



【AI開発道場】 AI概論



目次

- AIとは
- 機械学習とは
- AI開発フロー
- ツール・ライブラリ
- まとめ



目次

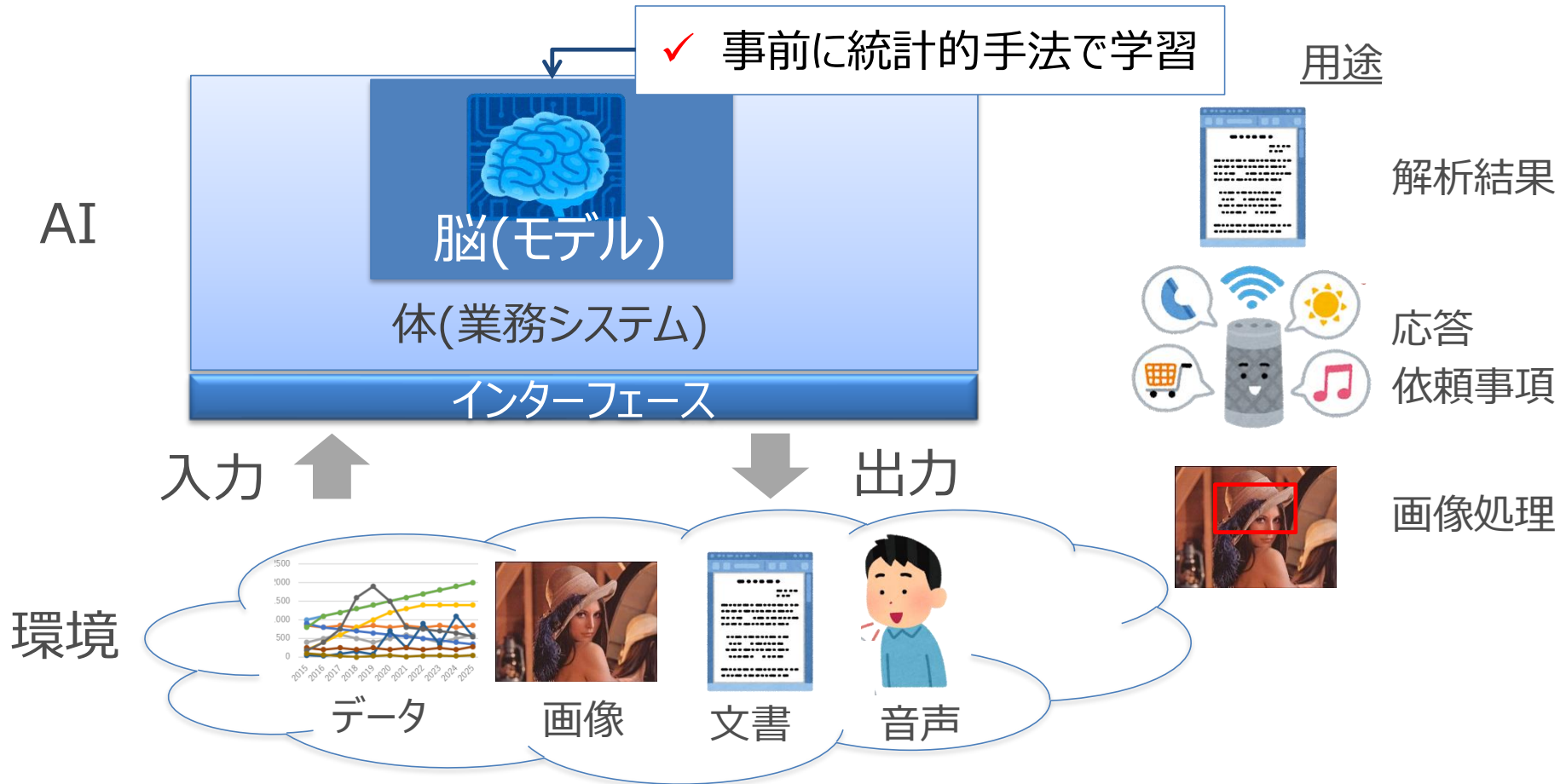
- AIとは
- 機械学習とは
- AI開発フロー
- ツール・ライブラリ
- まとめ

AIとは

- Artificial Intelligence : 人工知能
- 知覚・試行・推論・記憶など、人間の知的能力をコンピュータ・各種マシンによって代替する技術
- 厳密な定義は存在しない
- 『人工的につくられた人間のような知能』
— 松尾 豊 —
- 『ソフトウェアによる、知覚と知性の実現』
— 安宅 和人 —

AIとは

様々なデータを、コンピュータ上の事前学習済みの脳(モデル)で認知・処理することで価値あるアウトプットを行う仕組み



AI、機械学習、ディープラーニング

AI（人工知能）

人間の知的活動を再現する機械・システム

機械学習

コンピュータに学習能力を与える技術

要素技術

エキスパート
システム

ロボティクス

ヒューマンイン
ターフェース

⋮

アルゴリズム

SVM

ランダム
フォレスト

K-means

⋮

ニューラルネットワーク

人間の脳の仕組みを模した学習モデル

ディープラーニング

多層ネットワークによる学習

ネットワーク

RNN

CNN

GAN

DQN

弱いAIと強いAI

□ 強いAI

- 枠を超えて考えることのできる人工知能
- 人間のようにものを考え、認識・理解し、人間のような推論・価値判断をもとに実行することができるもの

□ 弱いAI

- ある一定の枠の範囲内で考える人工知能
- 範囲内に限れば、人間のレベルを超えている分野も存在する

AIのレベル

レベル 1 : 単純な制御プログラム

- 単純な規則に基づく人工知能（例：エアコン、冷蔵庫）

レベル 2 : 対応パターンが非常に多いもの

- 定義されている規則が多い人工知能（例：将棋プログラム、掃除ロボット、FAQ）

レベル 3 : 対応パターンを自動的に学習するもの

- 機械学習を取り入れた人工知能（例：検索エンジン、ビックデータ分析）

レベル 4 : 対応パターンの学習に使う特徴量も自力で獲得するもの

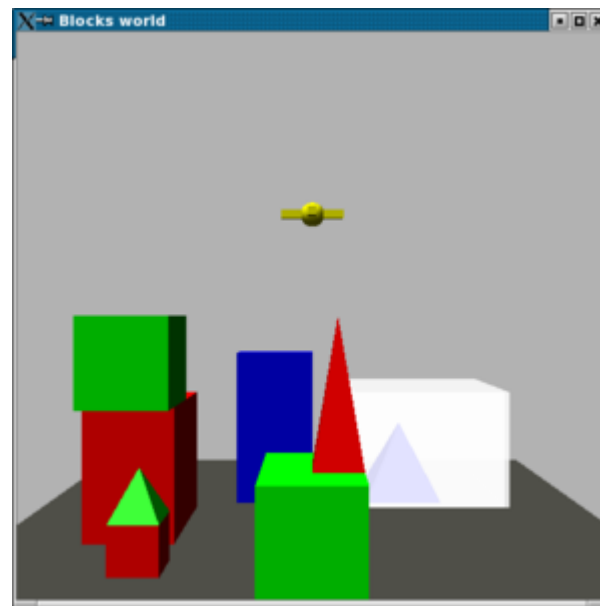
- 高度な分析が可能な人工知能（例：ディープラーニングを使った人工知能）

AIの歴史

| | 人工知能の置かれた状況 | 主な技術等 | 人工知能に関する出来事 |
|--------|-----------------------|---|---|
| 1950年代 | | | チューリングテストの提唱 (1950年) |
| 1960年代 | 第一次人工知能ブーム (探索と推論) | <ul style="list-style-type: none"> 探索、推論 自然言語処理 ニューラルネットワーク 遺伝的アルゴリズム | ダートマス会議にて「人工知能」という言葉が登場 (1956年) ニューラルネットワークのパーセプトロン開発 (1958年) 人工対話システムELIZA開発 (1964年) |
| 1970年代 | 冬の時代 | <ul style="list-style-type: none"> エキスパートシステム | 初のエキスパートシステムMYCIN開発 (1972年) MYCINの知識表現と推論を一般化したEMYCIN開発 (1979年) |
| 1980年代 | 第二次人工知能ブーム (知識表現) | <ul style="list-style-type: none"> 知識ベース 音声認識 | 第五世代コンピュータプロジェクト (1982～92年) 知識記述のサイクプロジェクト開始 (1984年) 誤差逆伝播法の発表 (1986年) |
| 1990年代 | 冬の時代 | <ul style="list-style-type: none"> データマイニング オントロジー | |
| 2000年代 | 第三次人工知能ブーム (機械学習) | <ul style="list-style-type: none"> 統計的自然言語処理 ディープラーニング | ディープラーニングの提唱 (2006年) |
| 2010年代 | | | ディープラーニング技術を画像認識コンテストに適用 (2012年) |

