

みんな大好き！ 機械学習モデルの トレーニング

ML Enablement Series: Training

呉 和仁
機械学習ソリューションアーキテクト



本動画のゴール

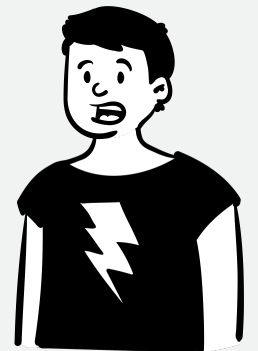
- 機械学習モデルのアルゴリズムの選定とトレーニングができるようになる
- 機械学習モデルのトレーニングの進捗報告を行いフィードバックがもらえるようになる



本動画のゴール

- 機械学習モデルのアルゴリズムの選定とトレーニングができるようになる
- 機械学習モデルのトレーニングの進捗報告を行い
フィードバックがもらえるようになる

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。

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

Plan

Code

Build

Test

Release

Deploy

Operate

Monitor

Code
Repository

Artifact

Verification

Software
1.0

Service

Raw
Data

Business
Understanding

Data
Warehouse

Collect

Data
Repository

Features

Model

Verification

Software
2.0

Service

Raw
Data

Analyze

Prepare

Preprocess

Train

Test

Release

Deploy

Operate

Monitor

MLOps

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

Plan

Code

Build

Test

Release

Deploy

Operate

Monitor

Code
Repository

Artifact

Verification

Software
1.0

Service

Raw
Data

Business
Understanding

Data
Warehouse

Collect

Data
Repository

Features

Model

Verification

Software
2.0

Service

Raw
Data

Analyze

Prepare

Preprocess

Train

Test

Release

Deploy

Operate

Monitor

MLOps

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

Plan

Code

Build

Test

Release

Deploy

Operate

Monitor

Code
Repository

Artifact

Verification

Software
1.0

Service

Raw
Data

Business
Understanding

Data
Warehouse

Collect

Data
Repository

Features

Model

Verification

Software
2.0

Service

Raw
Data

Analyze

Prepare

Preprocess

Train

Test

Release

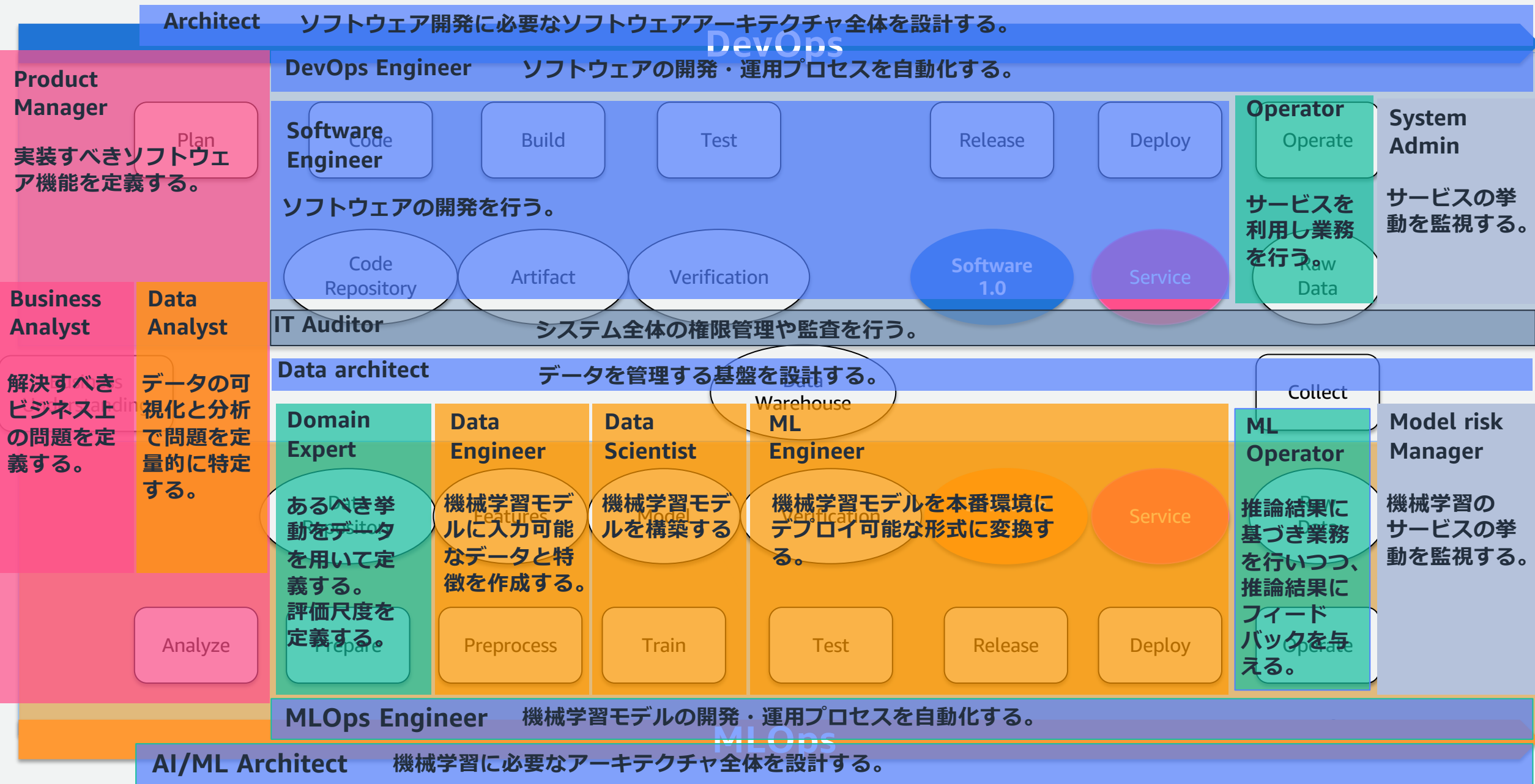
Deploy

Operate

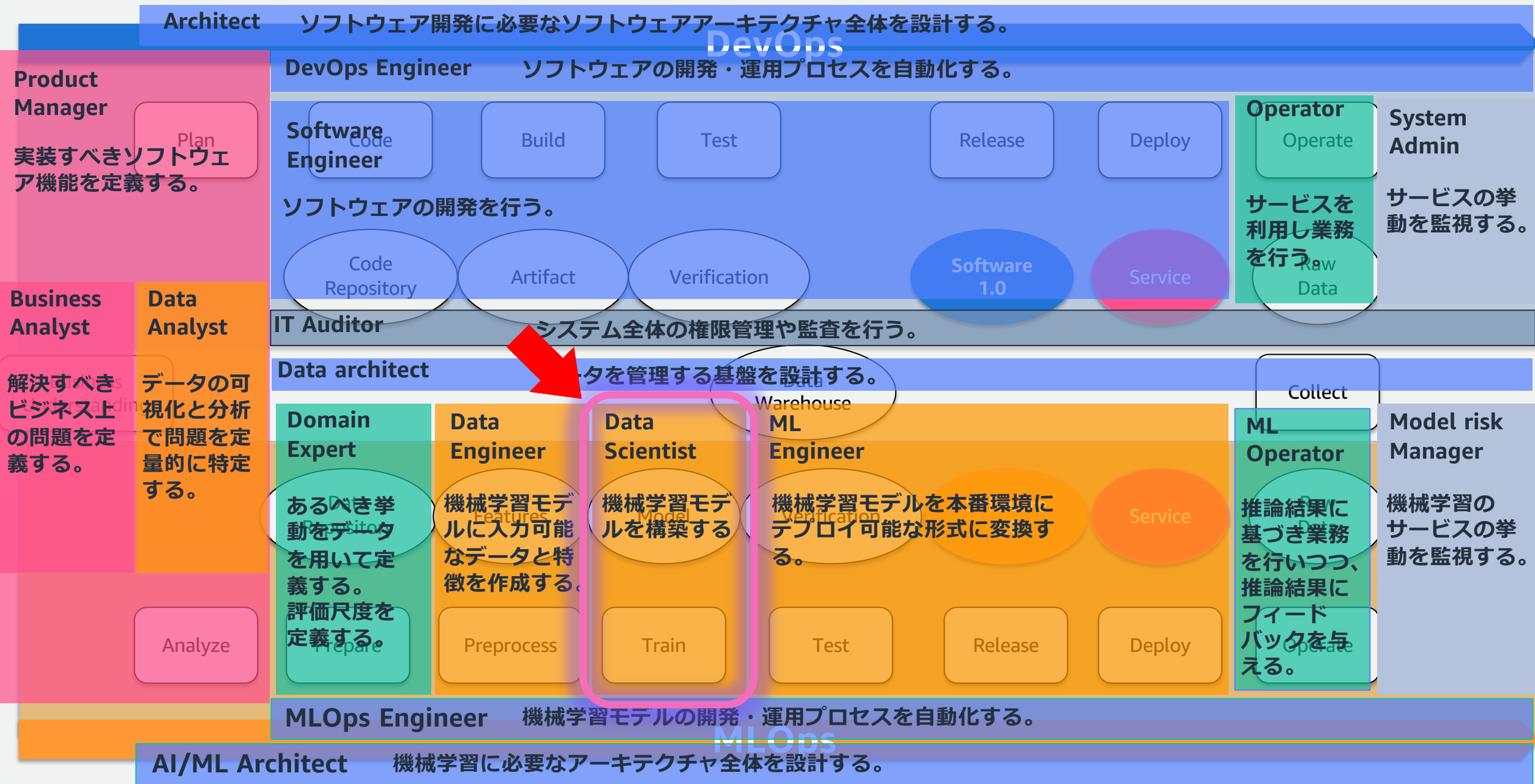
Monitor

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。



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

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?

チームでの認識合わせに役立つ

3.1 プログラミング演習

具体的な開発方法を学ぶのに役立つ

3.2 コミュニケーション演習

会議体などを設計するのに役立つ

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?

チームでの認識合わせに役立つ

3.1 プログラミング演習

具体的な開発方法を学ぶのに役立つ

3.2 コミュニケーション演習

会議体などを設計するのに役立つ

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?

