

機械学習モデルの開発環境を構築する

ML Enablement Series: Environment Setup

AWS Black Belt Online Seminar

久保 隆宏

Developer Relation

Machine Learning



本動画のゴール

- Pythonで機械学習モデルを開発する環境を構築できるようになる。
 - 開発環境の仕様についてDevOpsチームと合意を取れるようになる。
-

ML Enablement Seriesの特徴

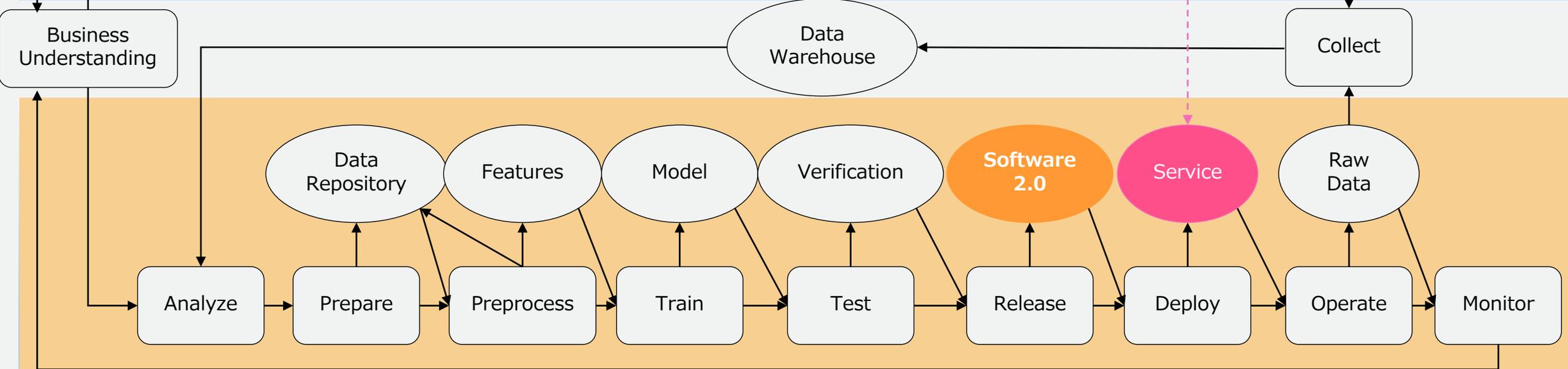
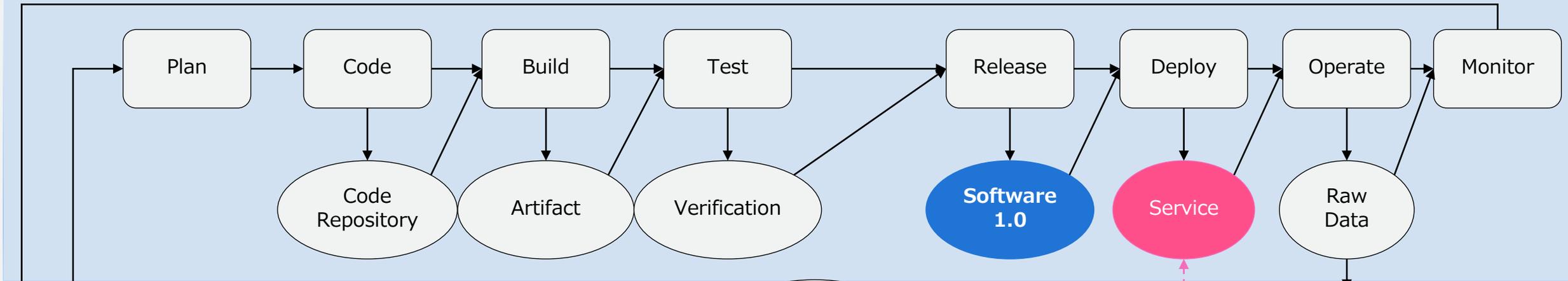
ML Enablement Seriesのゴール