第一次AIブーム

- 推論と探索の時代
- 登場した主な人工知能
 - パズルやゲームを解くもの
 - 行動計画を立てるもの
 - 数学の定理を証明するもの
- 課題
 - 組み合わせ爆発
 - 現実の問題へ適用できない



第二次AIブーム

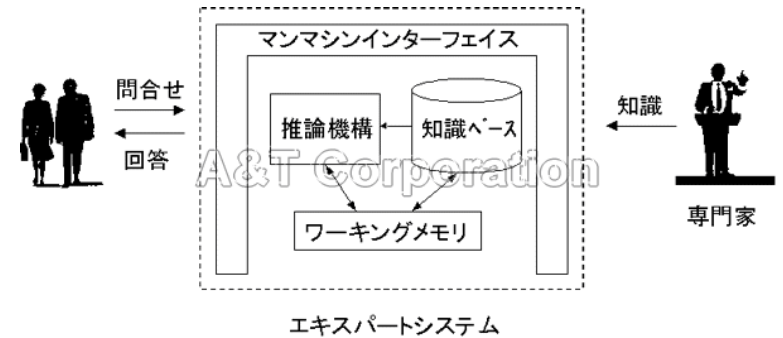
□ 知識表現の時代

□ 登場した主な人工知能

- 対話システム「ELZA」
- 感染症診断システム「MYCIN」
- 有機化合物推定システム「DENDRAL」

□ 課題

- 知識表現の難しさ
- シンボルグラウンディング問題
- フレーム問題



第三次AIブーム

- 機械学習の時代
- 登場した主な人工知能
 - 検索エンジン
 - Watson
 - AlphaGo
- 課題
 - フレーム問題





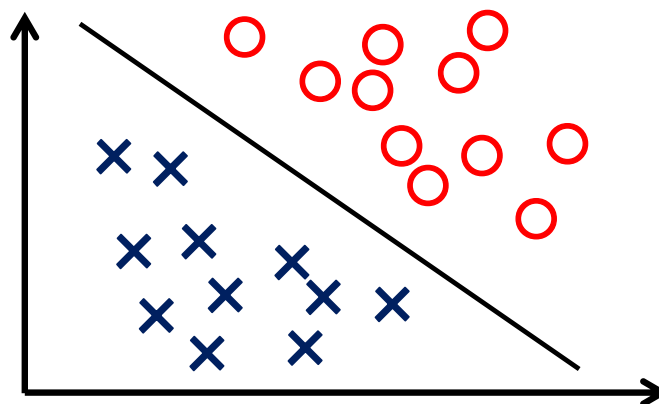
目次

- AIとは
- 機械学習とは
- AI開発フロー
- ツール・ライブラリ
- まとめ

機械学習とは

□ 機械学習とは・・・

- 人間が自然に行っている学習能力と同様の機能をコンピュータで実現しようとする技術
 - 学習の根幹は、「分ける」という処理
 - コンピュータが大量のデータを処理しながら、「分け方」を自動的に習得する
 - 分け方には様々な手法が存在する



機械学習とは

□ 定義

■ 厳密な定義は存在しない

- 「機械学習とは、データから反復的に学習し、そこに潜むパターンを見つけ出すこと。そして学習した結果を新たなデータにあてはめることで、パターンにしたがって将来を予測すること。

- SAS社 -

- 「明示的にプログラムしなくても学習する能力をコンピュータに与える研究分野」

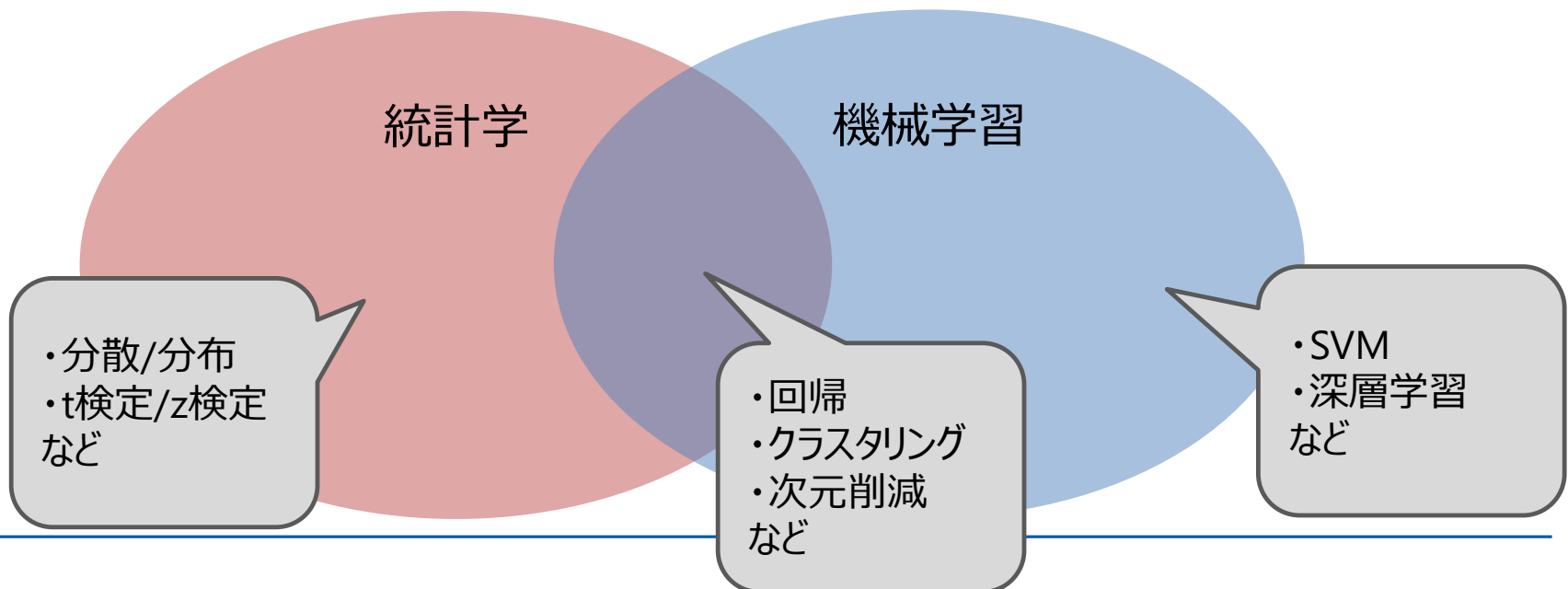
-アーサー・サミュエル-

- 「コンピュータプログラムが、ある種のタスクTと評価尺度Pにおいて、経験Eから学習するとは、タスクTにおけるその性能をPによって評価した際に、経験Eによってそれが改善されている場合である」