チームでの認識合わせに役立つ

3.1 プログラミング演習

具体的な開発方法を学ぶのに役立つ

3.2 コミュニケーション演習

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

「機械学習モデルのトレーニング」での解説

DevOps

アルゴリズムの選択とモデルをトレーニングする方法について解説

- プログラミング演習

機械学習で解く問題を理解して適切なアルゴリズムを選択し、
モデルのトレーニングができるようになる

- コミュニケーション演習

出来上がったモデルについてフィードバックを得る方法を解説

Understanding

Warehouse

Collect

Data
Repository

Features

Model

Verification

Software
2.0

Service

Raw
Data

Analyze

Prepare

Preprocess

Train

Test

Release

Deploy

Operate

Monitor

MLOps

「機械学習モデルのトレーニング」での解説

DevOps

アルゴリズムの選択とモデルをトレーニングする方法について解説

- プログラミング演習

機械学習で解く問題を理解して適切なアルゴリズムを選択し、
モデルのトレーニングができるようになる

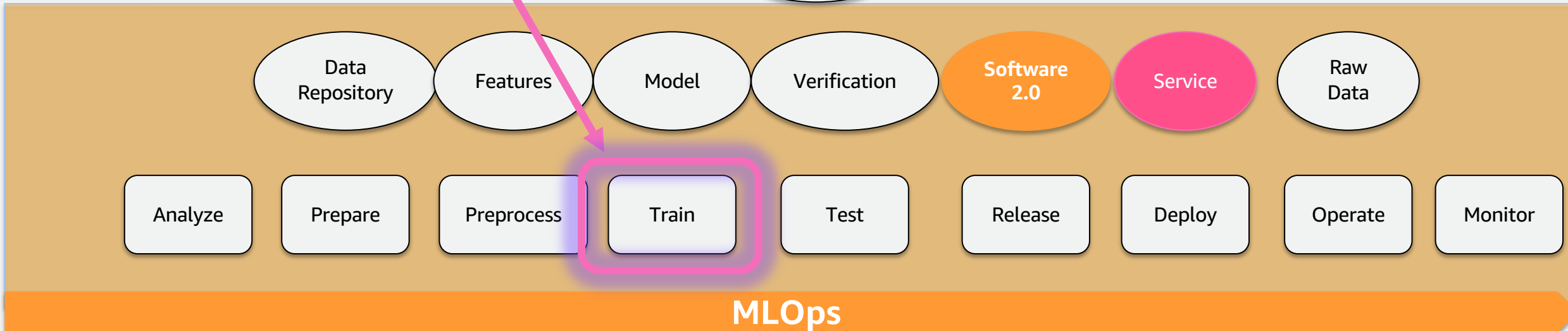
- コミュニケーション演習

出来上がったモデルについてフィードバックを得る方法を解説

Understanding

Warehouse

Collect



「機械学習モデルのトレーニング」での解説

DevOps

アルゴリズムの選択とモデルをトレーニングする方法について解説

- プログラミング演習

機械学習で解く問題を理解して適切なアルゴリズムを選択し、
モデルのトレーニングができるようになる

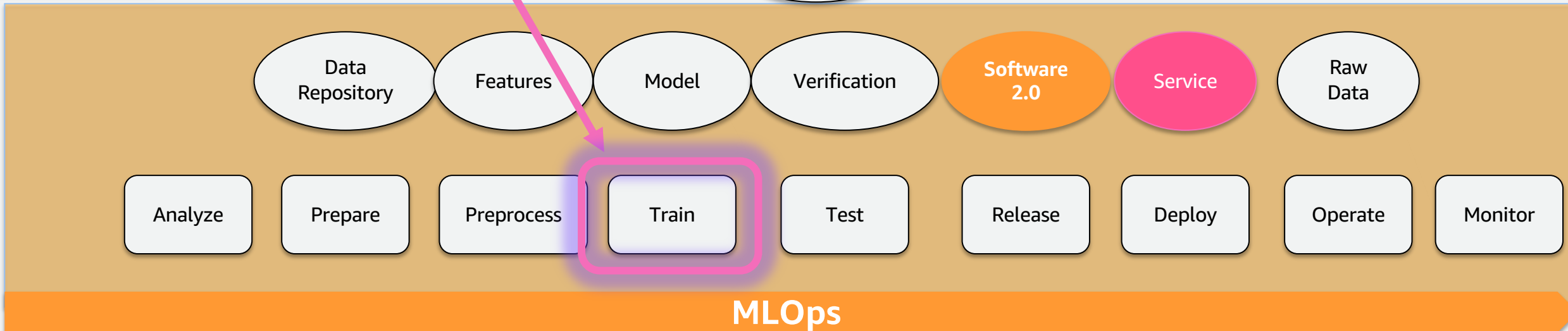
- コミュニケーション演習

出来上がったモデルについてフィードバックを得る方法を解説

Understanding

Warehouse

Collect



「機械学習モデルのトレーニング」での解説

DevOps

アルゴリズムの選択とモデルをトレーニングする方法について解説

- プログラミング演習

機械学習で解く問題を理解して適切なアルゴリズムを選択し、
モデルをトレーニングができるようになる

- コミュニケーション演習

出来上がったモデルについてフィードバックを得る方法を解説

Understanding

Warehouse

Collect

Data
Repository

Features

Model

Verification

Software
2.0

Service

Raw
Data

Analyze

Prepare

Preprocess

Train

Test

Release

Deploy

Operate

Monitor

MLOps

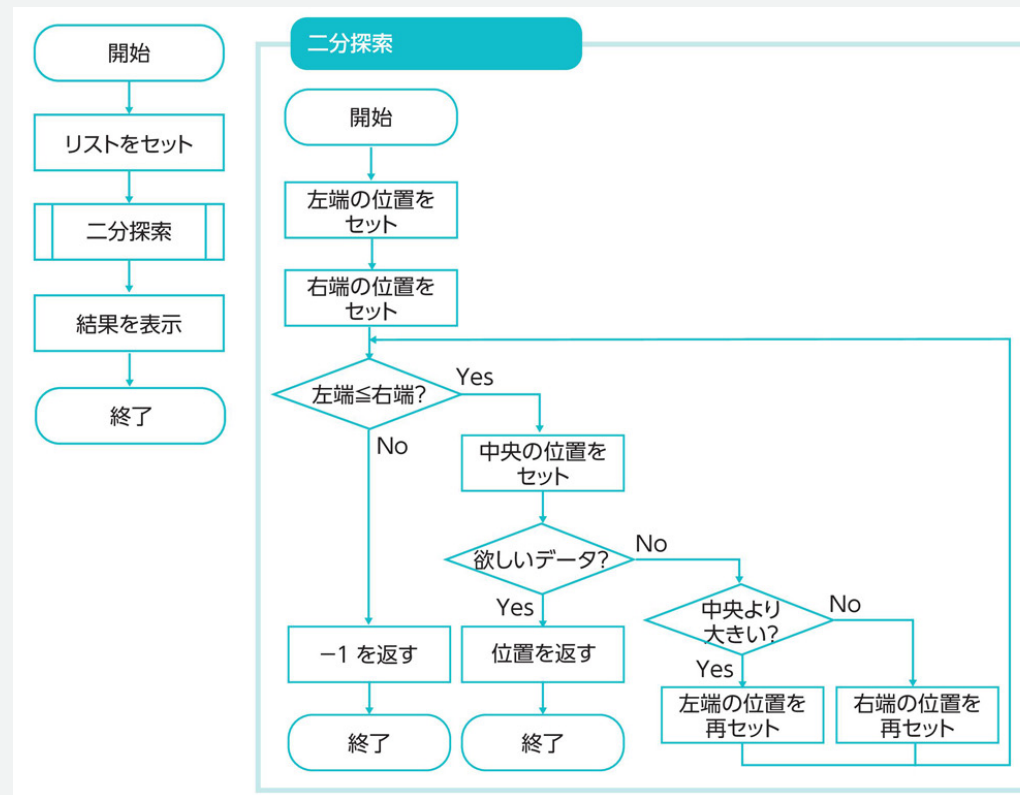
What: なにをするのか?