データサイエンティストを含むプロダクト開発チームが、機械学習モデルの開発において足りないスキルを補い合い、顧客に価値を届けられるようになること。

チームでのコミュニケーションが不可欠

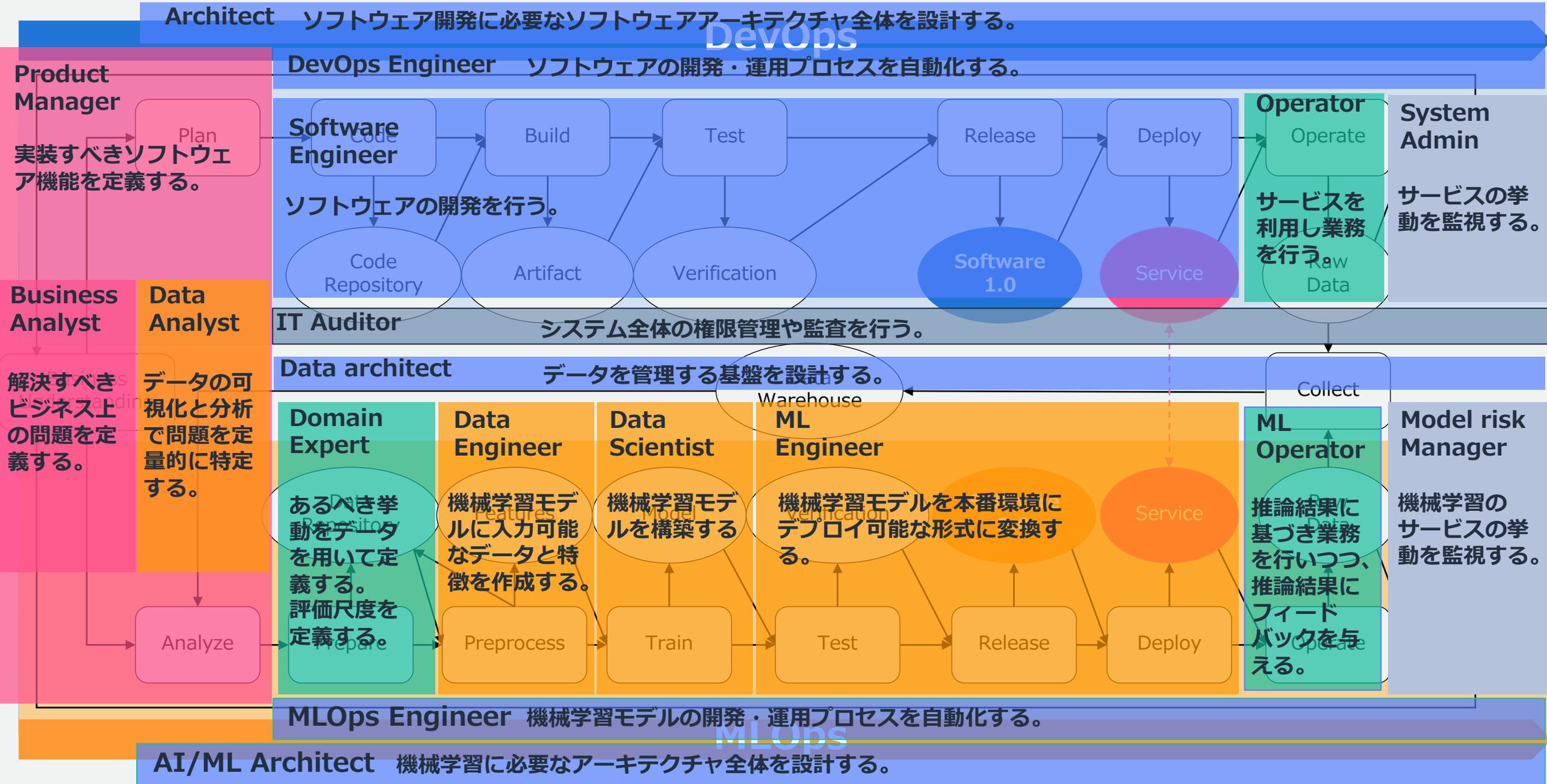
機械学習モデルを活用するプロダクトでは、DevOpsチームとMLOpsチームが連携する必要がある。

DevOps



MLOps

プロダクトのリリースまでには多くのロールが関わる。



ML Enablement Seriesの解説の流れと役立つポイント

1. What : なにをするのか?

2. Why : なぜやるのか?

3. How : どうやるのか?

3.1 プログラミング演習

3.2 コミュニケーション演習



チームでの認識合わせに役立つ

具体的な開発方法を学ぶのに役立つ

会議体などを設計するのに役立つ

機械学習モデルを実装する方法に興味がある方 ⇒ 3.1

開発プロジェクトのマネジメントに興味がある方 ⇒ 1~3, 3.2

「機械学習モデルの開発環境を構築する」での解説

DevOps

Plan

Code

Build

Test

Release

Deploy

Operate

Monitor

開発を始める前のEnvironment Setup:環境構築について解説

- プログラミング演習

Condaで開発環境を構築する手順を解説

- コミュニケーション演習

開発環境を構築する際に行うべきコミュニケーションを解説

Business
Understand

Analyze

Prepare

Preprocess

Train

Test

Release

Deploy

Operate

Monitor

Data
Repository

Features

Model

Verification

Software
2.0

Service

Raw
Data

MLOps

1. What: なにをするのか?

環境構築プロセスのゴール

データ分析に必要な環境の仕様と構築に合意する

①

②

③

1. データ分析
ビジネスゴールを達成するために今あるデータが必要十分か検証するプロセス(図中: Analyze)。
2. 環境の仕様
開発環境の構成図、Pythonやライブラリのバージョンなど。
3. 合意
MLOpsチーム内だけでなく、DevOpsチームやIT監督者といった機械学習モデルのリリースに必要な関係者の合意。