-トム・M・ミッチェル-

統計学と機械学習

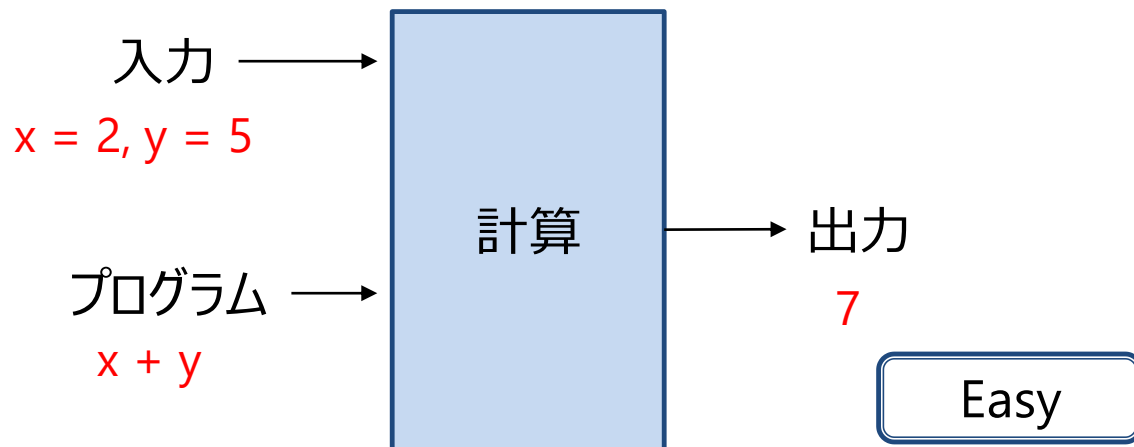
- ❑ 統計学はデータを「説明」することにより重きを置く
- ❑ 機械学習はデータから「予測」することにより重きを置く
- ❑ とはいえ、統計学と機械学習の違いは基本的にはそれほど大きくないし互いに重なる部分だらけ



プログラミングと機械学習

□ プログラミング

2 + 5 = ?



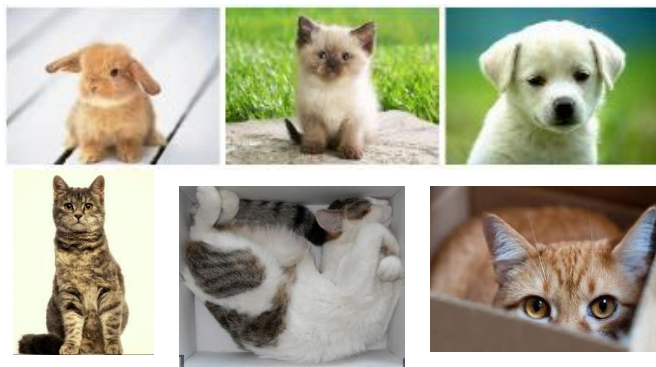
猫 or Not ?



プログラミングと機械学習

□ 機械学習

・画像



・それぞれの画像が猫か否か
(Not, 猫, Not,
猫, 猫, 猫)

入力 →

出力 →

アルゴリズム

SVM, K-means, DNN

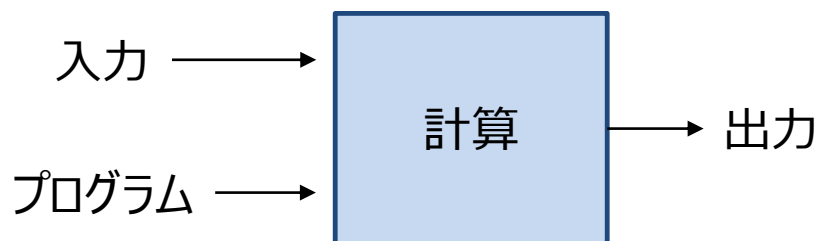


→ プログラム

画像が猫か否かを
判断するプログラム
(モデル)

プログラミングと機械学習

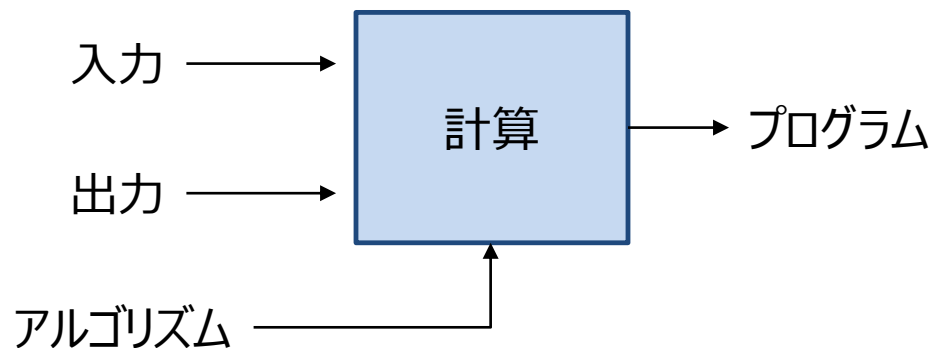
□ プログラミング



- **人が記述**
タスク仕様の定義
アルゴリズムは固定
プログラムの説明が容易

□ 機械学習

- **ソフトウェアが記述**
目的：**汎化**
アルゴリズムは**データに依存**
プログラムの**説明は困難**



人がデータ(答え)
を教える

データ中のパターンをコンピュータに探させる
(=学習させる)