機械学習モデルのトレーニングのゴール

- 使用する機械学習のアルゴリズムを選定する。
- Preprocess で用意したデータでトレーニングを行う

アルゴリズムってなに？

- 問題を解くための計算方法
- 機械学習の文脈に限ればモデルの種類



例) 二分探索アルゴリズム

機械学習のアルゴリズムは数多あり選択に困る

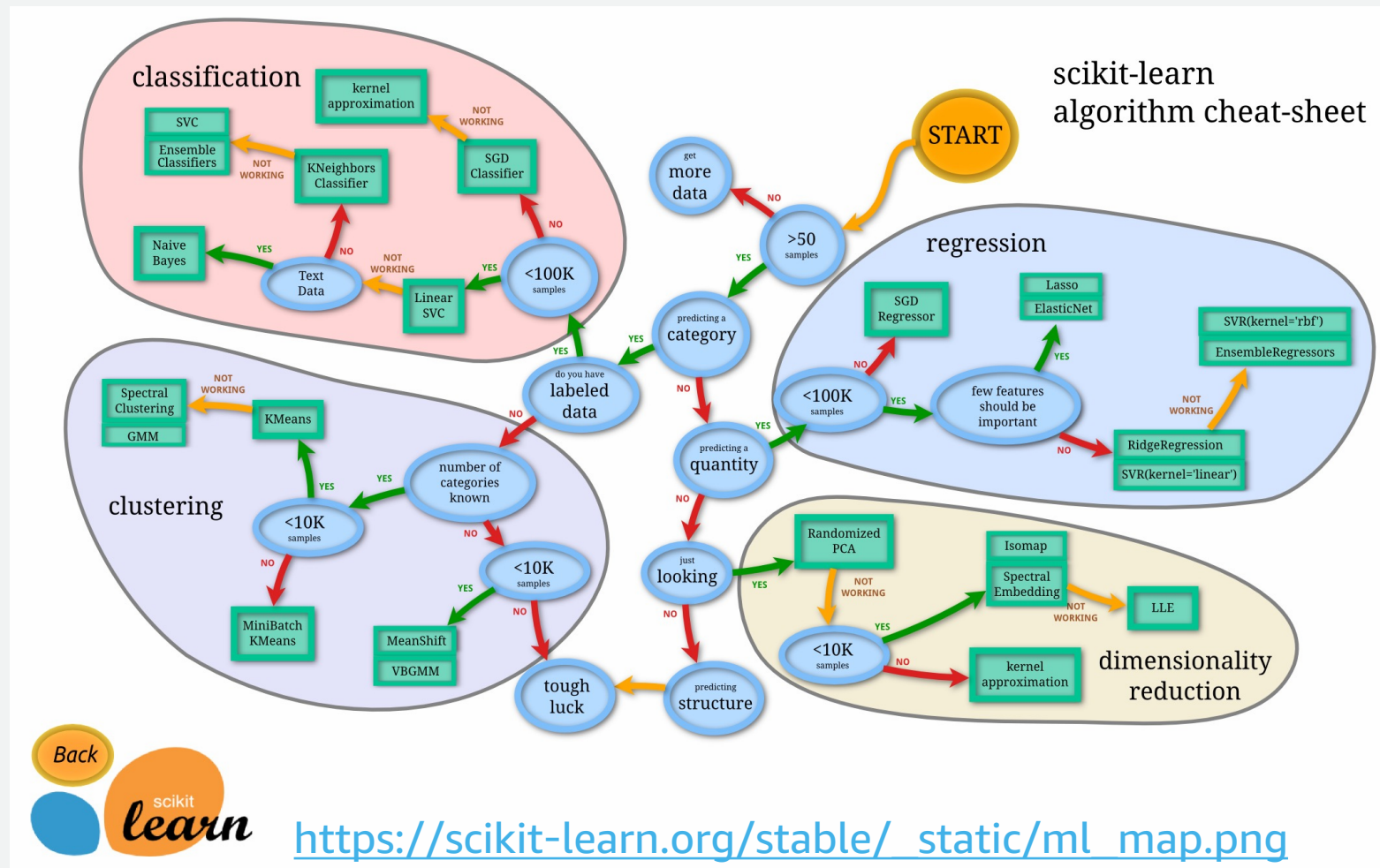
機械学習のアルゴリズムの例

- 教師あり学習
 - 線形回帰
 - 正則化
 - ロジスティック回帰
 - サポートベクトルマシン
 - ナイーブベイズ
 - ランダムフォレスト
 - ニューラルネットワーク
 - kNN
- 教師なし学習
 - PCA
 - LSA
 - NMF
 - LDA
 - k-means 法
 - 混合ガウス分布
 - LLE
 - t-SNE

出典：["見て試してわかる機械学習アルゴリズムの仕組み 機械学習図鑑"](#)の目次より引用

使用する機械学習のアルゴリズム選定のガイド

参考になるものはあるが・・・その前に！



アルゴリズム選定前に抑えるべきこと

前提条件を確認しておくと手戻りが少ない

- 解く問題の大別
- 要求する速度パフォーマンス
- 推論環境
- 複雑さに対する許容範囲
- 精度に関わる要件

解く問題の大別

まずは一般論を抑えよう (=例外はある)

部品の残存寿命を知りたい

商品の需要を予測したい

商品のレコメンドをしたい

融資可否の審査をしたい

製品/設備に異常があるか判定したい

機械的に顧客をグルーピングしたい

情報が多すぎるので削減したい

教師あり学習

• 回帰

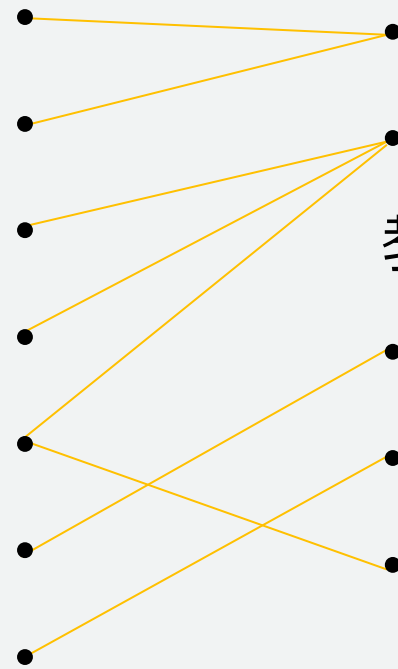
• 分類

教師なし学習

• クラスタリング

• 次元圧縮

• その他



要求する速度パフォーマンス

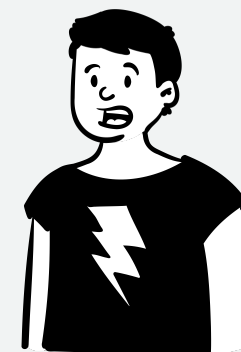
業務を考慮してレイテンシーやスループットに関するシステム要求を理解

小売における需要予測から発注の場合

前日の 24 時に揃ったデータを使い、当日の午後の需要を予測して9時に発注するので、余裕を見て

24時にスタートして6時に予測完了すれば OK です。

10000 アイテムの需要予測をする必要があります。



生産ラインにおける部品の異常検出の場合

部品が出来上がって次の工程に最短で10分で移るので、撮影してから異常を検出してアラートを出して止めるので、余裕を見てデータが出来てから5分で終えたいです。

製品は1分あたり5個ラインを流れて来ます。



推論環境も考慮に入れよう

クラウド/エッジ, 並列, CPU/GPU などを検討して最適な推論環境を検討する



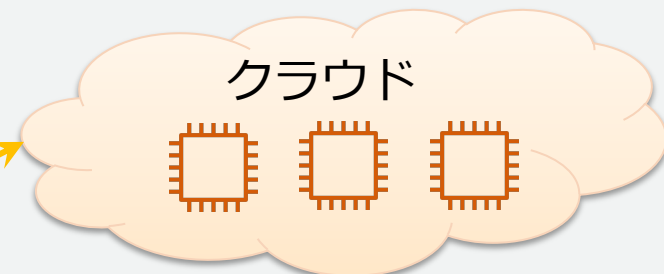
データが出来て 5 分で終わる必要があります。
製品は 1 分あたり 50 個ラインを流れて来ます。

データの転送時間を
考慮する必要がある



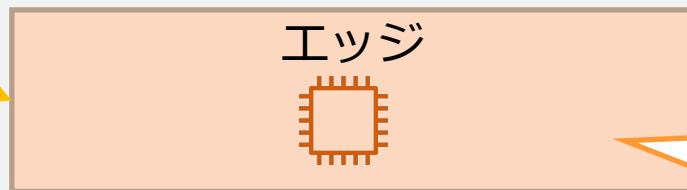
推論画像

同一デバイスで撮影して
推論できたら転送時間は
無視できる



コンピューティング
リソースを多数用意して
並列処理しやすく GPU も
容易に使える

推論は
クラウドとエッジ
どっち?

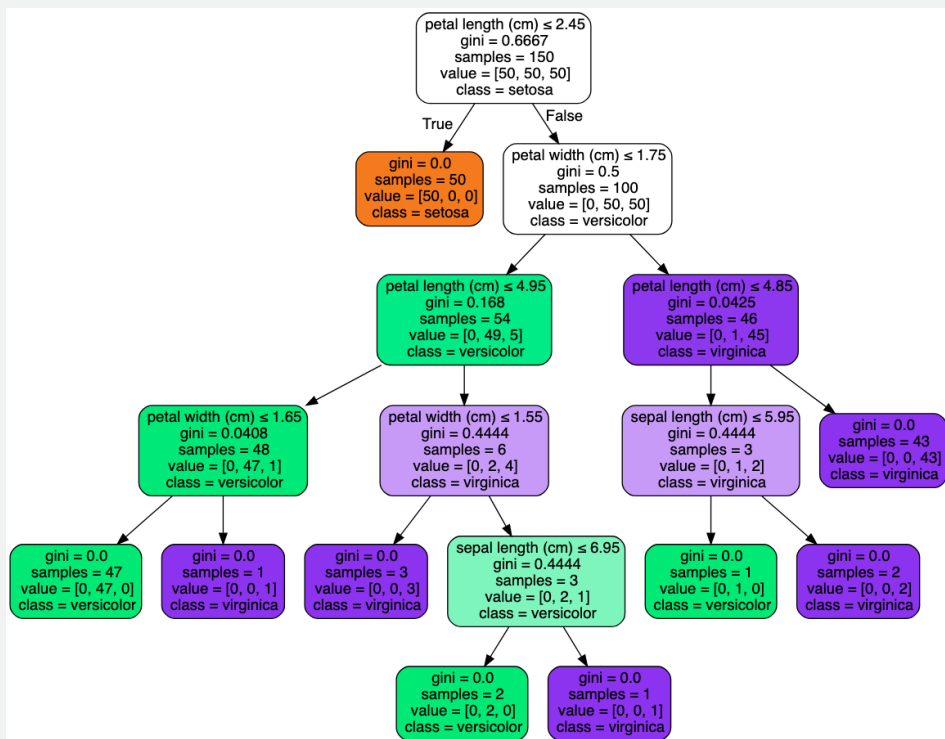


異常発生時は
現場へアラート

エッジ側の複数のリソース
準備はクラウドに比べると
大変

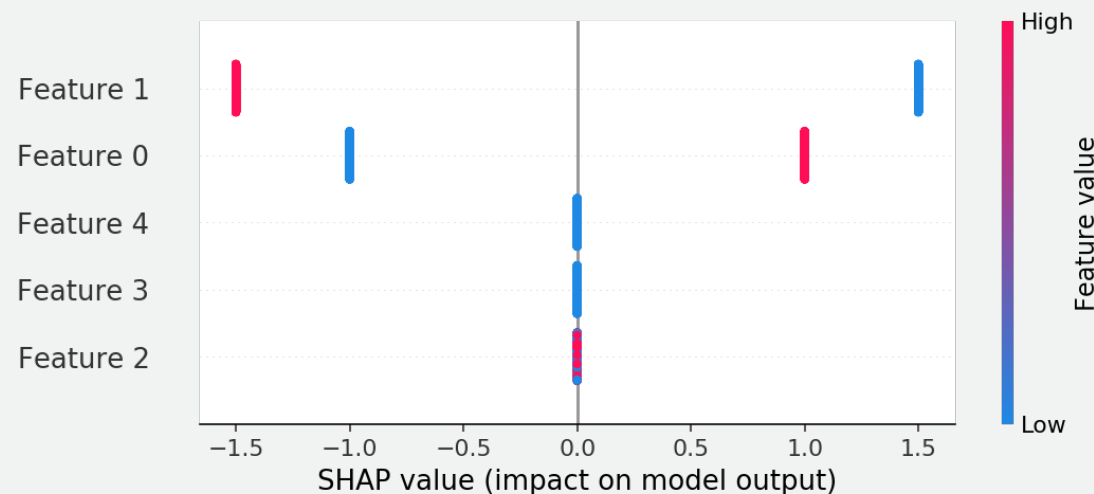
モデルの複雑さに対する許容

モデルが複雑になればなるほど中身の説明は難しくなるので、どこまでの説明が要求されるのかを抑えておく



決定木の可視化例:

<https://scikit-learn.org/stable/images/iris.svg>



XGBoost における SHAP 値の可視化例 :

https://shap.readthedocs.io/en/latest/images/example_not_ebooks_tabular_examples_tree_based_models_Basic_SHAP_Interaction_Value_Example_in_XGBoost_9_0.png

精度に関わる要件

Business Understandingで達成したい目標値に照らして、要求される精度をチェック
詳しくは [ML-Light-03](#) を復習！

事例1: 飲料メーカー A 社での機械学習プロジェクト

戦略:

漏れ・重複がない観点から価値の計算を行う(例: QCD)

1. (Quality: 品質) 製造後の工程(出荷や販売)におけるクレームの数

- 機械学習の Q の価値
= (Before のクレーム数 - After のクレーム数) x 損失額

2. (Cost: コスト) 検品コスト

- 機械学習の C の価値
= (Before の検品人数 - After の検品人数) x 時間 x 時給

3. (Delivery: 納期) 検品時間

- 機械学習の D の価値
= Before の時間 - After の時間



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

事例2: 道路のひび割れを検出する機械学習プロジェクト

戦略:

どの程度の検出数であれば価値を達成できるか、人間の能力値の上限と下限を目安に機械学習の達成目標を設定する。

1. 導入基準: 「そこそこの人員の代替」

- 素人の検出数 < モデルの検出数 < 玄人の検出数**

2. 検出コスト

具体的な目標値は、決めておきましょう!