環境構築プロセスで最も重要な成果物

開発環境の構築手順書

環境が構築されたPCやサーバーに対し「合意」することは難しいので、構築手順を書き起こして合意する。

手順書はGitHub、AWSの場合CodeCommitなどバージョン管理できるプラットフォーム上に保存するのが好ましいです。

開発環境の構築手順書の例

Wantedly Engineering Handbook

<https://docs.wantedly.dev/fields/dev-tools/codespaces>

The screenshot shows the 'Wantedly Engineering Handbook' website. The main content area is titled 'GitHub Codespaces で開発する' (Developing with GitHub Codespaces). Under the 'Why' section, there are three bullet points: '開発時のローカルマシンの負荷を下げたい' (I want to reduce the load on the local machine during development), '開発を開始する際のローカルマシンの環境構築を軽くしたい or なくしたい' (I want to make the environment setup on the local machine lighter or none when starting development), and '特にインターンなどを受け入れる際のオンボーディングコストを下げたい' (I want to reduce the onboarding cost, especially when accepting interns, etc.). Under the 'VSCode で使う' (Using VSCode) section, there is a sub-section '事前準備' (Preparation). A dashed orange box highlights the 'Point' (Points) section, which lists: '見やすい' (Easy to see), '見つけやすい' (Easy to find), and '更新履歴が管理されている。' (Update history is managed).

Wantedly Engineering Handbook

Podcast YouTube Blog Twitter

検索...

Infrastructure >

Data >

開発プロセス >

開発ツール ▾

kube

GitHub Codespaces で開発する

Code Coverage

Kubefork

おわりに

ロードマップ(未執筆)

Handbook の書き方

コントリビューター

GitHub Codespaces で開発する

Why

- 開発時のローカルマシンの負荷を下げたい
- 開発を開始する際のローカルマシンの環境構築を軽くしたい or なくしたい
 - 特にインターンなどを受け入れる際のオンボーディングコストを下げたい

VSCode で使う

事前準備

Point

- 見やすい
- 見つけやすい
- 更新履歴が管理されている。

リンクのコピー

目次

Why

VSCode で使う

事前準備

新しい Codespace を作る

既に存在する Codespace をアタッ...

Codespace を終了する

2. Why: なぜするのか?

プロダクション環境へのリリースを見据えて開発するため

MLOpsチームにDevOpsチーム出身者がいない時、機械学習モデルをリリースする段になって問題が発生しがち。

- DevOps: リリースに向けてCI/CDの仕組みを作ろうと思います。機械学習モデルを動かすための環境構築手順はありますか?
 - MLOps: え、ないです。開発者の端末に環境はありますが・・・
- DevOps: なるほど。Pythonやライブラリのバージョンは分かかりますか?
 - MLOps: はい。Pythonは3.xですね。
- DevOps: ははあ、こちらで使っているバージョンとは違いますね。Dockerコンテナ化はされていますか?コンテナ化されていれば・・・
 - MLOps: してないですね・・・

DevOpsチームにかかる負担を軽くする

DevOpsチームはソフトウェア開発で忙しいので、リソースを割いて機械学習モデルを組み込むのは大変。あまりに手間だと、リリースが遅れるかできなくなることも・・・

プロダクションで推論しない機械学習モデルは車に搭載されていないエンジンみたいなもので、性能が良くても価値を生まない。

機械学習モデルの価値を顧客に届けるために、環境構築手順を合意することでリリースに向けた道筋をつけておくことが大切。

3. How: どうやるのか?

演習の概要

3.1 プログラミング演習

- Condaで開発環境を構築する手順を学びます。

3.2 コミュニケーション演習

- 開発環境を構築する際に行うべきコミュニケーションを学びます。

※もちろん、環境構築の手順やツールは本来DevOpsチームと協議して決めます。

3.1 プログラミング演習

プログラミング演習の手順

1. SageMaker Studio Lab (JupyterLab環境)を起動
2. git cloneでファイルを取得
3. Condaを使用しPythonとライブラリをインストールする
4. インストールしたライブラリを記録したファイルを作成する

※オンプレミスやクラウドでJupyterLabを起動する環境を作るにはネットワークや権限の設計、構築が必要です。今回はそうした構築を行わず無料の環境でまず試すというシーンを想定し SageMaker Studio Labを使用します。

