn-predict	3 minutes ago
sagemaker-project-customer-churn-p...	No local path clone repo...	https://git-codecommit.ap-northeast-1.amazonaws.com/v1/repos/sagemaker-project-customer-churn-predict-p-xb3ea01coqye-modelbuild	3 minutes ago

At the bottom of the table, it says 'End of the list'.

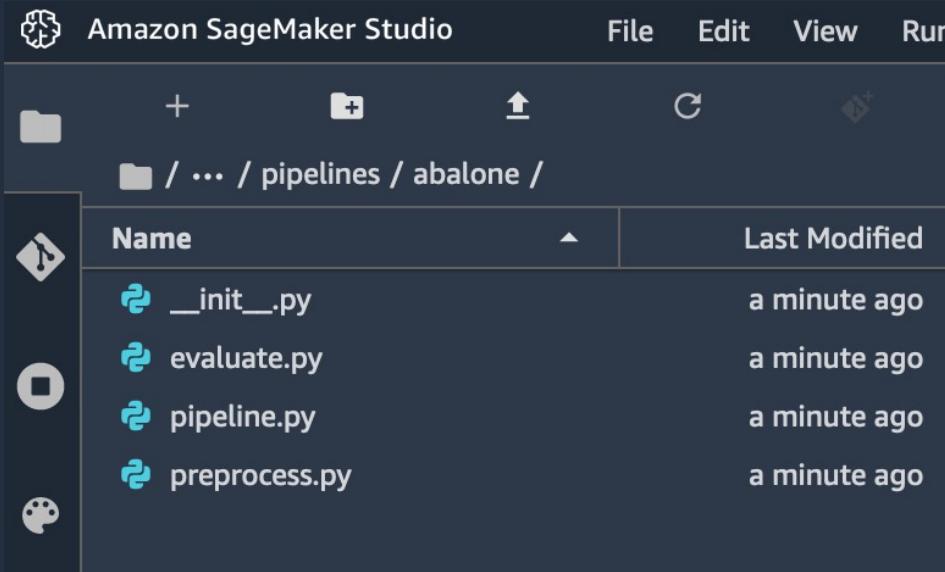
Cloneされたファイル群

The screenshot shows the file browser in Amazon SageMaker Studio. The path displayed is '/project-customer-churn-predict-p-xb3ea01coqye/sagemaker-project-customer-churn-predict-p-xb3ea01coqye-modelbuild/'. The left sidebar shows icons for different file types: folder, Python script, Jupyter notebook, test file, buildspec, CONTRIBUTING.md, LICENSE, README.md, Pipfile, setup.cfg, setup.py, and tox.ini. The right pane lists these files with their names and last modified times. All files were modified 'seconds ago'.