- 機械学習の C の価値
= (Before担当者数 - After担当者数) x 時間 x 時給



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

72

さっそくトレーニング！…の前にデータを読み込む

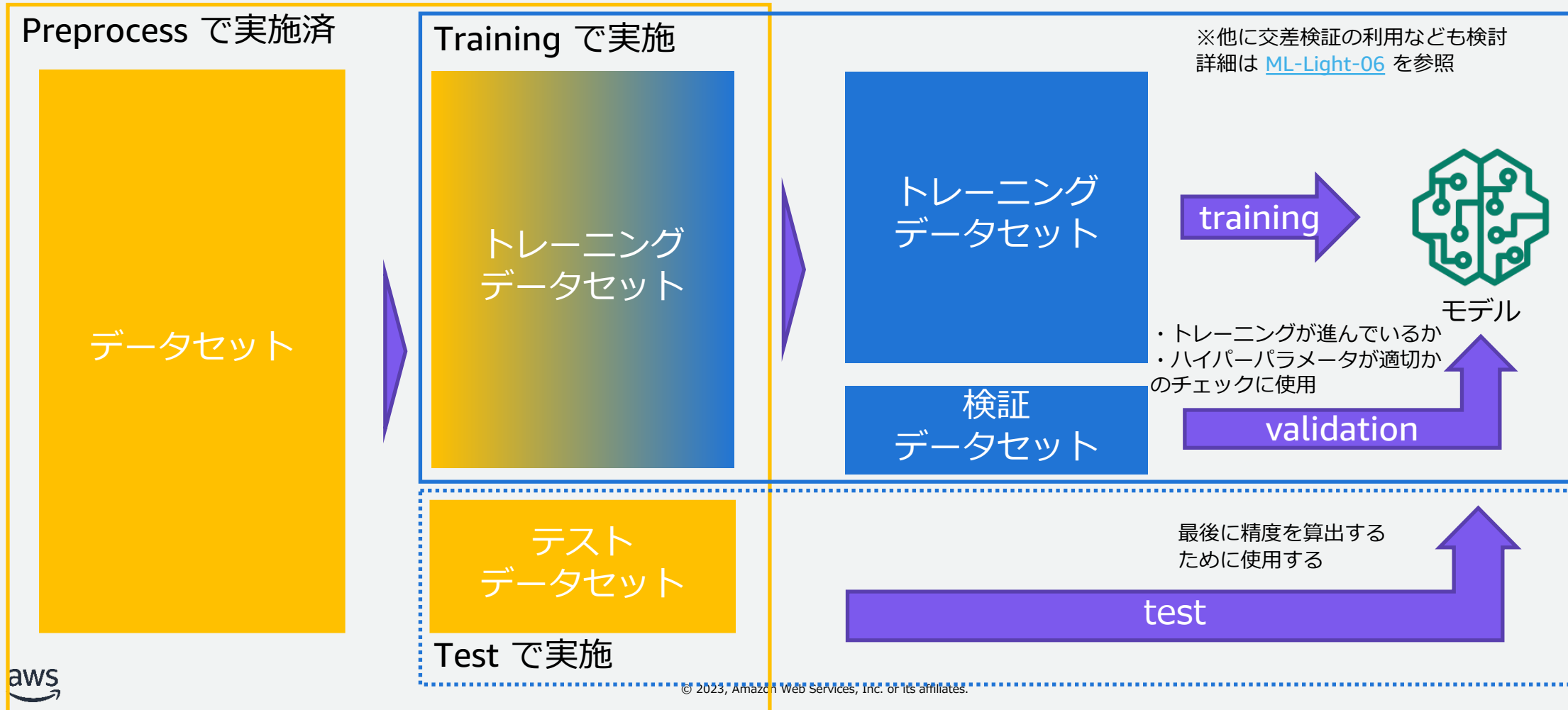
トレーニングするために Preprocess で用意したデータをロード
アルゴリズム/ライブラリによって使えるフォーマットは異なる

- 入力フォーマット例
csv, tsv, parquet, Hierarchical Data Format, npy/npz, etc...
- 読み込みフォーマット例
pandas DataFrame, ndarray, TFRecord, etc...

例) csv ファイルを pandas DataFrame で読む場合
df = pd.read_csv("hoge.csv")

データセットを分割

ハイパーパラメータを調整する際、どのような値が良いのか探索するため、トレーニングを回して精度を確認するための検証データセットを用意



ハイパーパラメータの決定

トレーニング時に人が設定する必要があるパラメータの決定

Parameters:	criterion : {"gini", "entropy", "log_loss"}, default="gini" The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation .
	splitter : {"best", "random"}, default="best" The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.
	max_depth : int, default=None The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
	min_samples_split : int or float, default=2 The minimum number of samples required to split an internal node: <ul style="list-style-type: none">• If int, then consider min_samples_split as the minimum number.• If float, then min_samples_split is a fraction and $\text{ceil}(\text{min_samples_split} * \text{n_samples})$ are the minimum number of samples for each split. <p><i>Changed in version 0.18: Added float values for fractions.</i></p>
	min_samples_leaf : int or float, default=1 The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression. <ul style="list-style-type: none">• If int, then consider min_samples_leaf as the minimum number.• If float, then min_samples_leaf is a fraction and $\text{ceil}(\text{min_samples_leaf} * \text{n_samples})$ are the minimum number of samples for each node. <p><i>Changed in version 0.18: Added float values for fractions.</i></p>

決定木のハイパーパラメータ

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier> より引用

トレーニング実行

scikit-learn や Keras などのハイレベル API を提供しているフレームワークであれば、ハイパーパラメータを指定したモデルインスタンスを設定して、トレーニング API をコールする。

XGBRegressor の場合のモデルインスタンス生成

```
xgb_regressor = xgbost.XGBRegressor(*, objective='reg:squarederror', **kwargs)
```

ハイパーパラメータの設定

(必須) トレーニングデータセットの指定

評価データセットの指定

Training

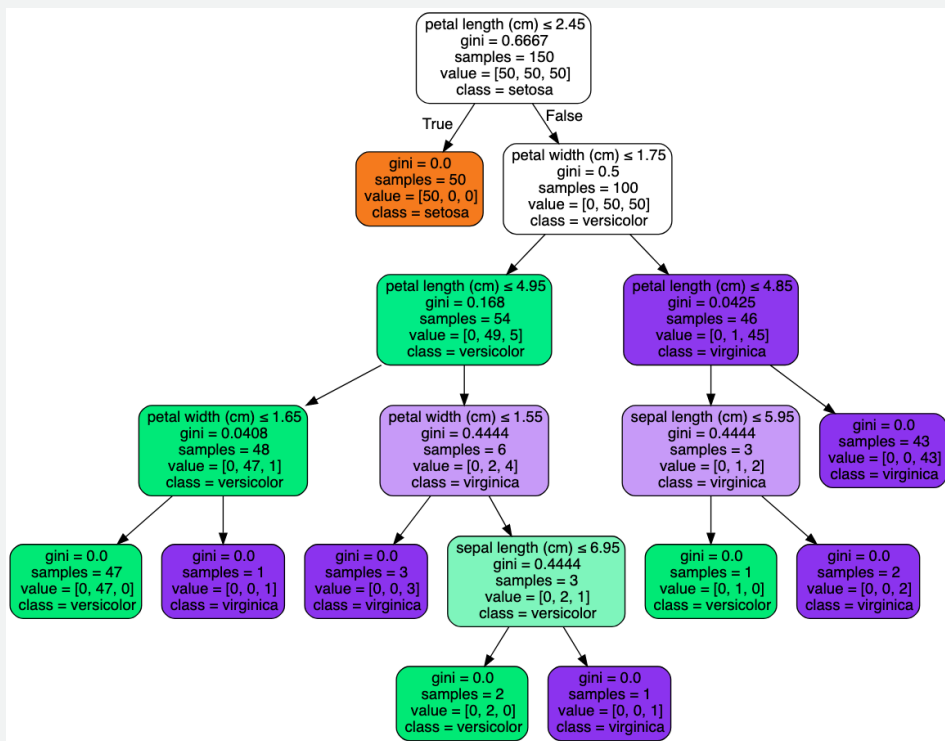
```
xgb_regressor.fit(X, y, *, sample_weight=None, base_margin=None, eval_set=None, eval_metric=None, early_stopping_rounds=None, verbose=True, xgb_model=None, sample_weight_eval_set=None, base_margin_eval_set=None, feature_weights=None, callbacks=None)
```

https://xgboost.readthedocs.io/en/stable/python/python_api.html

モデルの中身を覗いて妥当性を確認

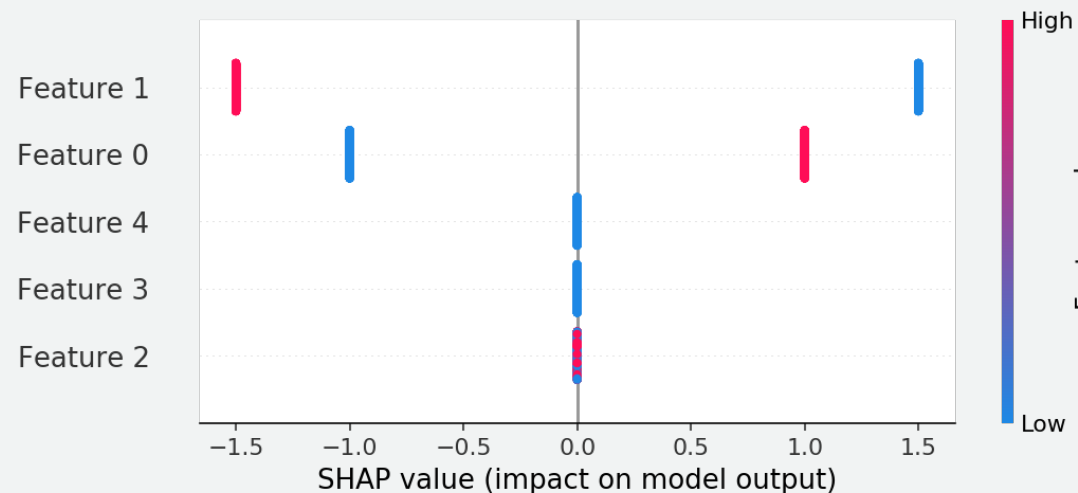
例えば各特徴量の影響力の大小からモデルの妥当性を判断

(「モデルの複雑さに対する許容」より再掲)