SageMaker Studio Labはメールアドレスのみ、無料でブラウザからJupyterLab環境が利用できるサービスです。

Amazon SageMaker Studio Labとは

初学者にも使いやすい
無料の機械学習環境です

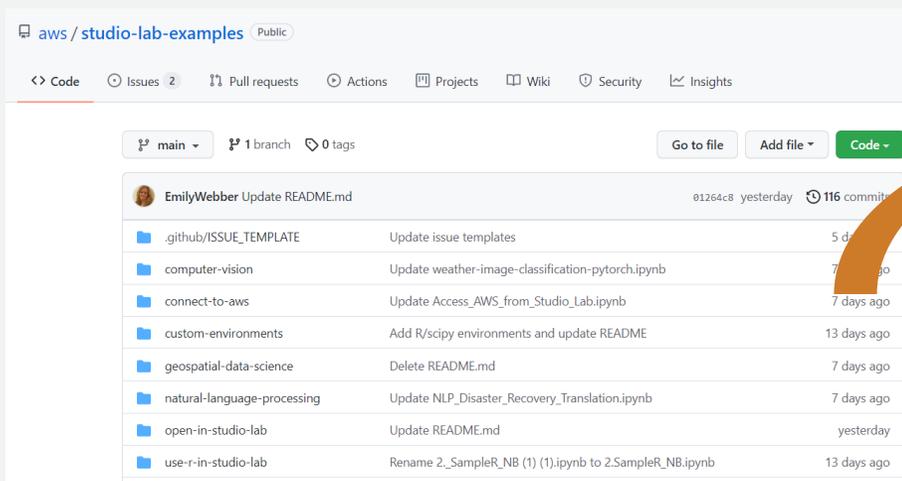


機能とスペック

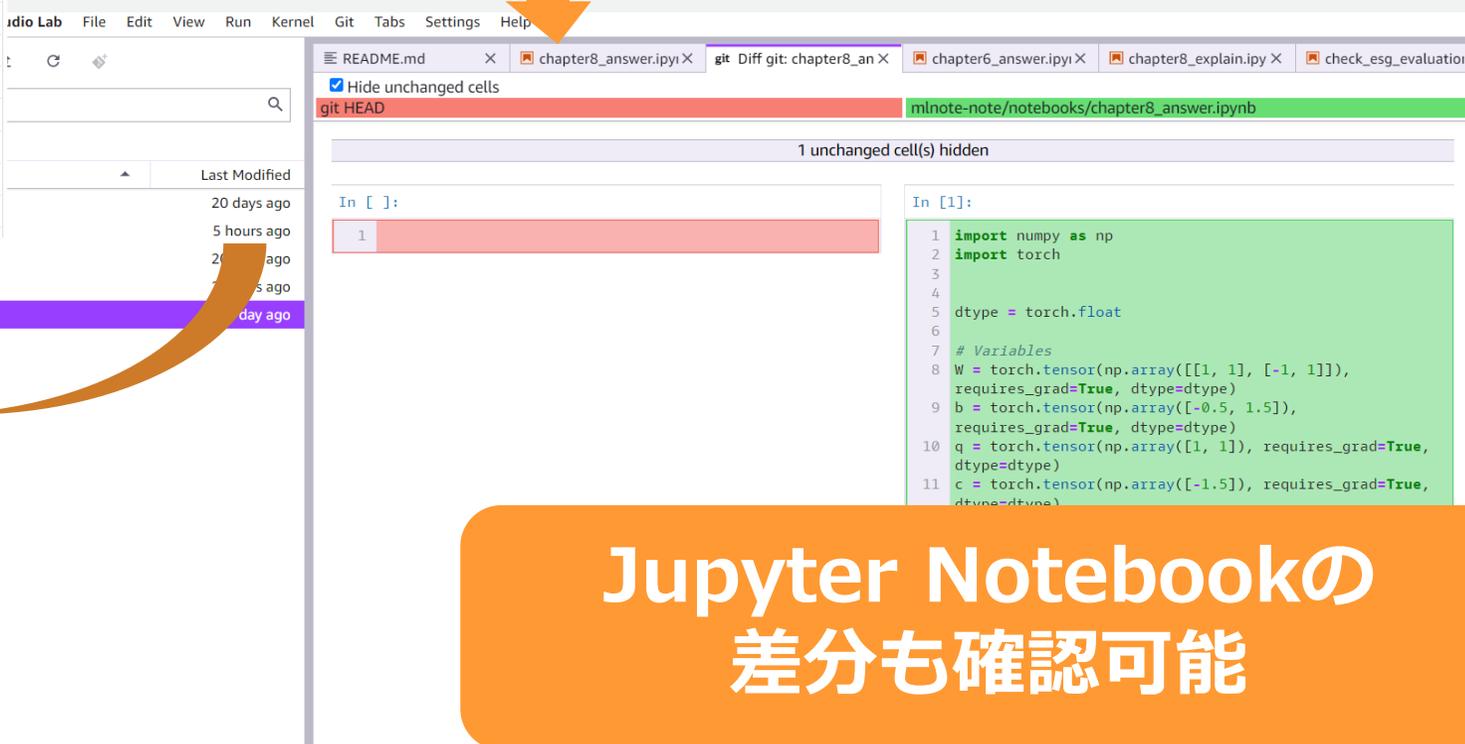
機械学習の学習に十分な
機能とスペックを備えています