学習って？

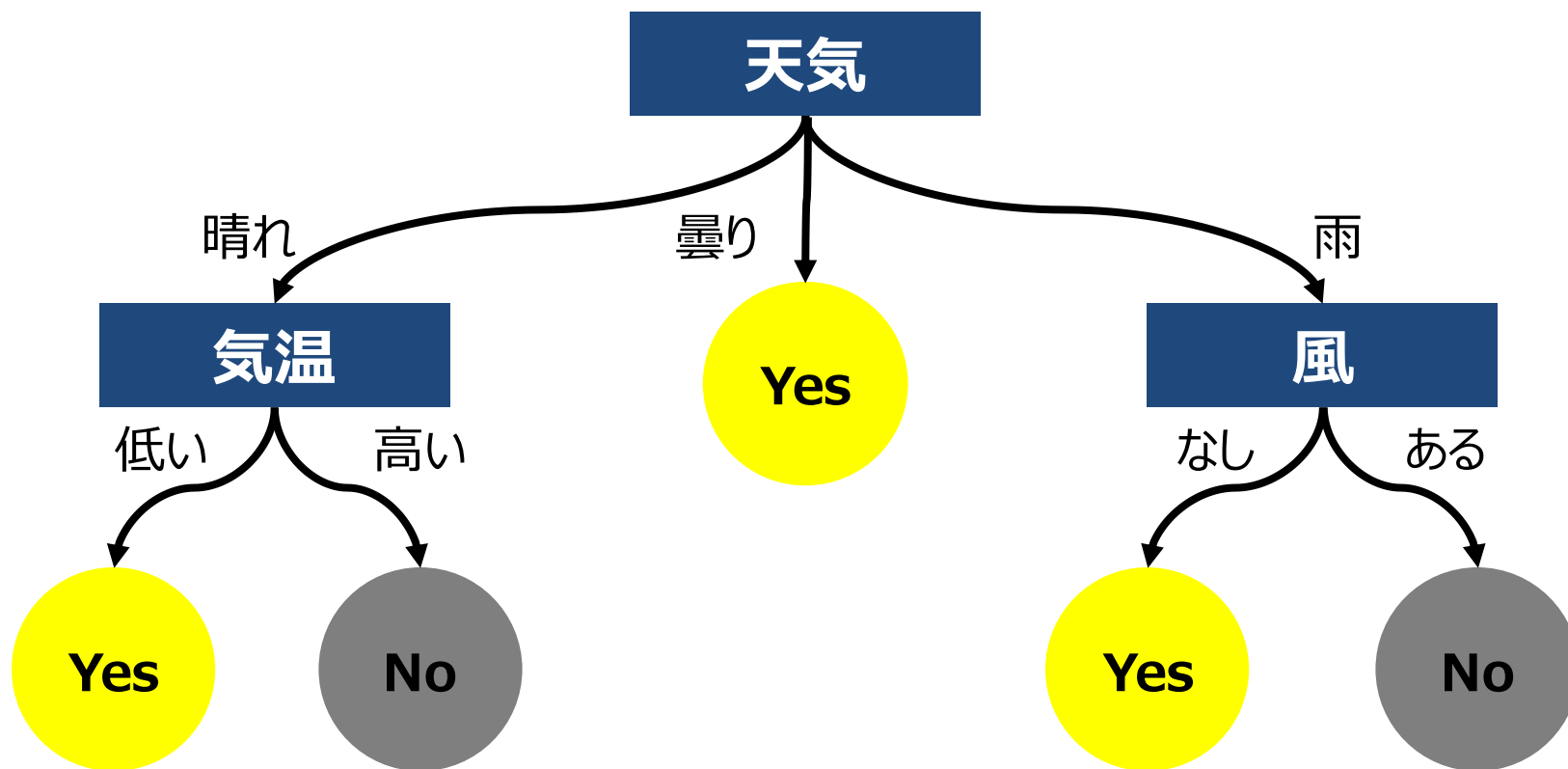
□ 試合はあるでしょうか？

| 天気 | 気温 | 風 | 試合をしたか？ |
|----|----|----|---------|
| 晴れ | 低い | ある | Yes |
| 晴れ | 高い | ある | No |
| 晴れ | 高い | なし | No |
| 曇り | 低い | ある | Yes |
| 曇り | 高い | なし | Yes |
| 曇り | 低い | なし | Yes |
| 雨 | 低い | ある | No |
| 雨 | 低い | なし | Yes |
| 晴れ | 低い | なし | ？ |

学習って？

□ 判断ロジック

“Model”



データに注意

入力形式が
バラバラ

データの偏り
データ不足

| 天気 | 気温 | 風 | 場所 | 試合をしたか？ |
|----|----|----|------|---------|
| 晴れ | 25 | ある | さいたま | Yes |
| 晴れ | 17 | ある | さいたま | Yes |
| 晴れ | 高い | なし | 東京 | No |
| 曇り | 5 | — | 千葉 | No |
| 雨 | 低い | なし | 神奈川 | No |

欠損値がある
(NULL)

機械学習の分類

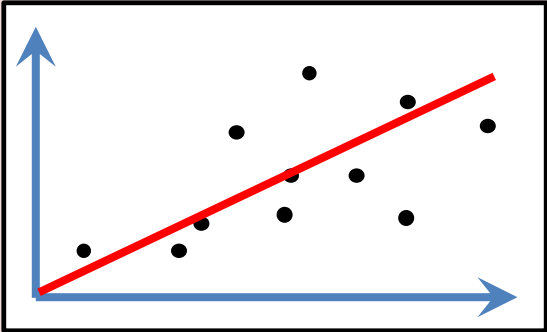
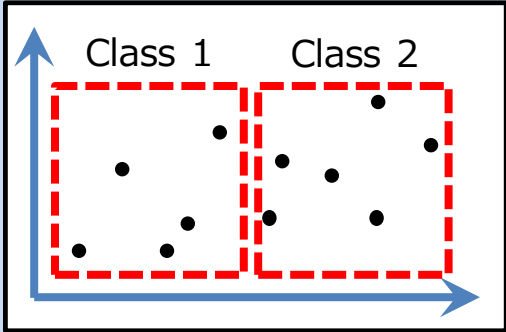
タスクで分ける

- 回帰
- 分類

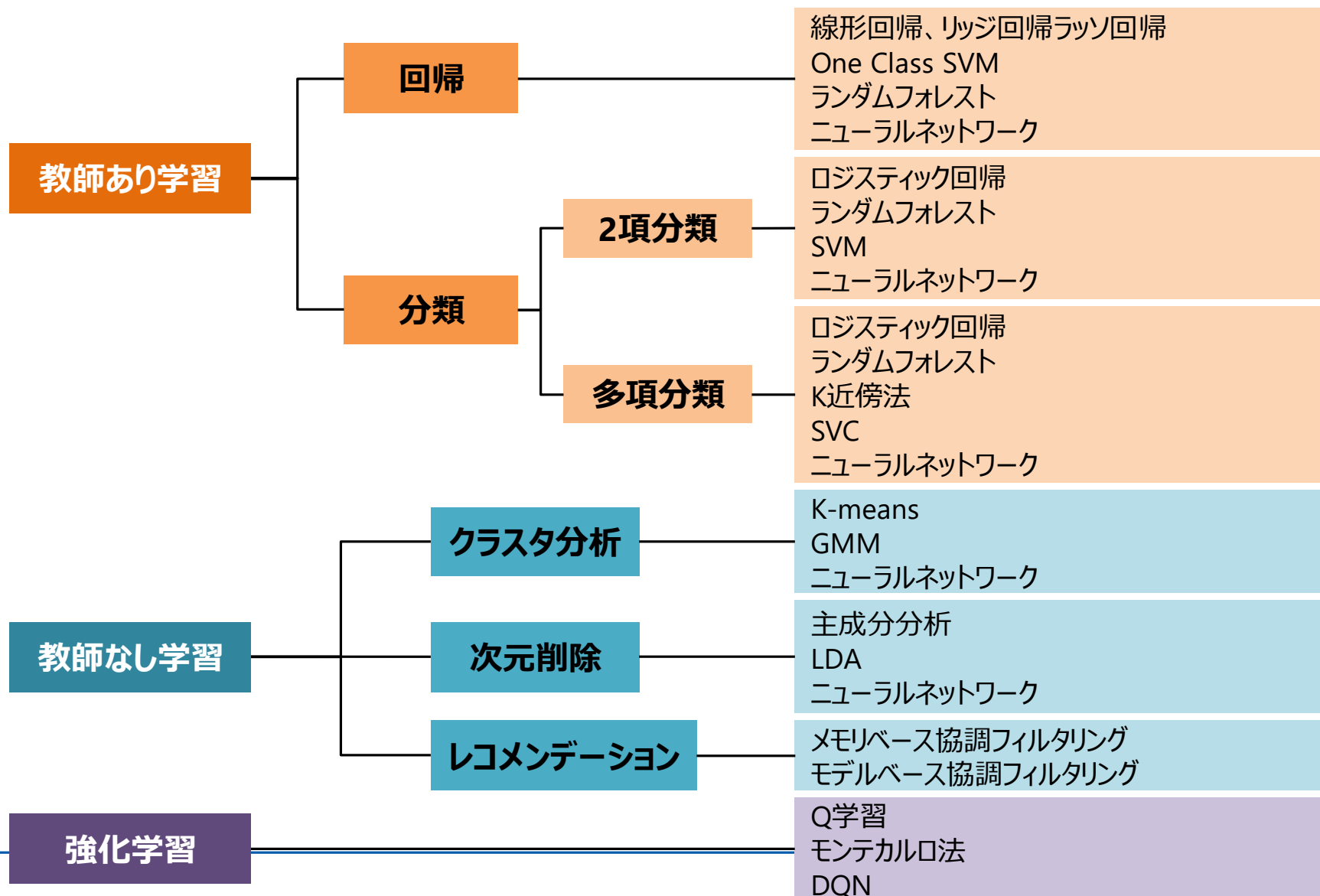
教師のあり・なし

- 教師あり
- 教師なし
- 半教師あり
- 強化学習

タスクで分ける

| | 回帰 | 分類 |
|------|--|---|
| 出力 | 実数値を出力する | クラス（分類結果）を出力する |
| 入力 | 正解値の付いたデータセットを使用 | クラスラベル（正解）の付いたデータセットを使用 |
| 用途例 | <ul style="list-style-type: none">売上予測需要予測品質管理貸倒率予測 | <ul style="list-style-type: none">不良品予測故障予測販売効果測定顧客セグメンテーション |
| イメージ |  |  |