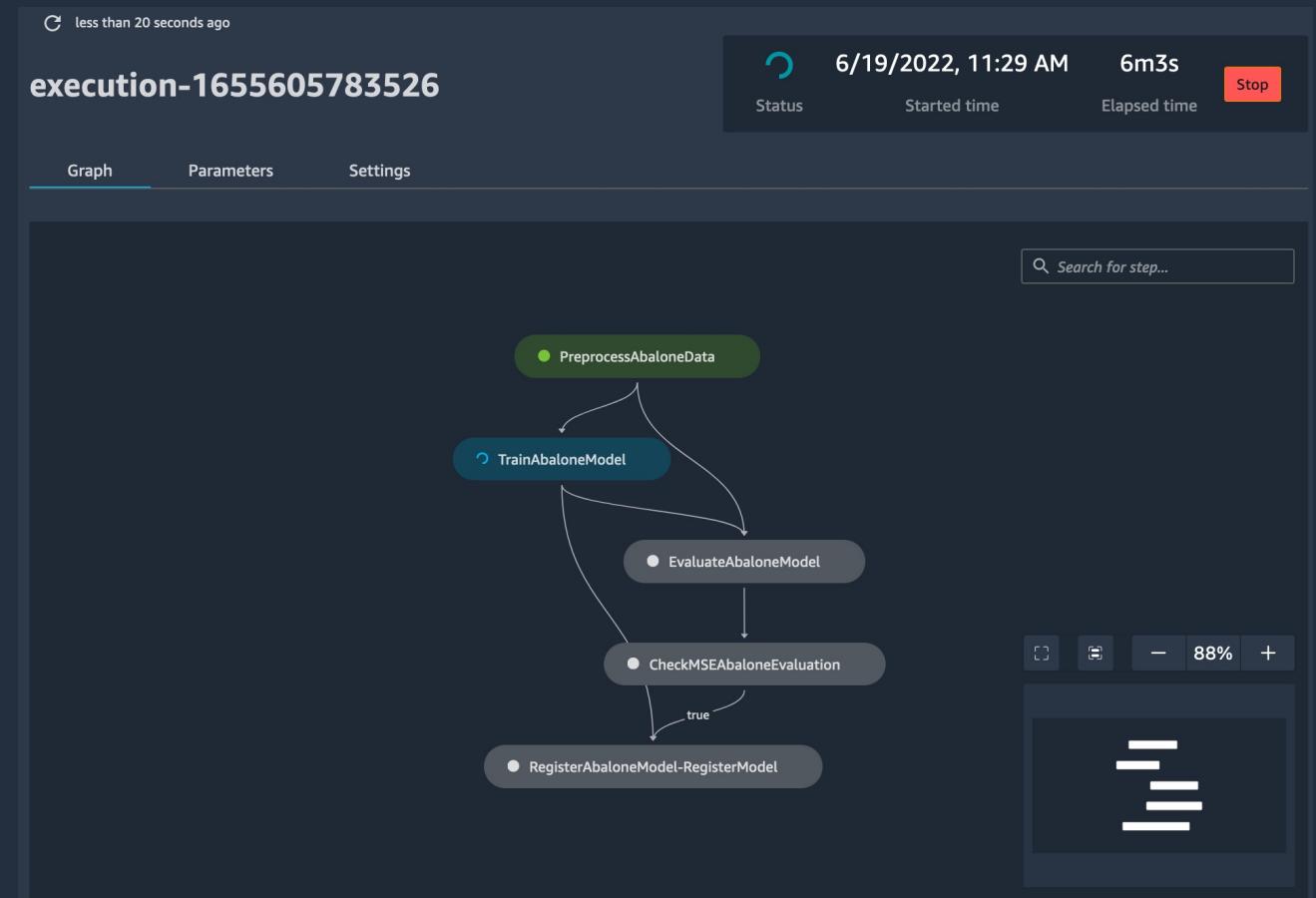
Name	Last Modified
img	seconds ago
pipelines	seconds ago
tests	seconds ago
codebuild-buildspec.yml	seconds ago
CONTRIBUTING.md	seconds ago
LICENSE	seconds ago
README.md	seconds ago
sagemaker-pipelines-project.ip...	seconds ago
setup.cfg	seconds ago
setup.py	seconds ago
tox.ini	seconds ago