ターミナルでもGUIでもGitの操作が可能



git clone/git pull

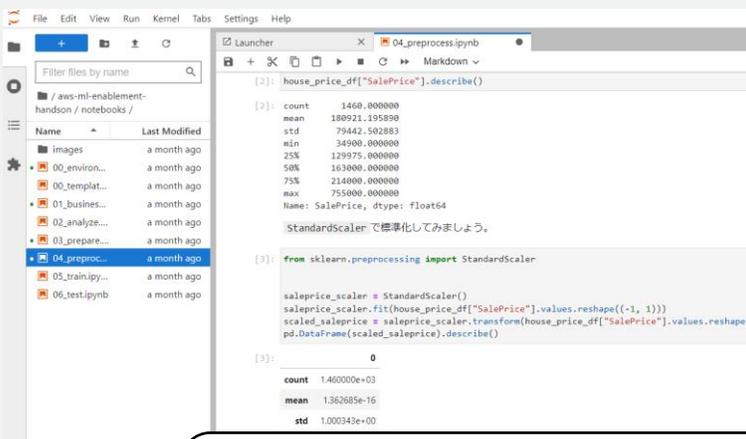


git push

**Jupyter Notebookの
差分も確認可能**

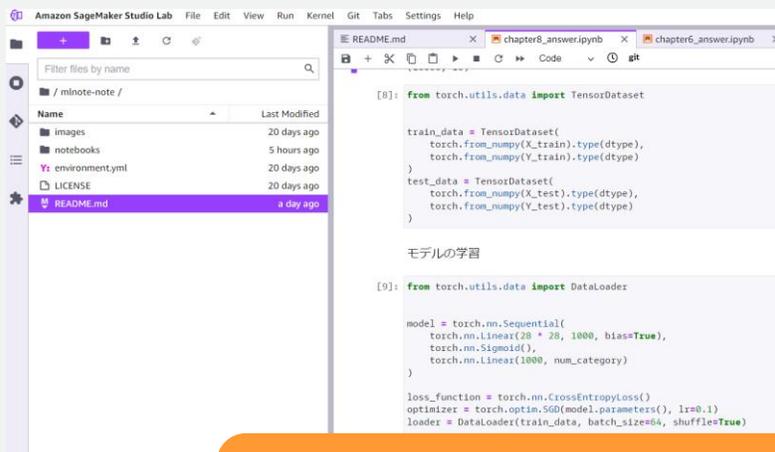
SageMakerはStudio Labも含めてオープンソースのJupyterLab環境を提供しています。

JupyterLab(OSS)