教師のあり・なしで分ける



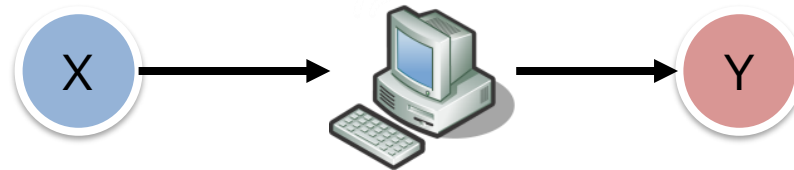
教師あり学習

□ 学習時

- 入力 X に対して期待される出力 Y を教える

□ 予測時

- 未知の X に対応する出力 Y を予測する



□ 分類と回帰

分類

- 出力 Y がカテゴリーの場合
- 応用例：スパム判定、記事分類、属性推定、etc
- 手法例：SVM、ナイーブベイズ、決定木学習、etc

回帰

- 出力 Y が実数値の場合
- 応用例：電力消費予測、年収予測、株価予測、etc
- 手法例：ベイジアンネットワーク、ランダムフォレスト、ニューラルネット、etc

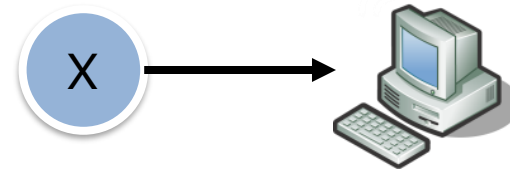
教師なし学習

□ 学習時

- 入力 X をたくさん与えると、入力情報自体の性質に関して何らかの結果を返す

□ 予測時

- 未知の X の性質を返す



□ 手法例

- クラスタリング
 - 与えられたデータを、いくつかのまとまりに分ける
- 異常検知
 - 入力データが異常かどうか判定する

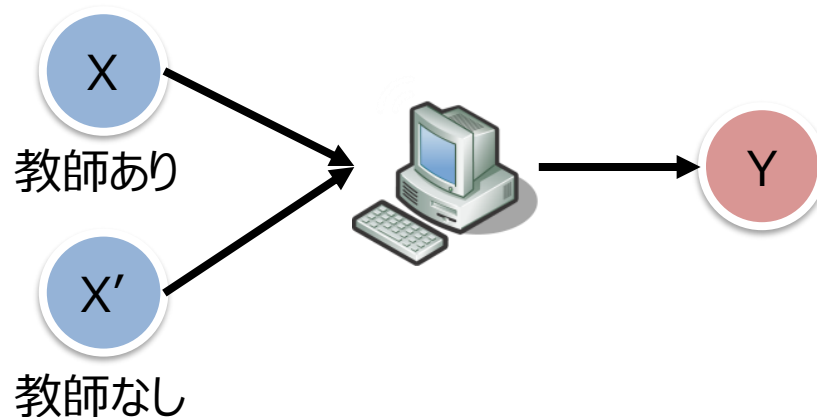
半教師あり学習

□ 学習時

- 教師ありデータ X （入力と対応する出力のペア）と、教師なしデータ X' （入力のみ）から、学習する

□ 予測時

- 未知の X に対応する出力 Y を予測する



□ 手法例

- Self-training
 - 教師ありデータのみで学習し、教師無データを分類する
その後、高いものを教師ありデータに加え、再度学習する

強化学習

□ 学習時

- 環境の観測結果 S と、ある行動選択に基づき、行動 A を行うと環境から報酬 R が返ってくる
- この報酬 R がより大きくなるように、行動選択の方針を更新していく