リポジトリpushをトリガに、パイプラインを実行

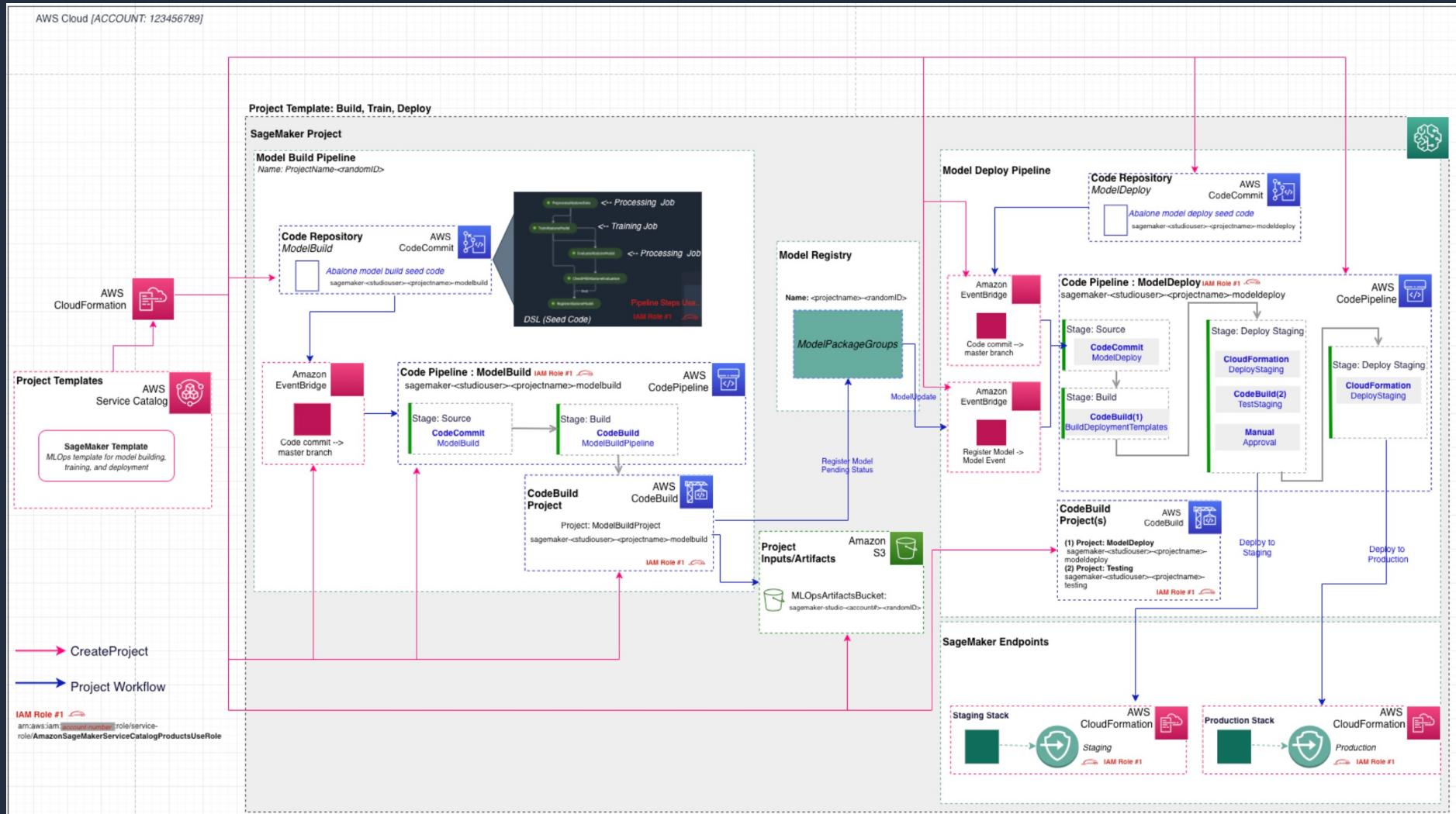
コードを変更してpush



パイプラインが実行される

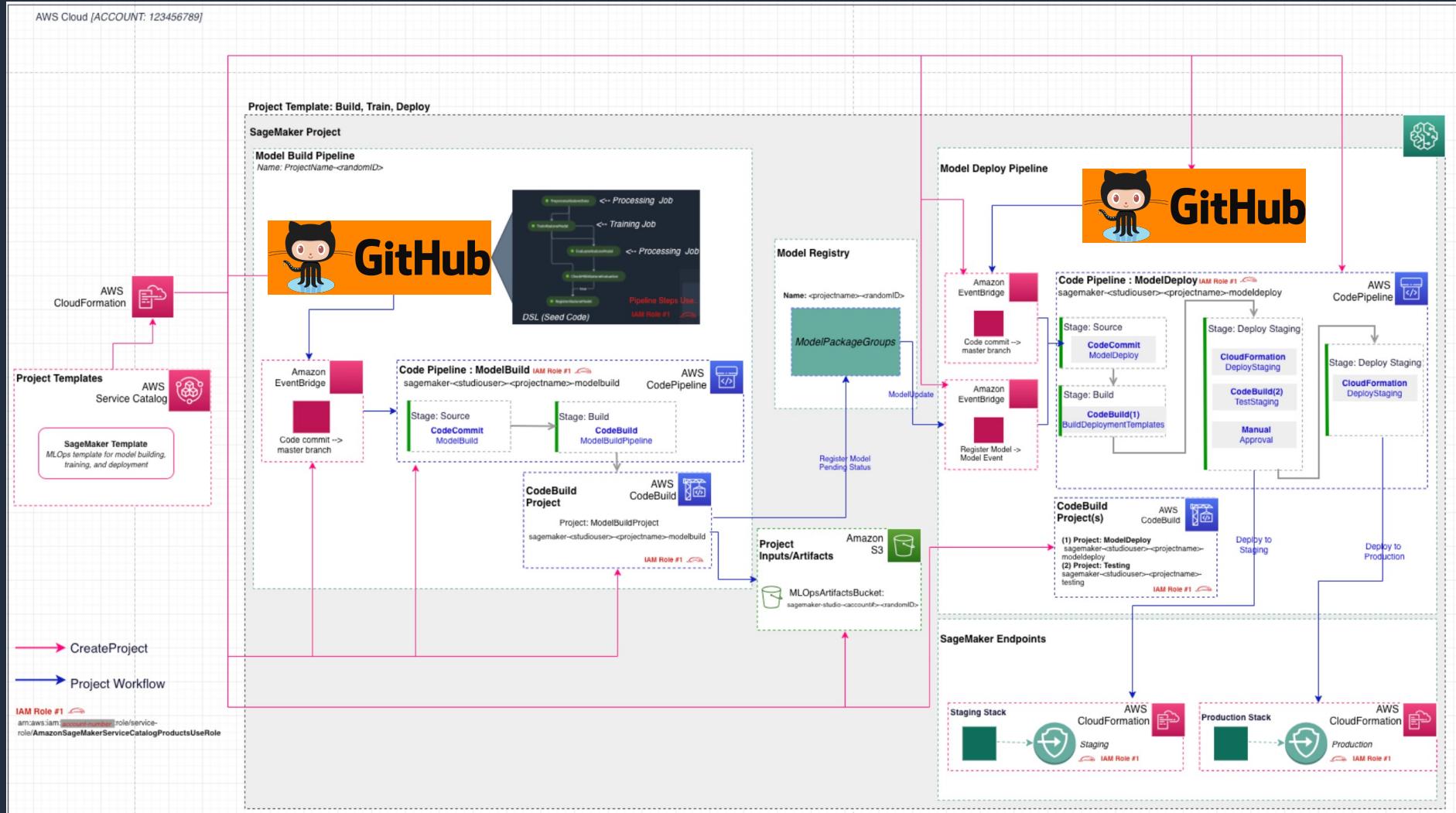


参考：アーキテクチャ - AWS CodeCommit 版



<https://catalog.us-east-1.prod.workshops.aws/workshops/63069e26-921c-4ce1-9cc7-dd882ff62575/ja-JP/lab6>

参考：アーキテクチャ - GitHub 版



AWS CodeStar Connectionの作成

コネクション作成

```
aws codestar-connections create-connection \
--provider-type GitHub \
--connection-name MyConnection \
--tags Key=sagemaker,Value=true
```

コネクション確認

```
aws codestar-connections list-connections
```

タグ確認

```
aws codestar-connections list-tags-for-resource \
--resource-arn <codestar-connections ARN>
```

この時点では[Pending]ステータス

```
aws CloudShell
ap-northeast-1
① If the arrow keys aren't working correctly in PowerShell, see Troubleshooting AWS CloudShell
Don't show this message again
Preparing your terminal...
[cloudshell-user@ip-10-0-174-255 ~]$ Try these commands to get started:
aws help or aws <command> help or aws <command> --cli-auto-prompt
[cloudshell-user@ip-10-0-174-255 ~]$ aws codestar-connections create-connection \
>   --provider-type GitHub \
>   --connection-name MyConnection \
>   --tags Key=sagemaker,Value=true
{
  "ConnectionArn": "arn:aws:codestar-connections:ap-northeast-1:510361903830:connection/f8fb5116-b5cc-4fad-bef0-ef647bfc255",
  "Tags": [
    {
      "Key": "sagemaker",