決定木の可視化例:

<https://scikit-learn.org/stable/images/iris.svg>



XGBoost における SHAP 値の可視化例:

https://shap.readthedocs.io/en/latest/images/example_not_ebooks_tabular_examples_tree_based_models_Basic_SHAP_Interaction_Value_Example_in_XGBoost_9_0.png

モデルとトレーニングのメタデータを保存しよう

トレーニング後のモデルは一般的にメモリに格納されるので揮発する。
忘れずにストレージに保存しよう。



精度を出すためにアルゴリズムやハイパーパラメータなどを変えて何度もトレーニングを行う。
同じことを何度もやったりしないように、また、誰でも再現できるように記録する。

(**実験管理**と言う→詳しくは [ML-Dark-02](#) を参照)

記録すべきトレーニングのメタデータ例

- トレーニングコード (やコードの commit ID)
- 使用したライブラリ (や使用したコンテナや仮想環境)
- ハイパーパラメータ
- トレーニング結果のメトリクス
- 乱数のシード
- 使用したデータ

など

できたモデルで一度は推論して推論コードを保存しよう

Test でモデルの評価を行うが、後工程のために準備

```
model = load(Path("../model/model.joblib"))  
other_model = load(Path("../model/other_model.joblib"))
```

完成したモデルのロード方法は
モデルを作った人にしかわからない

```
# 特徴から予測を実施  
prediction = model.predict(X)
```

ロードしたモデルにインプットすべきデータの内容も
モデルを作った人にしかわからない

https://github.com/aws-samples/aws-ml-enablement-workshop/blob/main/notebooks/06_test.ipynb より引用

Why: なぜするのか?

機械学習はトレーニングが必須

アルゴリズムを実装すれば動くものではなくトレーニングが必須

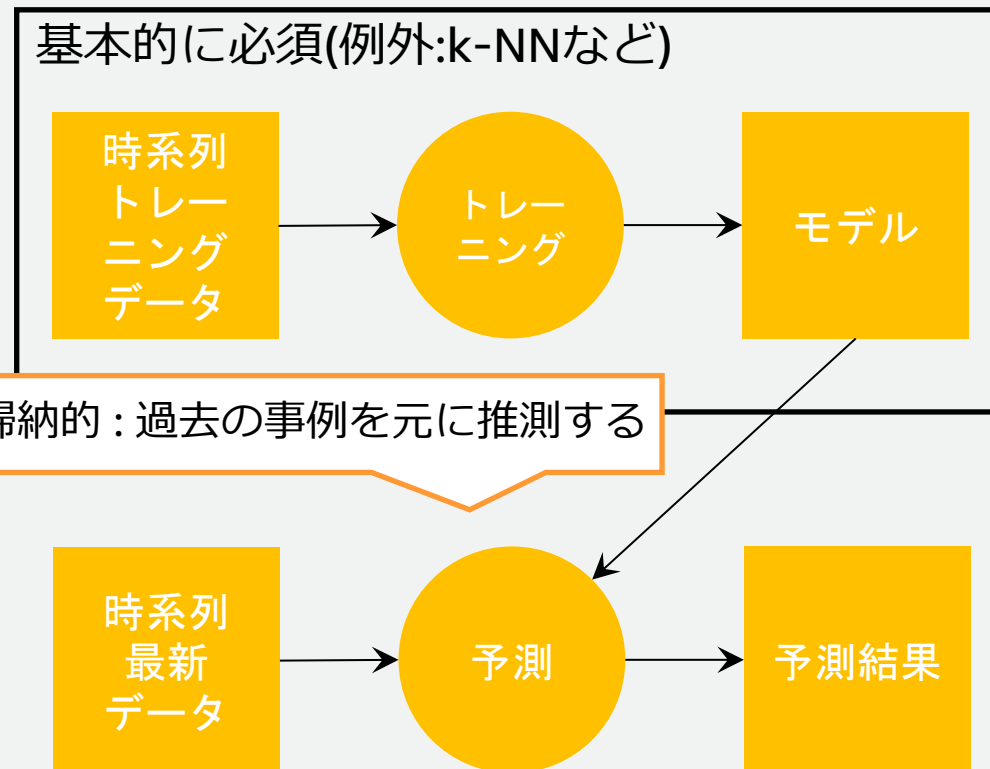
一般的なアルゴリズム(二分探索の場合)



演繹的：一般化された事柄(※)から結果を導き出す

※二分探索の場合、ある値の位置を知る際、ソート済の配列を二分分割して存在する側を再帰的に探索することで、 $O(\log n)$ で探索できる

機械学習の場合(時系列予測の場合)



How: どうやるのか?