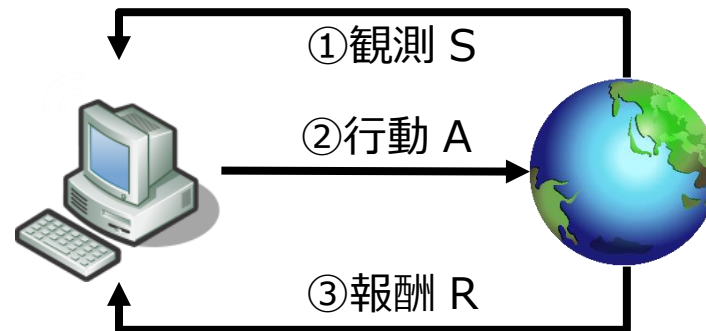
□ 予測時

- 観測結果 S と、行動選択に基づき、行動 X を起こす

□ 手法例

■ Q学習

- 観測結果 S のもと、行動 A を選択する価値を学習する





目次

- AIとは
- 機械学習とは
- AI開発フロー
- ツール・ライブラリ
- まとめ

AI開発フロー

対象業務選定

AI適用業務を選定する

基礎検討

業務要件、アーキテクチャ、KPIを検討する

業務システム
開発

業務システムの開発を行う

モデル開発

モデルの開発を行う

適用・KPI評価

業務システムへ適用し、KPIの評価を行う

運用

実運用を行いながら、PDCAサイクルを回す

対象業務選定

データの
収集が容易

- モデルの学習に必要なデータを集めることができる



類似事例が
存在

- 類似事例があれば、参考や流用が可能

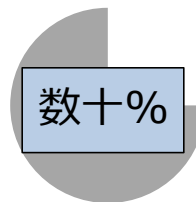


Point

AIが一定の
精度でよい

- 例えばAIの精度が60%でも以下を検討することで活用場面が増える
 - うまく人のサポートツールとして使う
 - アプリケーションでフォローする

数十%



KPIが
設定しやすい

- 業務上のKPI設定、評価を行うことでAI導入効果を適切に把握できる
 - AIの精度に一喜一憂するのはNG

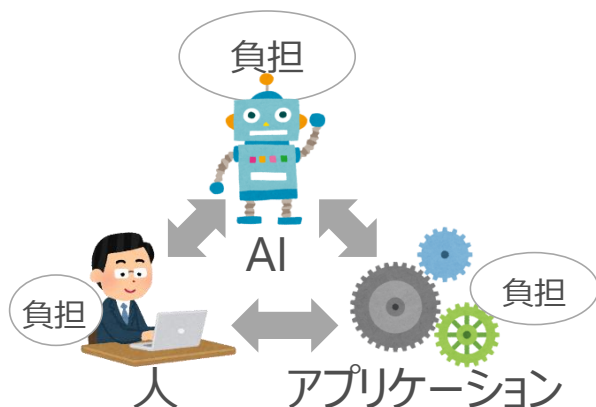


AI適用特有の観点が必要

- 業務要件の確認
- アーキテクチャ検討
- KPIの設計

業務要件の確認

AI・アプリケーション・人の作業分担の定義



- 業務改善を実現するため、「AI・アプリケーション・人の全てを活用する」という視点を持って業務を定義すべき
- AIへの過度な期待を防止



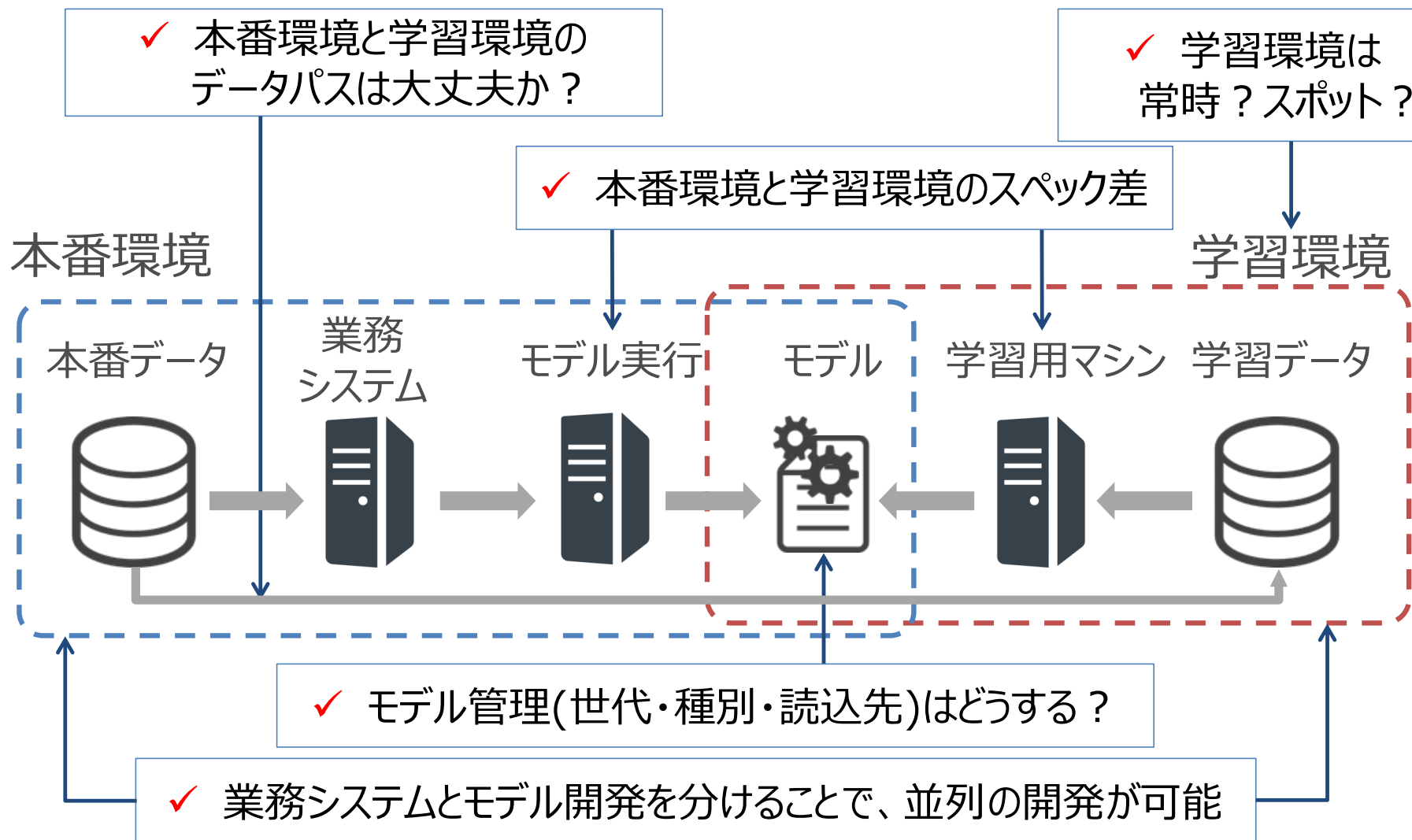
AIが全部やってくるんじゃないの？
こんなはずじゃなかった・・・

性能要件の確認



- 業務フローを回すために必要な性能
 - 学習時間
 - 実行時間
- アーキテクチャ構築のベース





AIを導入による業務効果を定義し、KPIを設定

コスト削減



品質、売上向上



リスク低減



Point

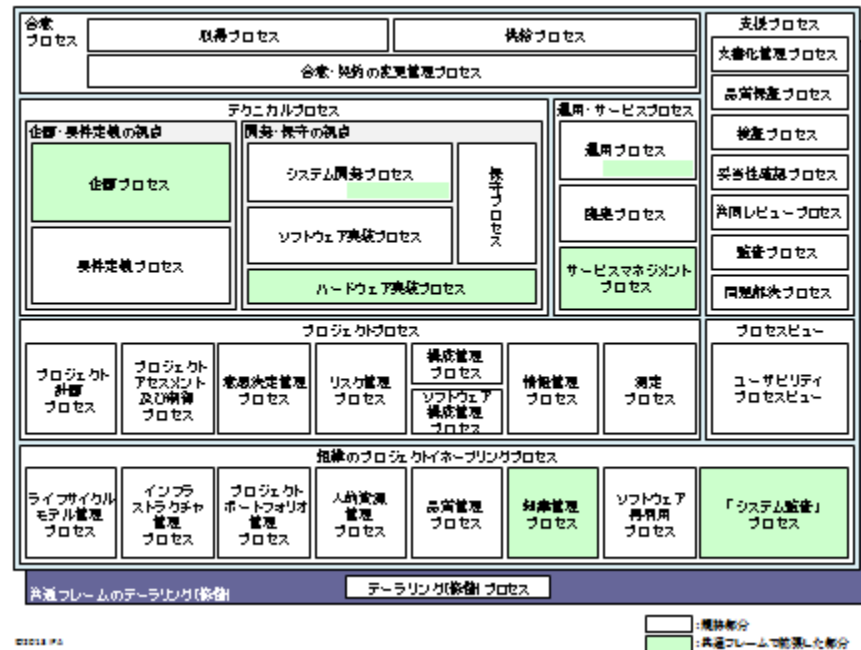
- ✓ KPIは**業務的な**指標を設定
- ✓ AIの精度はそれを元にして決定