      "Value": "true"
    }
  ]
}

[cloudshell-user@ip-10-0-174-255 ~]$ aws codestar-connections list-connections
{
  "Connections": [
    {
      "ConnectionName": "MyConnection",
      "ConnectionArn": "arn:aws:codestar-connections:ap-northeast-1:510361903830:connection/f8fb5116-b5cc-4fad-bef0-ef647bfc255",
      "ProviderType": "GitHub",
      "OwnerAccountId": "510361903830",
      "ConnectionStatus": "PENDING"
    }
  ]
}
```

```
[cloudshell-user@ip-10-0-174-255 ~]$ aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codesta
r-connections:ap-northeast-1:510361903830:connection/f8fb5116-b5cc-4fad-bef0-ef647bfc255
{
  "Tags": [
    {
      "Key": "sagemaker",
      "Value": "true"
    }
  ]
}
```

Connection の Available 化

マネジメントコンソール上から、作成したGitHubへのコネクションをactivateする

The screenshot shows the AWS Management Console interface for Developer Tools, specifically the Connections section. A success message at the top indicates that the connection 'MyConnection' was updated successfully. The main table lists one connection entry:

Connection name	Provider	Status	ARN
MyConnection	GitHub	Available	arn:aws:codestar-connections:ap-northeast-1:510361903830:connection/f8fb5116-b5cc-4fad-bef0-ef647bfcd255

A red box highlights the ARN value in the table. Below the table, an orange text overlay reads: "SageMaker Pipelines のテンプレート入力データに利用".