演習の概要

3.1 プログラミング演習

- アルゴリズムを選択してモデルをトレーニングする

3.2 コミュニケーション演習

- 選択したアルゴリズムやトレーニング結果の情報共有やレビューをしてもらう

演習の概要

3.1 プログラミング演習

- アルゴリズムを選択してモデルをトレーニングする

3.2 コミュニケーション演習

- 選択したアルゴリズムやトレーニング結果の情報共有やレビューをしてもらう

演習の概要

3.1 プログラミング演習

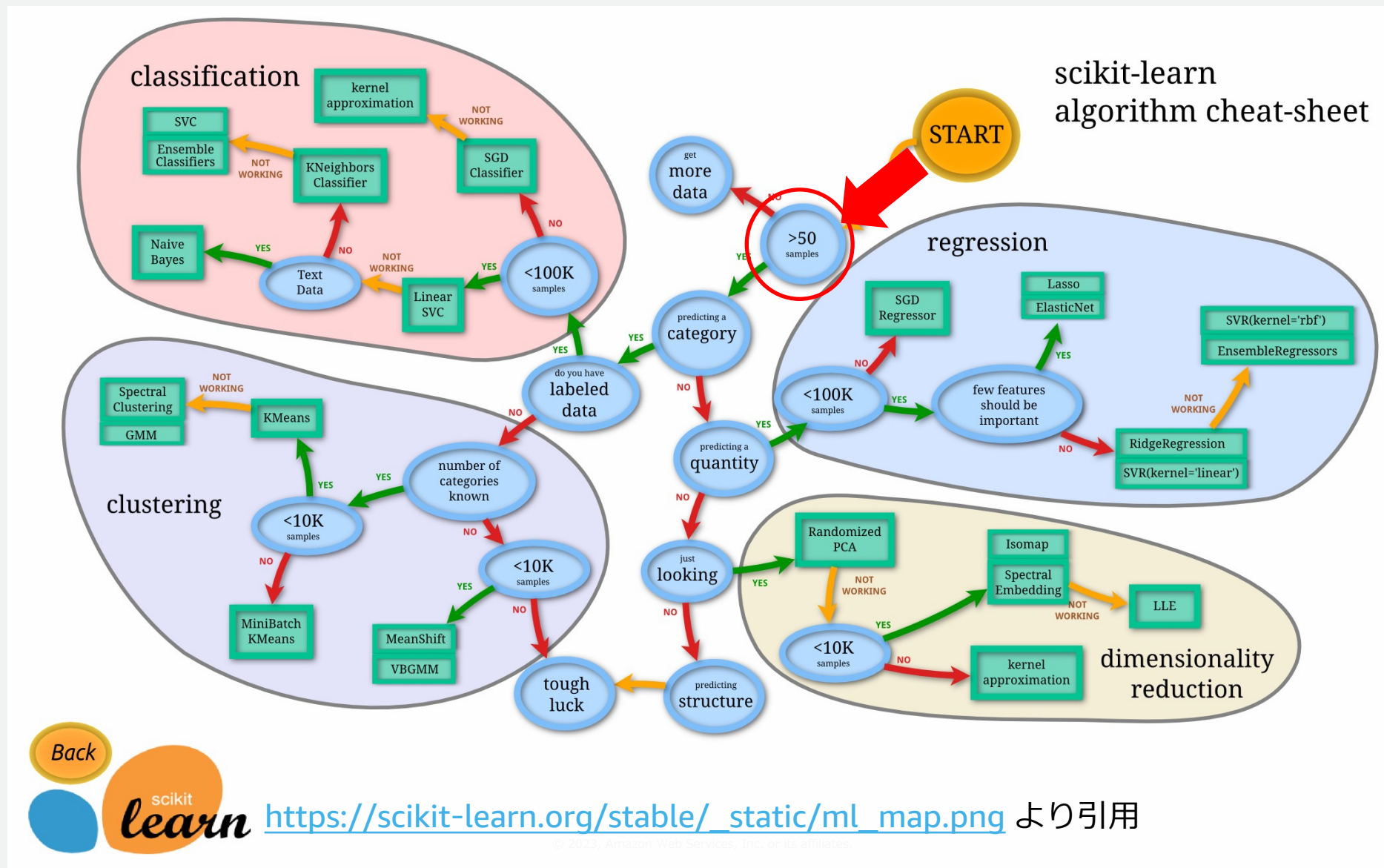
- アルゴリズムを選択してモデルをトレーニングする

3.2 コミュニケーション演習

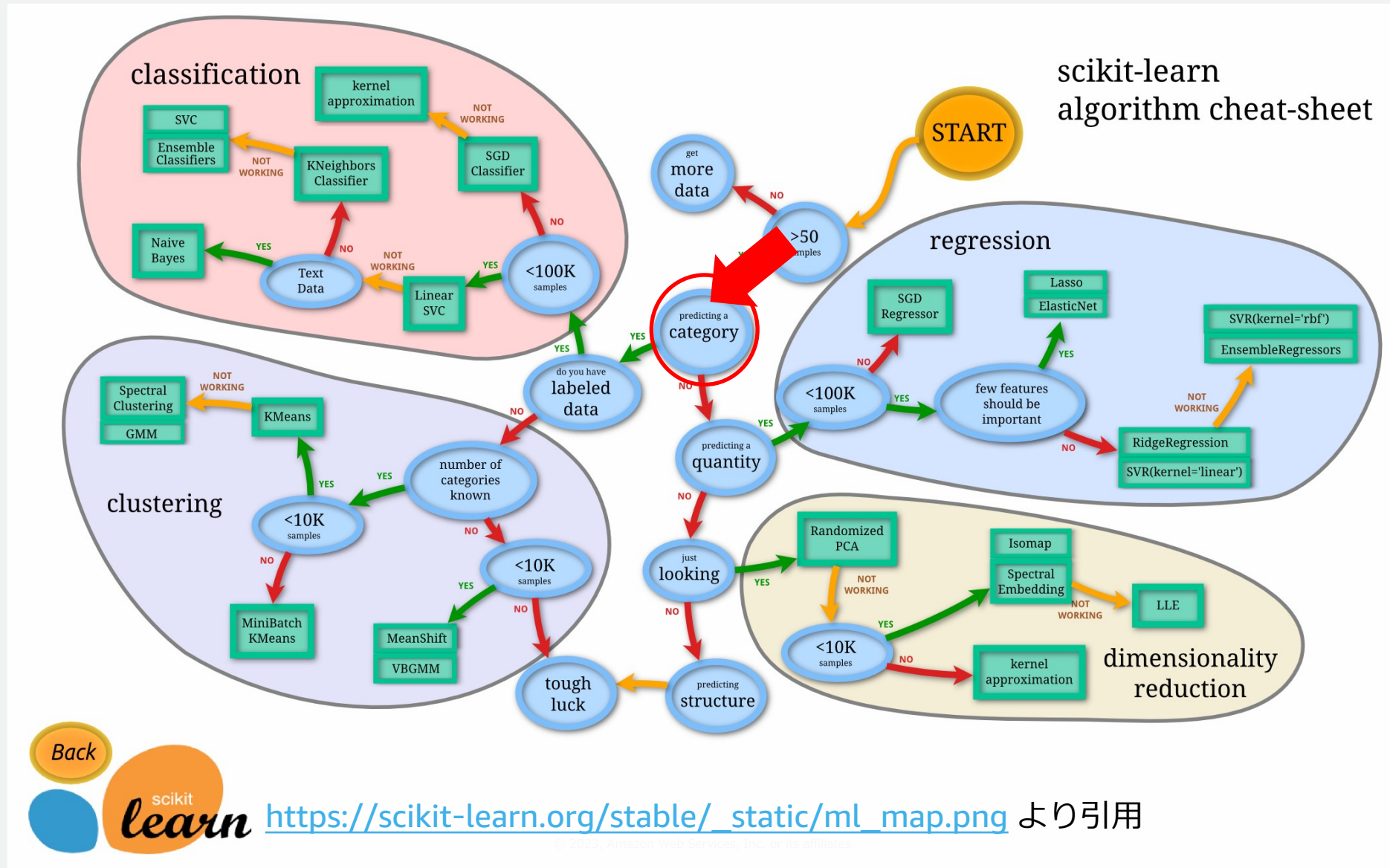
- 選択したアルゴリズムやトレーニング結果の情報共有やレビューをしてもらう

3.1 プログラミング演習

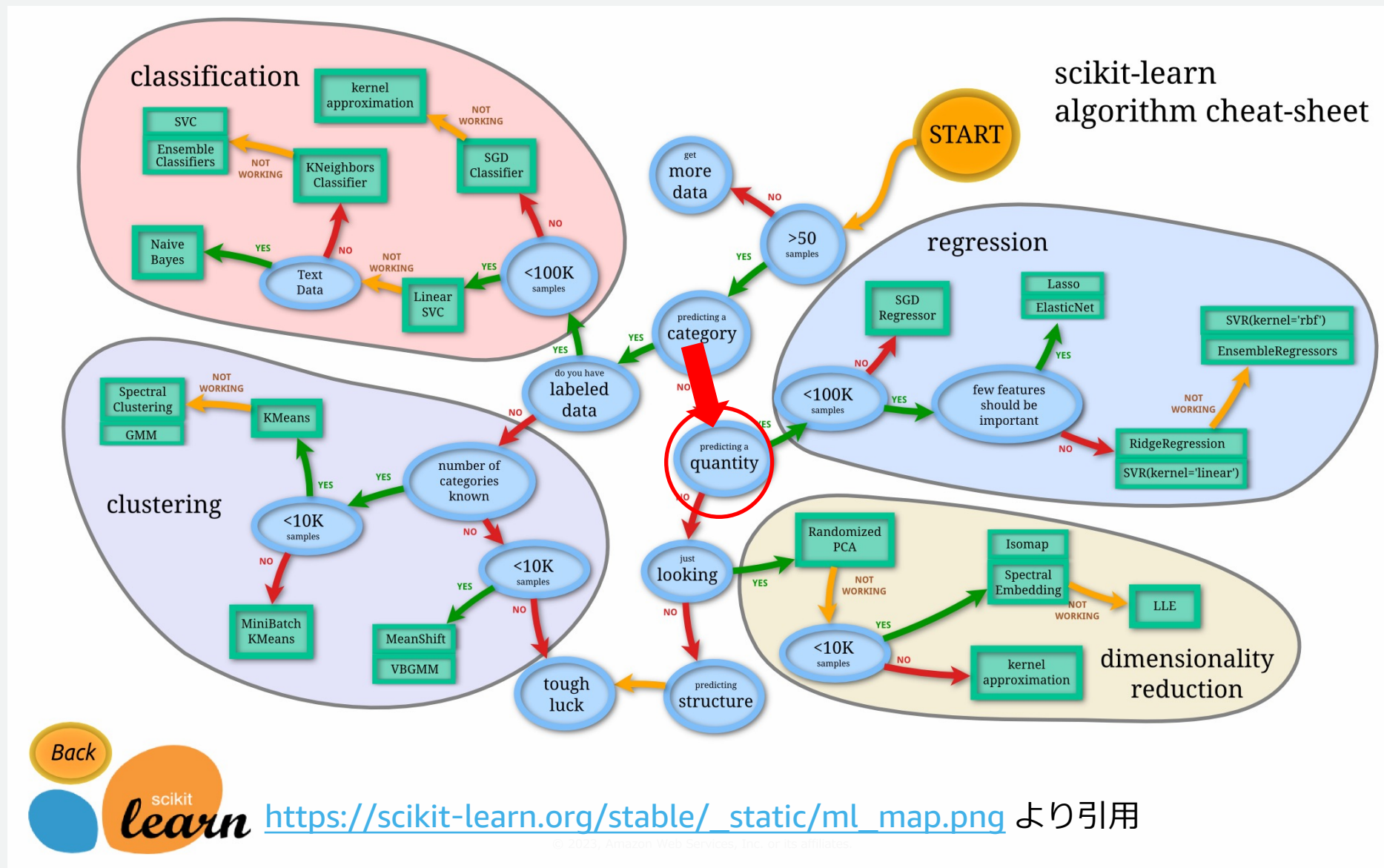
(再掲) チートシート



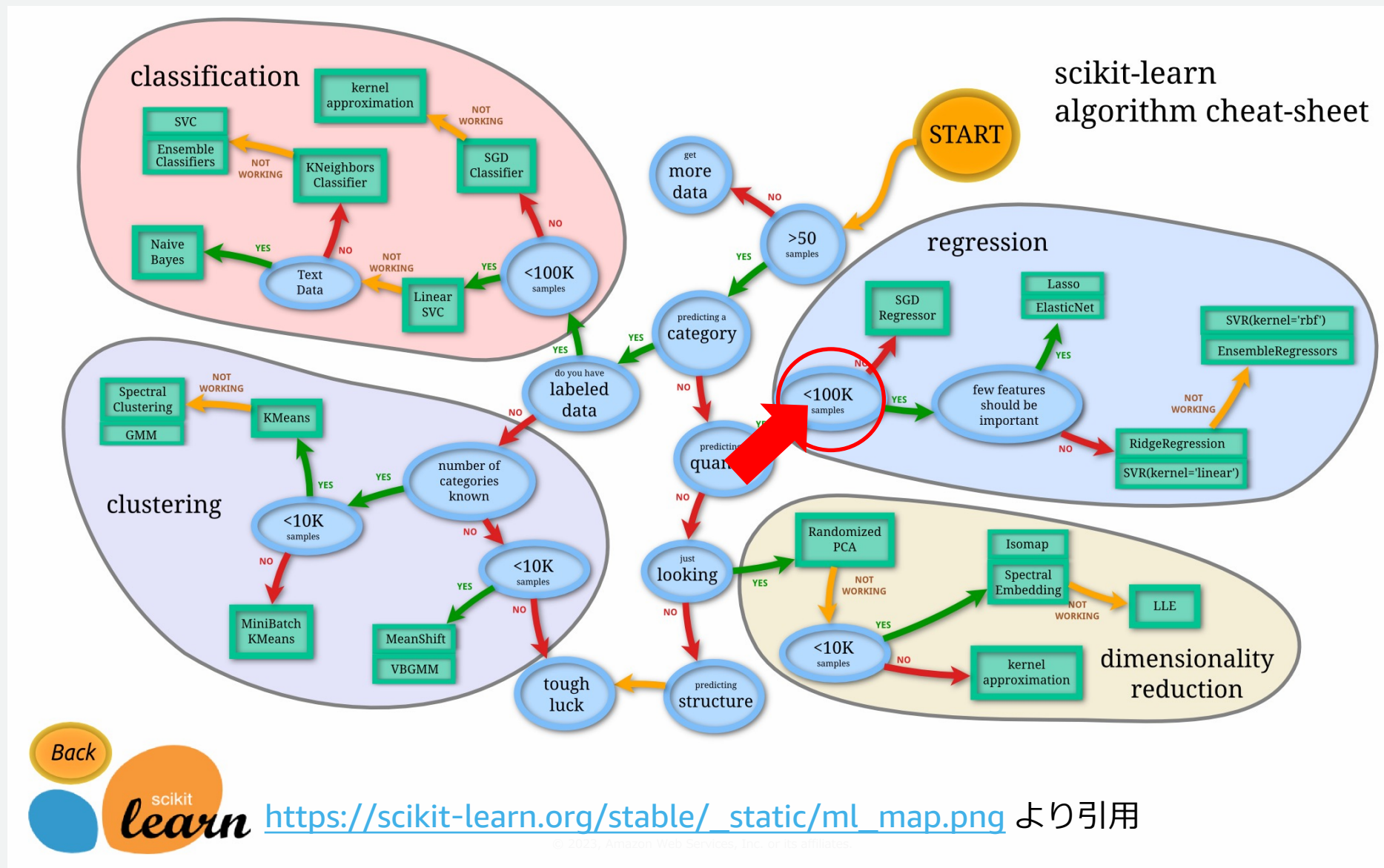
(再掲) チートシート



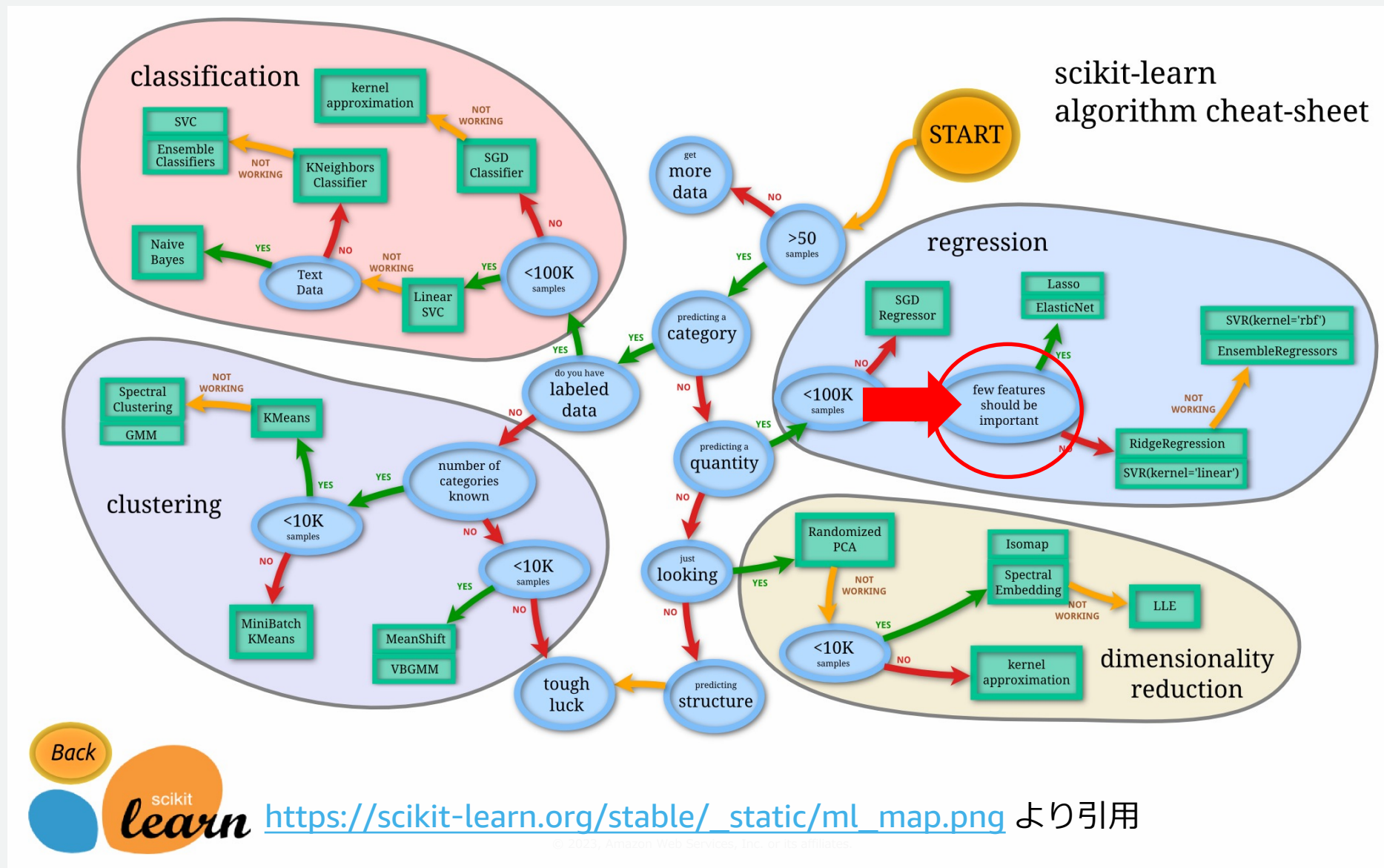
(再掲) チートシート



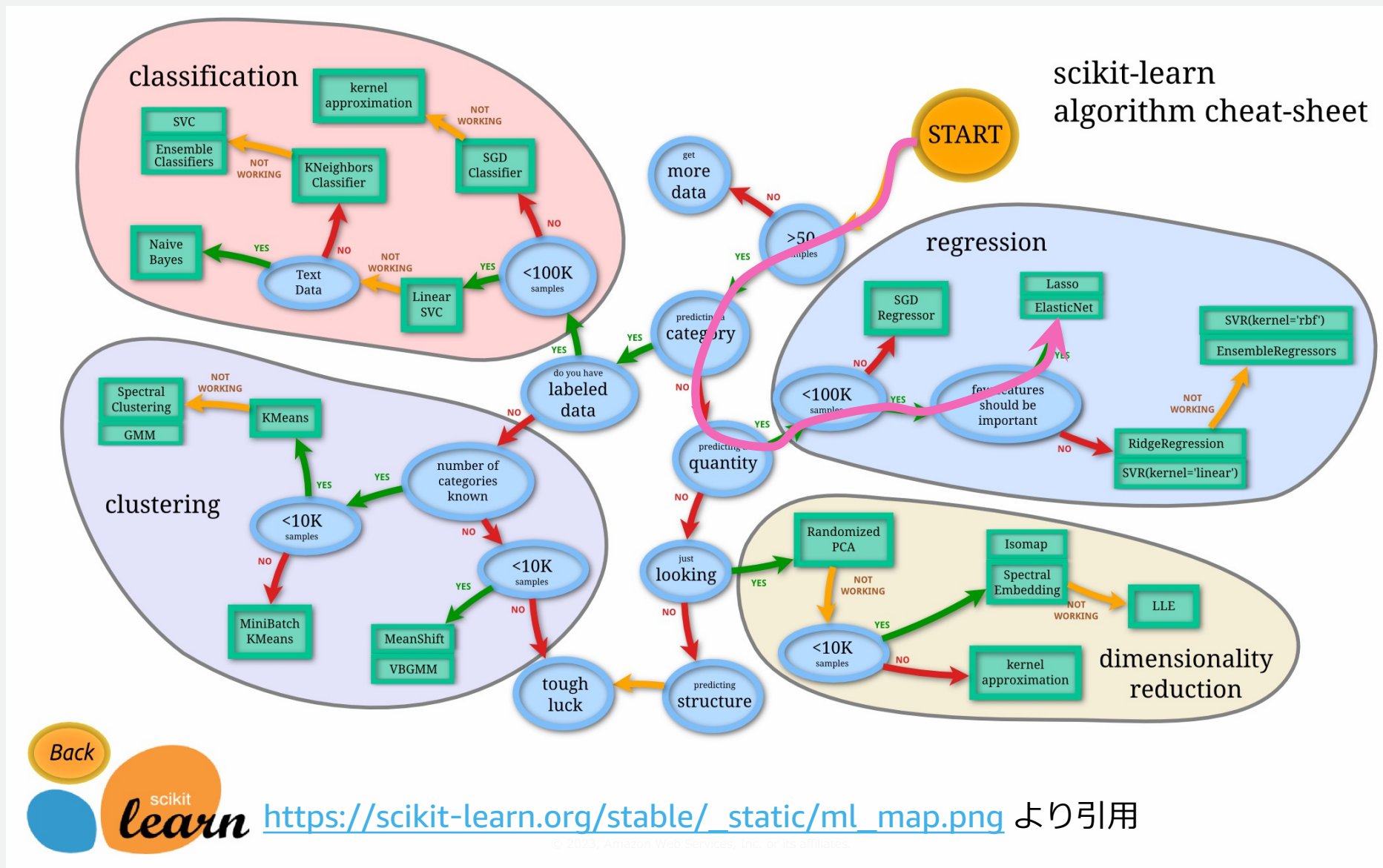
(再掲) チートシート



(再掲) チートシート

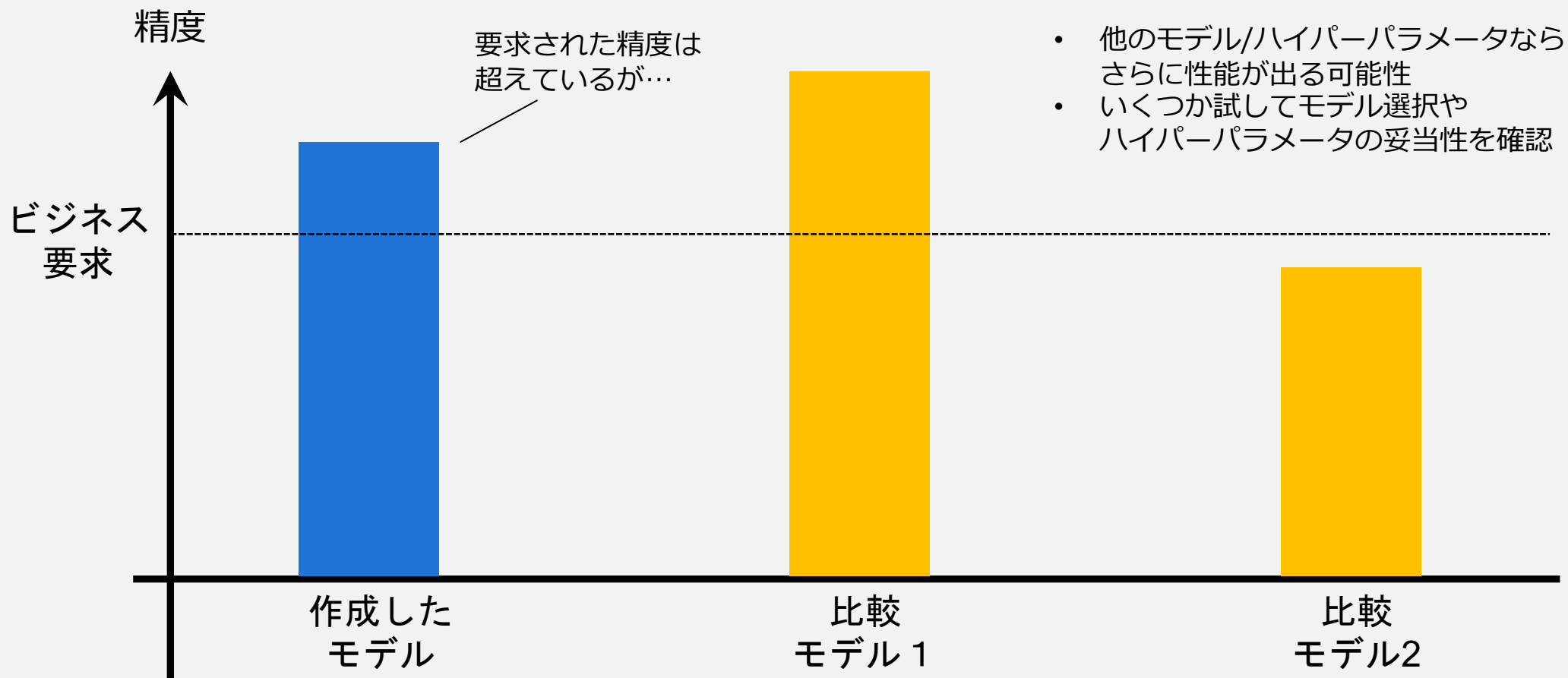


(再掲) チートシート



別のモデルと比較しよう

ビジネスが要求する精度は超えても、妥当な精度か確認をしたほうがよい



※プロダクトを出すことのほうが大事なので、試しすぎによる時間の浪費に注意。時間で打ち切るなど工夫しよう。

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?



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。

Software Engineer

Build

Test

Release

Deploy

ソフトウェアの開発を行う。

Code Repository

Artifact

Verification

Software 1.0

Service

Operator
Operate

System Admin

サービスを利用し業務を行う。
Raw Data

サービスの挙動を監視する。

IT Auditor

システム全体の権限管理や監査を行う。

Data architect

データを扱うための基盤を設計する。

Domain Expert

Data Engineer

Data Scientist

Warehouse ML Engineer

ML Operator

Model risk Manager

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。

機械学習モデルに入力可能なデータと特徴を作成する。

機械学習モデルを構築する。

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。

推論結果に基づき業務を行い、推論結果にフィードバックを与える。

機械学習のサービスの挙動を監視する。

Preprocess

Train

Test

Release

Deploy

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。

Product Manager

実装すべきソフトウェア機能を定義する。

Plan

Business Analyst

Data Analyst

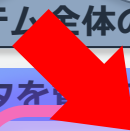
解決すべきビジネス上の問題を定義する。

データの可視化と分析で問題を定量的に特定する。

Analyze

DevOps

MLOps



トレーニングで行ったこと

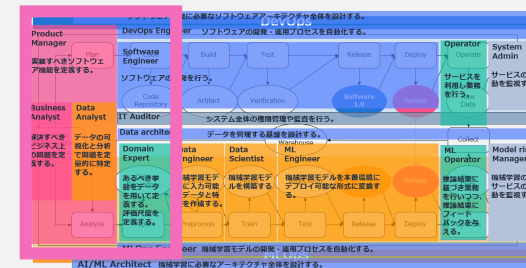
1. 前提条件からトレーニングのアルゴリズムを選定
2. トレーニング実行
3. モデルの保存
4. 推論コードの作成

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

Who: ビジネス要求に詳しい人



英語名	日本語名	役割
Product Manager	プロダクトマネージャー	ユーザー体験を向上させるために実装すべきソフトウェア機能を定義する。
Business Analyst	ビジネスアナリスト	解決すべきビジネス上の問題を定義する
Data Analyst	データアナリスト	データの可視化と分析で問題を定量的に特定する。