業務システム開発

- 業務要件に合わせ開発する
- 標準的なシステム開発プロセスに準じる



図2-12
共通フレームの基本構成



□ モデル開発の流れ

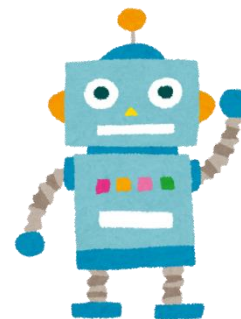
データの
収集

データの
整形

モデルの
生成

モデルの
評価

チューニング



データの収集

データの
収集

データの
整形

モデルの
生成

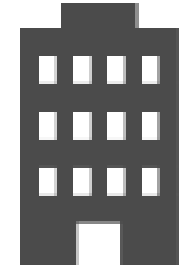
モデルの
評価

チューニ
ング

候補は3つ

社内データ

- データが社内にのみにある場合は必須
- クラウド利用などのセキュリティに注意



オープン
データ

- 一般的に出回っているデータ
- 安価に大量に手に入れることも可能
- 社内にはないデータある



人力で作成

- 上記で手に入らないデータも取得可能
- 手作業なのでコストが高い（最終手段）



データ整形

データの
収集

データの
整形

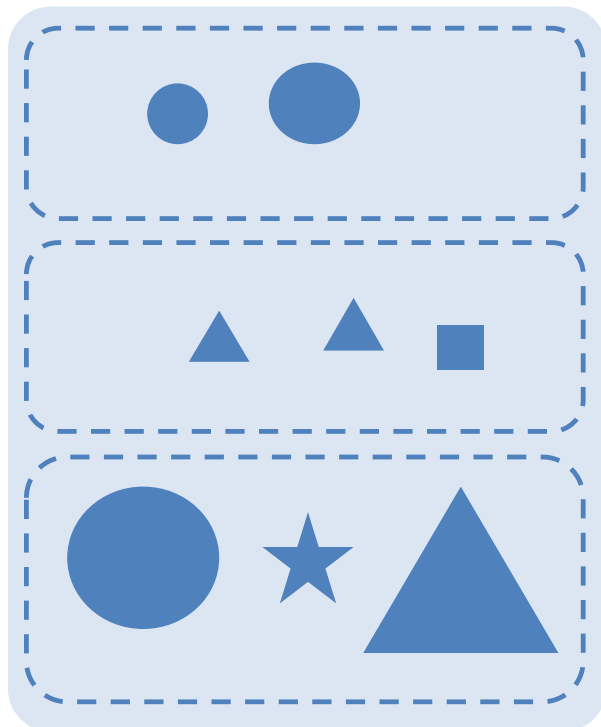
モデルの
生成

モデルの
評価

チューニ
ング

- ❑ 収集したデータをモデル生成の入力に合わせて整形
- ❑ 複数データの結合、欠損値・異常値の対応

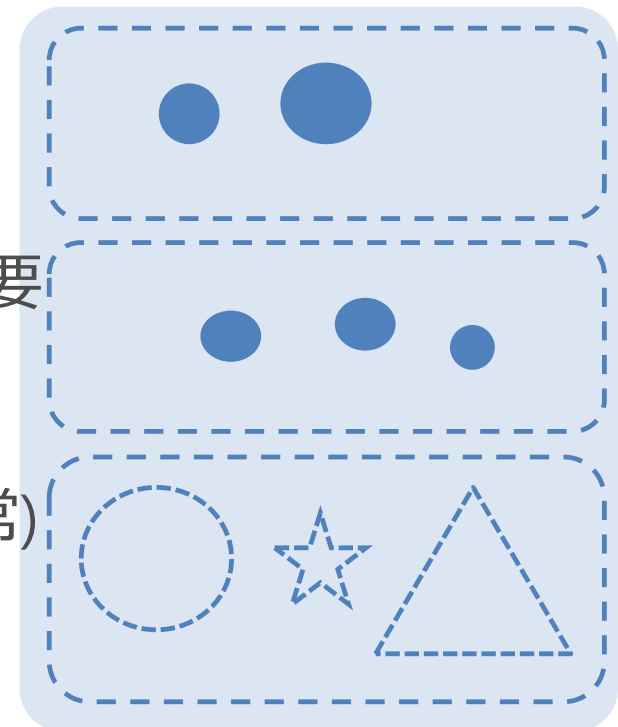
収集データ



そのまま使える



整形済みデータ



データ整形が必要



採用しない(異常)