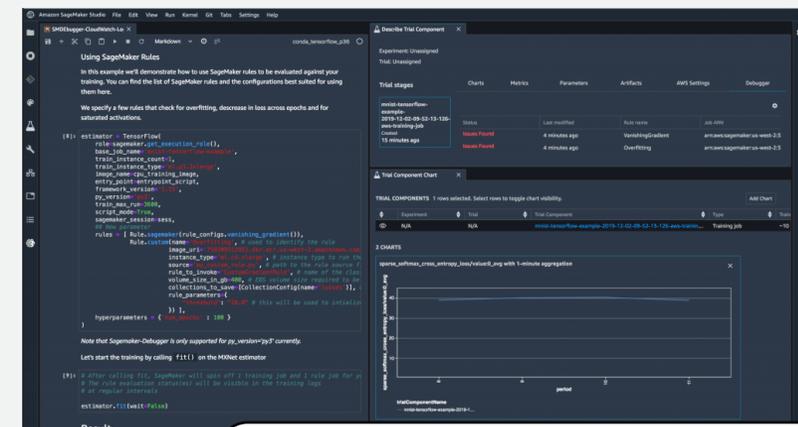
開発者のPC/オンプレミスのサーバーで稼働

SageMaker Studio Lab



チーム間で統一かつ十分なCPU/GPUのある環境

SageMaker Studio

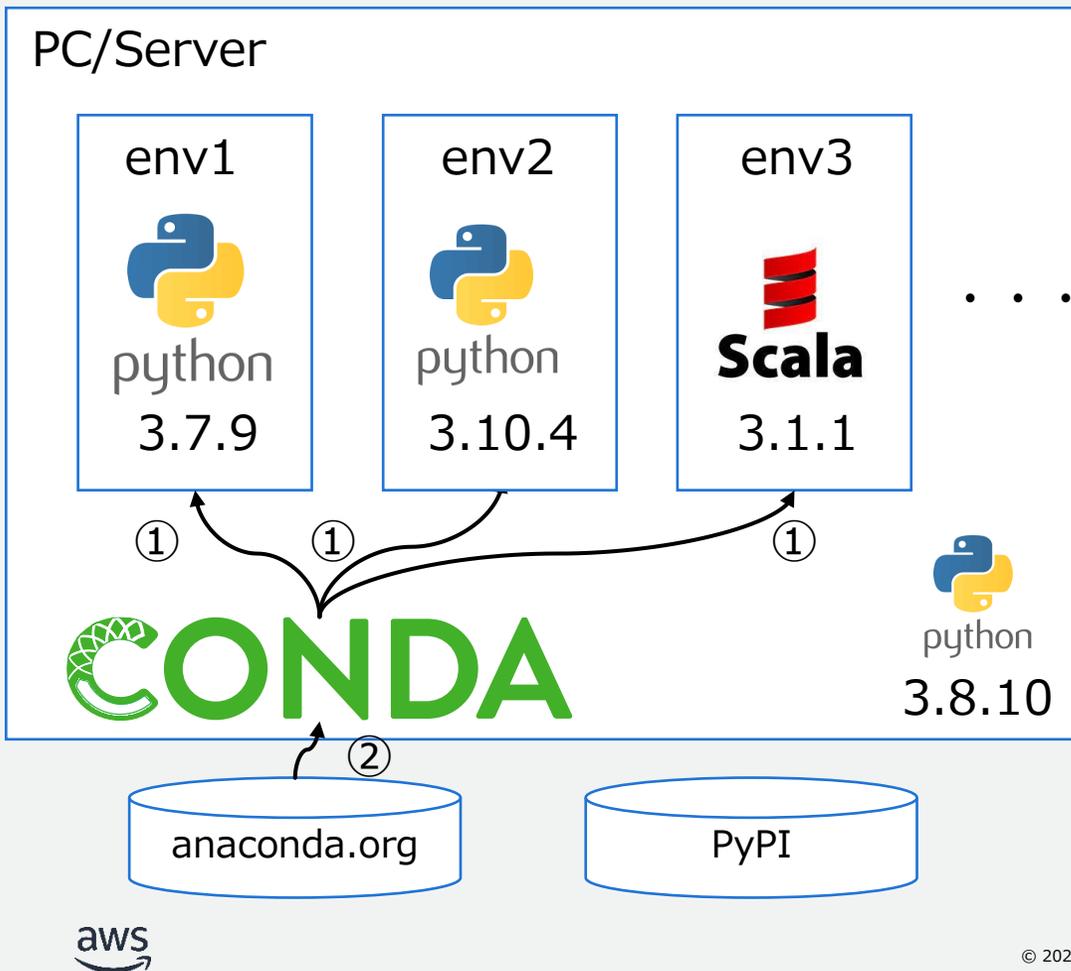


エンタープライズレベルのインフラがある環境

ローカル(オンプレミス)からクラウド上での開発まで一貫した開発体験を提供

Condaとは

プロジェクトごとの開発環境(仮想環境)を作成、管理するためのオープンソース。Python以外の言語の環境も構築できる。



virtualenv(venv)・pip・pipenv・poetryとの差異

virtualenv(venv)・pipenv・poetryについて、仮想環境が作れる点は同じ。CondaはシステムにインストールされているPythonと別個にPythonをインストールして実行環境が作成できる点が異なる(①)。pyenvと同じ機能を持つ。

pipenv・poetry・pipとはパッケージのインストールができる点は同じ。CondaはPythonパッケージ以外にPython本体やPythonとは関係がないライブラリもインストールすることができる。(そのため)PyPIからインストールしない点が異なる(②)。Condaで作成した環境でpipを利用することは可能。

※同じパッケージをCondaとpip両方でインストールしないよう注意

ツール名	開発	特徴・歴史的経緯
virtualenv	PyPA	もっと歴史が長い?仮想環境の構築ツール。
venv	PyPA	virtualenvの機能が一部切り出され、Python 3.3から標準で付属。
pip	PyPA	Python公式のパッケージ管理ツール。Python 3.4から標準で付属。
pipenv	PyPA	仮想環境の作成はvirtualenv、ライブラリのインストールはpip、と使い分けるのではなく一つのツールで済むよう開発されている。
Conda	Anaconda, Inc.	Pythonパッケージが集まるリポジトリであるPyPIではなく、 anaconda.org からWindowsやLinuxなど環境に応じたバイナリをインストールする。 PythonのパッケージはPython以外のライブラリ(HDF5やMKL、LLVMなど)に依存するにもかかわらずpipでインストールできないためインストールのビルド時にエラーになることがしばしばあった。コンパイル済みのバイナリ、Python以外のライブラリインストールできるようにしたもの(※Wheelが登場しpipでもバイナリがインストールできるようになった)。
poetry	Sébastien Eustace (python-poetry)	PyPIへのパッケージ公開までサポートする。Pythonのプロジェクト(パッケージ)情報をpyproject.toml一本にまとめて管理する方針。PyPIのJSON APIが使えれば依存関係解決に使用するため、手元にダウンロードしてbuildするpipenvよりもlockが高速。