Domain Expert	業務有識者	深い業務知識を持ちデータの意味やあるべき状態について定義する。

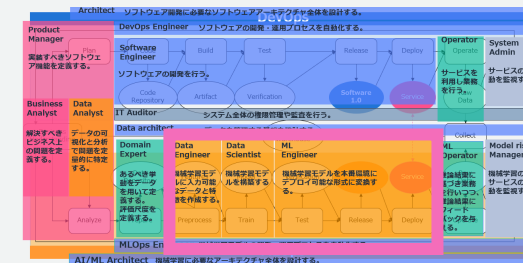
What: 何を依頼すべきか?

情報共有

- metrics (精度、学習時間、推論時間) を共有し、ビジネスが成り立つかを確認してもらう。
- 当初の見込みと大きく変わる場合は対応を考える。

コミュニケーション演習

Who: トレーニング前後のプロセスを担う人への依頼例



英語名	日本語名	役割
Data Engineer	データエンジニア	機械学習モデルに入力可能なデータと特徴を作成する。
ML Engineer	機械学習エンジニア	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。

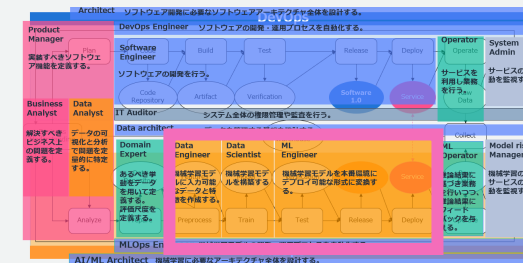
What: 何を依頼すべきか?

レビュー

- アルゴリズム選択の妥当性や精度に違和感がないかを確認してもらう

コミュニケーション演習

Who: 推論や運用に携わる人



英語名	日本語名	役割
ML Engineer	機械学習エンジニア	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。
MLOps Engineer	MLOps エンジニア	機械学習モデルの開発・運用プロセスを自動化する。

What: 何を依頼すべきか?

情報共有

- モデルと推論コードの使い方をDocにまとめて連携する。
- トレーニング時間や推論時間を連携し、テストや運用に備えてもらう。

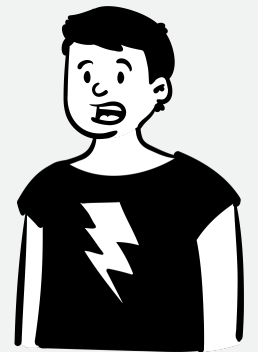
おわりに



(再掲) 本動画のゴール

- 機械学習モデルのアルゴリズムの選定とトレーニングができるようになる
- 機械学習モデルのトレーニングの進捗報告を行いフィードバックがもらえるようになる

Let's Challenge!!



Amazon SageMaker

Prepare

Ground Truth

Create high quality datasets for ML

Data Wrangler

Aggregate and prepare data for ML

Processing

Built-in Python, BYO R/Spark

Feature Store

Store, catalog, search, and reuse features

Clarify

Detect bias and understand model predictions

Build

Studio Notebooks & Notebook Instances

Fully managed Jupyter notebooks with elastic compute

Studio Lab

Free ML development environment

Built-in Algorithms

Integrated tabular, NLP, and vision algorithms

JumpStart

UI based discovery, training, and deployment of models, solutions, and examples

Autopilot

Automatically create ML models with full visibility

Bring Your Own

Bring your own container and algorithms

Local Mode

Test and prototype on your local machine

Train & tune

Fully Managed Training

Broad hardware options, easy to setup and scale

Distributed Training Libraries

High performance training for large datasets and models

Training Compiler

Faster deep learning model training

Automatic Model Tuning

Hyperparameter optimization

Managed Spot Training

Reduce training cost by up to 90%

Debugger and Profiler

Debug and profile training runs

Experiments

Track, visualize, and share model artifacts across teams