<https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-projects-walkthrough-3rdgit.html>



© 2022, Amazon Web Services, Inc. or its affiliates.

テンプレート入力（ビルト側設定）

The screenshot shows the 'Create project' dialog in Amazon SageMaker Studio. The 'Project details' section includes fields for 'Name' (sm-pipelines-github), 'Description - optional', and 'Tags - optional'. The 'Project template parameters' section includes fields for 'ModelBuild CodeRepository Info': 'URL' (https://github.com/yito0427/test-sm-build), 'Branch' (main), 'Full Repository Name' (yito0427/test-sm-build), and 'Codestar Connection ARN' (:510361903830:connection/f8fb5116-b5cc-4fad-bef0-ef647bfcd255). A large red box highlights the URL, Branch, Full Repository Name, and Codestar Connection ARN fields. To the right of these highlighted fields are Japanese labels: 'GitHubリポジトリURL', 'ブランチ名', 'アカウント名/リポジトリ名', and 'コネクションARN'. At the bottom right of the dialog, the text 'デプロイ側（スクロール）も同様に設定' (Deployment side (scroll) is also set similarly) is displayed.

GitHubリポジトリURL

ブランチ名

アカウント名/リポジトリ名

コネクションARN

デプロイ側（スクロール）も同様に設定

構築後、GitHubへのpushでパイプラインが起動

リファレンス

Amazon SageMaker Experiments Python SDK doc

<https://sagemaker-experiments.readthedocs.io/en/latest/#>

SageMaker Experiments Python SDK

<https://github.com/aws/sagemaker-experiments>

GitHub samples

<https://github.com/aws/amazon-sagemaker-examples/tree/main/sagemaker-experiments>

Developer Guide

<https://docs.aws.amazon.com/sagemaker/latest/dg/experiments.html>

次回予告 (Dark Part)

Amazon SageMaker Training (デモ編)

前回の動画「Amazon SageMaker Training (座学編)」で紹介したコードのデモを実施します



ML Enablement Seriesの動画

機械学習モデルをビジネス価値につなげる方法を全力で解説！

Light Part

製品やサービスに機械学習を導入するプロジェクトの進め方

<https://bit.ly/3M1F9as>

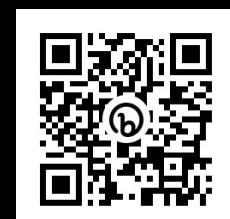


Step Up!!

Dark Part

機械学習モデルの開発や運用をマネージドサービスで効率的に行う方法

<https://bit.ly/3927PCN>



資料集・お問合せ・Special Thanks

AWSの日本語資料の場所: 「AWS 資料」で検索

The screenshot shows the AWS Documentation homepage in Japanese. The top navigation bar includes links for 'お問い合わせ' (Contact), 'サポート' (Support), '日本語' (Japanese), 'アカウント' (Account), and a prominent orange '今すぐ無料サインアップ' (Sign up now) button. Below the navigation is a secondary menu with links to '製品', 'ソリューション', '料金', 'ドキュメント', '学ぶ', 'パートナーネットワーク', 'AWS Marketplace', 'イベント', and 'さらに詳しく見る'. A search bar is also present. The main content area features a large title 'AWS クラウドサービス活用資料集トップ' (Top of the AWS Cloud Service Utilization Document Collection). Below it is a detailed description of what the site offers, followed by four call-to-action buttons: 'AWS Webinar お申込」 (Apply for AWS Webinar), 'AWS 初心者向け」 (For beginners), 'サービス別資料」 (Service-specific documentation), and 'ハンズオン資料」 (Hands-on documentation).

AWSのハンズオン資料の場所: 「AWS ハンズオン」で検索

The screenshot shows the AWS Hands-on Documentation homepage in Japanese. The layout is identical to the AWS Documentation page, featuring the same top navigation bar, secondary menu, and search bar. The main title is 'AWS クラウドサービス活用資料集トップ'. The description below the title is identical to the documentation page. The bottom section contains the same four call-to-action buttons: 'AWS Webinar お申込」, 'AWS 初心者向け」, 'サービス別資料」, and 'ハンズオン資料」.

お問合せ

技術的なお問合せ

料金のお問合せ

個別相談会のお申込み

Special Thanks

- 音楽素材: [PANICPUMPKIN](#)様



Thank you!