Condaとは

CondaとPythonのパッケージを組み合わせた配布形態が複数あります。Anacondaは商用利用で有償となるケースがあるので注意してください。



conda + Python



conda + Python + 少数の
package



conda + Python + 1,500 package
※大規模な商用利用では有償

プログラミング演習の開始

Open
Handson

プログラミング演習

1. SageMaker Studio Lab (JupyterLab環境)を起動

2. git cloneでファイルを取得

3. Condaを使用しPythonとライブラリをインストールする

4. インストールしたライブラリを記録したファイルを作成する

プログラミング演習

1. SageMaker Studio Lab (JupyterLab環境)を起動
- 2. git cloneでファイルを取得**
3. Condaを使用しPythonとライブラリをインストールする
4. インストールしたライブラリを記録したファイルを作成する

<https://github.com/aws-samples/aws-ml-enablement-handson>



プログラミング演習

1. SageMaker Studio Lab (JupyterLab環境)を起動
2. git cloneでファイルを取得
- 3. Condaを使用しPythonとライブラリをインストールする**
4. インストールしたライブラリを記録したファイルを作成する

プログラミング演習

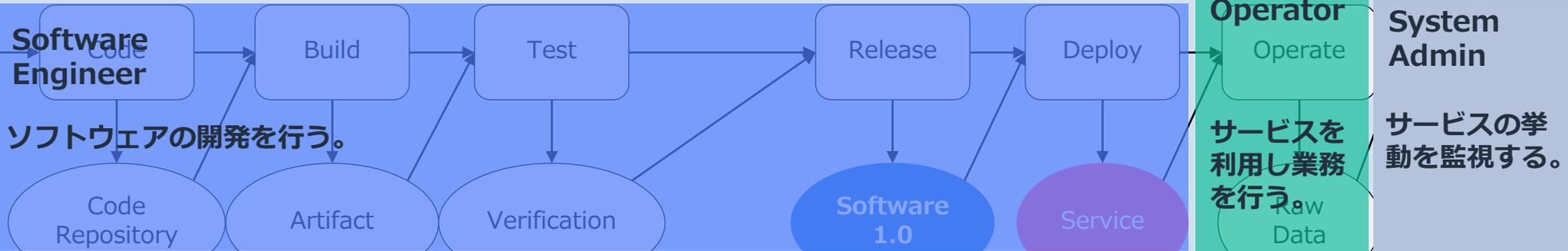
1. SageMaker Studio Lab (JupyterLab環境)を起動
2. git cloneでファイルを取得
3. Condaを使用しPythonとライブラリをインストールする
- 4. インストールしたライブラリを記録したファイルを作成する**

3.2 コミュニケーション演習

コミュニケーション演習：誰に何を確認すべきか？

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。

DevOps Engineer ソフトウェアの開発・運用プロセスを自動化する。



Product Manager
実装すべきソフトウェア機能を定義する。

Business Analyst
解決すべきビジネス上の問題を定義する。

Data Analyst
データの可視化と分析で問題を定量的に特定する。

IT Auditor システム全体の権限管理や監査を行う。

Data architect データを管理する基盤を設計する。



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。

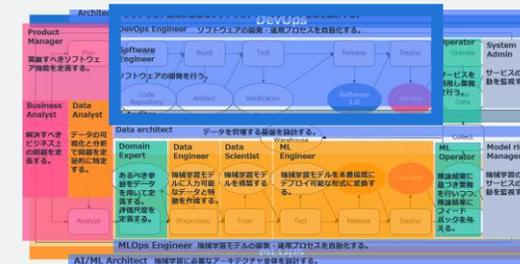
環境構築で行ったこと

1. SageMaker Studio Labを起動
2. git cloneでファイルを取得
3. Condaを使用しPythonとライブラリをインストールする
4. インストールしたライブラリを記録したファイルを作成する

開発環境の構築手順を決めるとき、誰に何を依頼すべきか？

- 情報共有 : 開発に必要な情報/ドキュメントの提供を依頼する
- 作業 : 開発に必要な作業を分担(依頼)する
- レビュー : 作業の確度を上げるため/承認を得るためレビューを依頼する