Customization Support

Integrate with popular open source frameworks and libraries

Deploy & manage

Fully Managed Deployment

Ultra low latency, high throughput inference

Real-Time Inference

For steady traffic patterns

Serverless Inference

For intermittent traffic patterns

Asynchronous Inference

For large payloads or long processing times

Batch Transform

For offline inference on batches of large datasets

Multi-Model Endpoints

Reduce cost by hosting multiple models per instance

Multi-Container Endpoints

Reduce cost by hosting multiple containers per instance

Inference Recommender

Automatically select compute instance and configuration

Model Monitor

Maintain accuracy of deployed models

Kubernetes & Kubeflow Integration

Simplify Kubernetes-based ML

Edge Manager

Manage and monitor models on edge devices

Studio | RStudio

Integrated development environment (IDE) for ML

MLOps: Pipelines | Projects | Model Registry

Workflow automation, CI/CD for ML, central model catalog

Canvas

Generate, visualize, and share ML models. No code required

AWSの支援例：Amazon SageMaker Training

- 詳しくは [ML-Dark-01](#) へ GO！

Amazon SageMaker Training でオーケストレーション

Amazon SageMaker Training が提供するサービスを 1 文 で表すならば

「用意したコード」と「用意したデータ」を
「指定したコンピューティングリソース」と「用意した環境」で実行し、
「実行履歴を自動記録」して「アーティファクトを自動で保存」する機能を
提供する



※ただし全てをユーザで用意する必要はなく AWS でも用意している

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

9

Train & tune

Fully Managed Training

Broad hardware options, easy to setup and scale

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-workshop>

aws-samples / aws-ml-enablement-workshop Public

<> Code Issues 1 Pull requests Discussions Actions Projects Security Insights

main 2 branches 0 tags Go to file Code

icoxfog417 and icoxfog417 プロダクトマネージャー向け機械学習のインプット資料の... 9370a29 4 days ago 28 commits

.github/ISSUE_TEMPLATE Update based on feedbacks 4 months ago

data 解約率予測シナリオの追加 (#6)

docs

model

notebooks

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

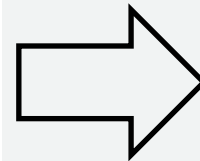
ML Enablement Seriesの動画

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](#)



Thank you!