コミュニケーション演習: DevOpsチーム編



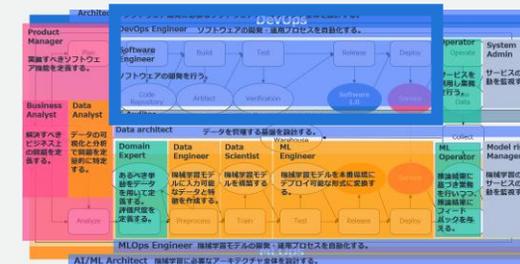
Who: サービス開発を担当している人

英語名	日本語名	役割
Architect	アーキテクト	ソフトウェアアーキテクチャ全体を設計する。
DevOps Engineer	DevOpsエンジニア	ソフトウェアの開発・運用プロセスを自動化する。
Software Engineer	ソフトウェアエンジニア	ソフトウェアの開発を行う。
Architect	アーキテクト	ソフトウェアアーキテクチャ全体を設計する。

What: 何を依頼すべきか?

- 情報共有
 - ソースコード管理にどんなサービスを利用しているか。ブランチ運用ルールはあるか。
 - 開発環境として利用してよい環境はあるか(AWSアカウントなど)。
 - 現在の開発でPythonは利用しているか。利用しているPythonのバージョンはなにか。
 - コーディングルールなどの開発標準があるか。

コミュニケーション演習：DevOpsチーム編



Who: サービス開発を担当している人

英語名	日本語名	役割
Architect	アーキテクト	ソフトウェアアーキテクチャ全体を設計する。
DevOps Engineer	DevOpsエンジニア	ソフトウェアの開発・運用プロセスを自動化する。
Software Engineer	ソフトウェアエンジニア	ソフトウェアの開発を行う。
Architect	アーキテクト	ソフトウェアアーキテクチャ全体を設計する。

What: 何を依頼すべきか?

- 作業
 - 開発環境にアクセスするためのIDの払い出し。
- レビュー
 - 作成した環境構築手順書のレビュー。

コミュニケーション演習: ITインフラ管理者編



Who: ITインフラを管理している人

英語名	日本語名	役割
System Admin	システム管理者	サービスの挙動を監視する。
IT Auditor	IT監督者	システムや会社全体の権限管理や監査を行う。
Operator	業務担当者	ソフトウェアを利用し業務を行う。
ML Operator	業務担当者(機械学習側)	推論結果に基づき業務を行いつつ、フィードバックを与える。
Model risk Manager	機械学習モデルリスク管理者	機械学習モデルの推論の挙動を監視する。

What: 何を依頼すべきか?

- 情報共有
 - インストール可能なソフトウェアの制限。
 - プロキシなどネットワークアクセスの制限。
 - 環境構築を行う上で制限を解除する必要がある場合の申請方法/申請ルート。
- 依頼
 - 開発端末の調達。
 - (制限の解除)。



コミュニケーションにおいて最も大切なこと

敬意

開発端末のメモリが4Gしかない、バージョン管理がSubversion、docker buildがプロキシでブロックされて失敗する、そんな日々でも be humble。

おわりに

(再掲)本動画のゴール

- Pythonで機械学習モデルを開発する環境を構築できるようになる。
- 開発環境の仕様についてDevOpsチームと合意を取れるようになる。

Let's Challenge!!

資料はすべてオープンな場所で公開しています
動画はYouTube、ハンズオンはGitHubに掲載。

<https://github.com/aws-samples/aws-ml-enablement-handson>

aws-samples / aws-ml-enablement-handson Public

<> Code ① Issues ② Pull requests ③ Discussions Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

icoxfog417 Update README.md 797ac75 5 days ago 3 commits

- .github/ISSUE_TEMPLATE Add hands-on contents 5 days ago
- docs Add hands-on contents 5 days ago
- moder
- notebooks
- .giti

① プログラミング演習でエラーが発生したときはIssuesへ
② Notebookなどの修正はPull requestsへ
③ 内容についての質問や議論はDiscussionsへ

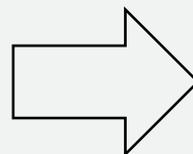
ML Enablement Seriesの動画

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 (本動画シリーズ)

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



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](#)

AWSのハンズオン資料の場所: 「AWS ハンズオン」で検索



Special Thanks

- 音楽素材: [PANICPUMPKIN様](#)

参考文献

1. Anaconda. [Terms of Service](#)
2. Greg Markovich. [Miniconda vs Anaconda | The Similarities and Differences](#). 2020.
3. [What is the difference between pip and conda?](#)
4. [wheelのありがたさとAnacondaへの要望](#)
5. [Misleading description of poetry](#)
6. vaaaaanquish. [pipとpipenvとpoetryの技術的・歴史的背景とその展望](#). 2021.
7. Pythonの教材
 1. 岡崎直観. [Python早見帳](#). 2020.
 2. 三谷純. [Python ゼロからはじめるプログラミング](#). 2021.
 3. 東京大学 数理・情報教育研究センター. [Pythonプログラミング入門](#). 2020.



Thank you!