モデルの生成

データの
収集

データの
整形

モデルの
生成

モデルの
評価

チューニ
ング

整形したデータを入力に、機械学習アルゴリズムを用いてモデルを作成

学習用データ

機械学習アルゴリズム

モデル



統計的にデータの
特徴を抽出

モデルの評価

データの
収集

データの
整形

モデルの
生成

モデルの
評価

チュー
ニング

- 精度・解釈性・過学習度合い・実行時間等の指標で作成したモデルを評価
- 評価が悪い場合は、チューニングへ

評価用データ



モデル



判定



結果

精度

| | |
|---|-----|
| ○ | 70% |
| × | 30% |

性能



評価

チューニング

データの
収集

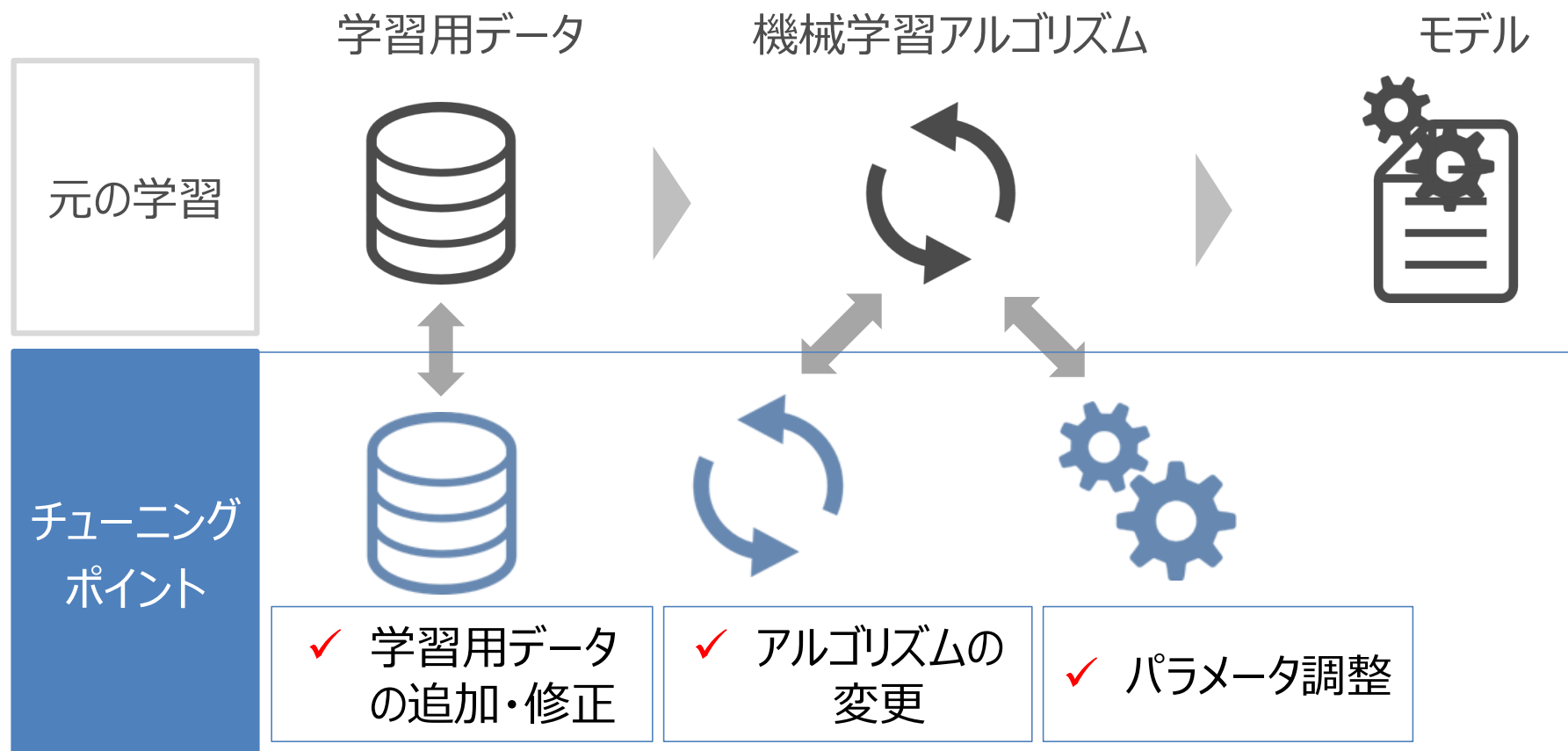
データの
整形

モデルの
生成

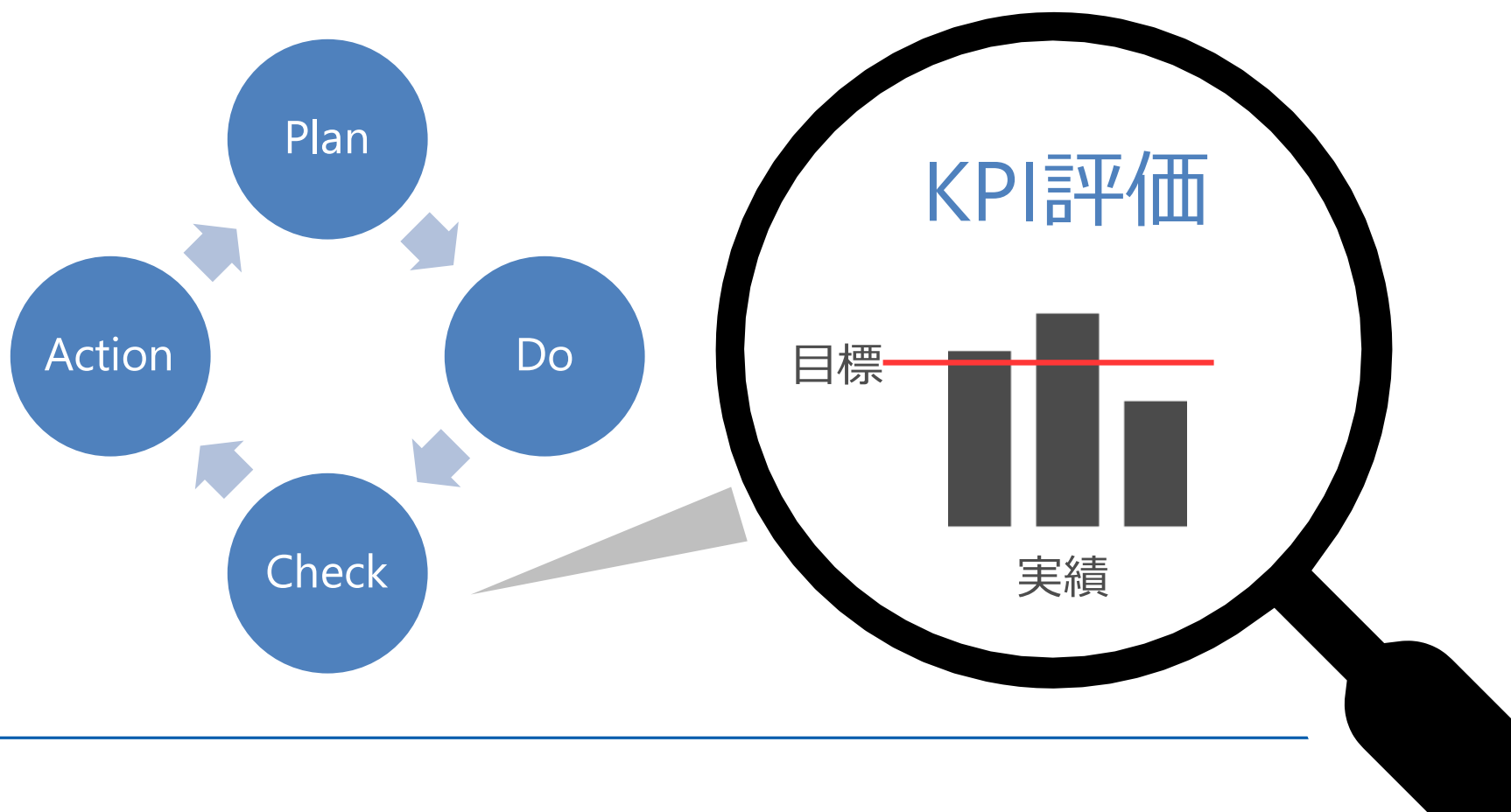
モデルの
評価

チューニ
ング

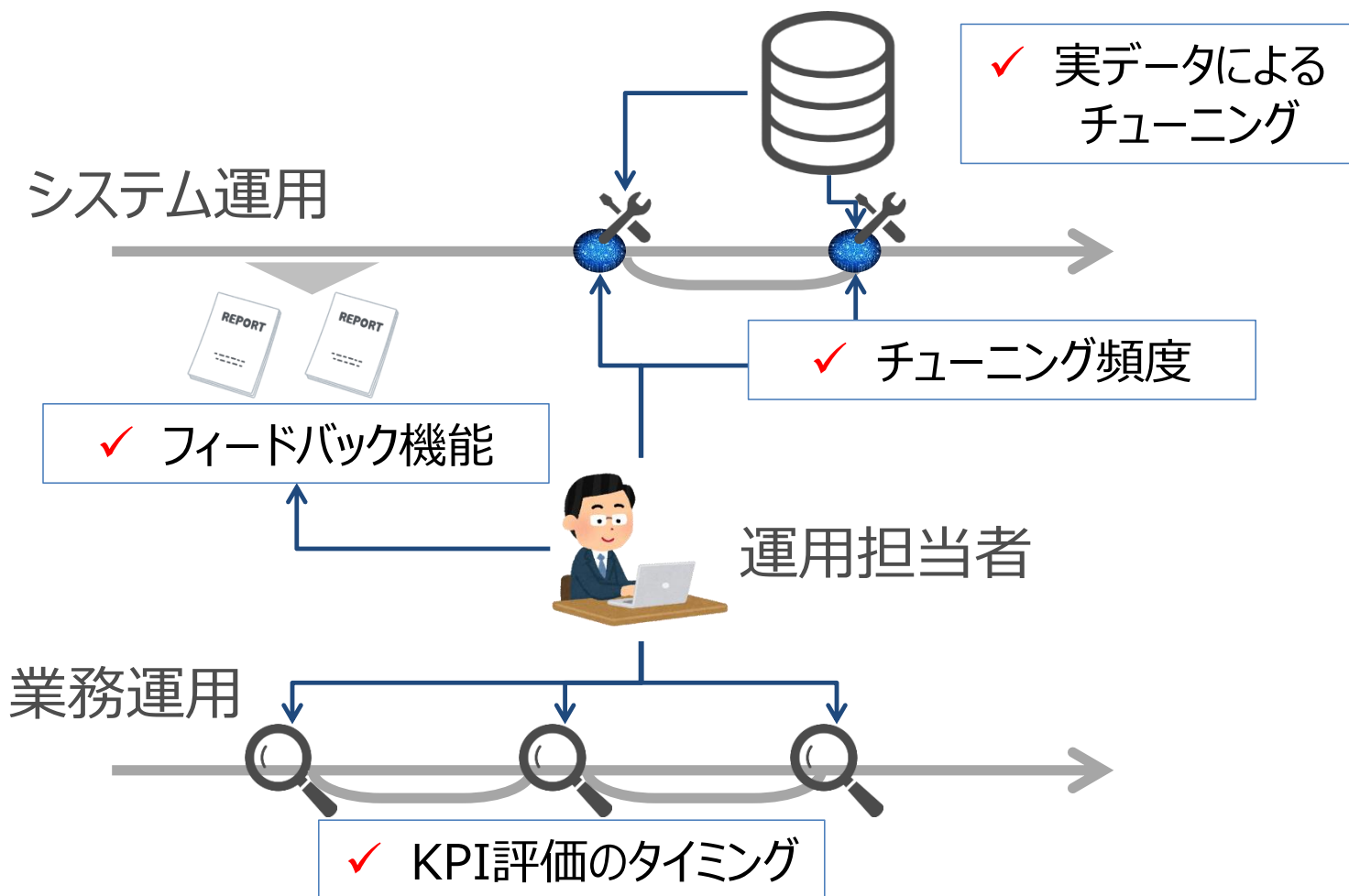
入力データ・アルゴリズム・学習時のパラメータを修正
することで、モデルをチューニングする



- ❑ 設定したKPIをもとに業務システムとして評価
- ❑ PDCAサイクルを回して改善する



AI独自の運用としてモニタリング・チューニングサイクルを構築





ケーススタディ： モザイク処理業務へのAI導入

対象業務選定

□ 提案プロセスの評価基準をもとに業務を選定

| 業務 | データ収集 | 実現性 | 精度対効果 | KPI設定 |
|------------------|-------|-----|-------|-------|
| 経理における 消込業務 | × | × | ○ | ○ |
| モザイク処理 業務 | △ | ○ | ○ | ○ |
| 映像と音声の ズレ確認業務 | × | △ | △ | ○ |

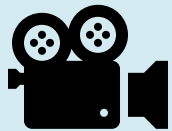
基礎検討：業務要件

□ 業務フローの整理

Point

- ✓ AIと人との役割分担を明確にする
- ✓ 最終確認は人の手で

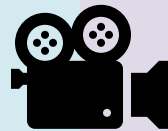
元動画



自動モザイク



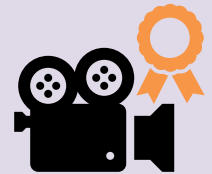
AI処理済み動画



確認・修正



成果物



AIの作業

人の作業

基礎検討：業務要件

□ ケーススタディのスコープで 業務要件を整理

Point

✓ 業務要件 → AIの精度目標の順

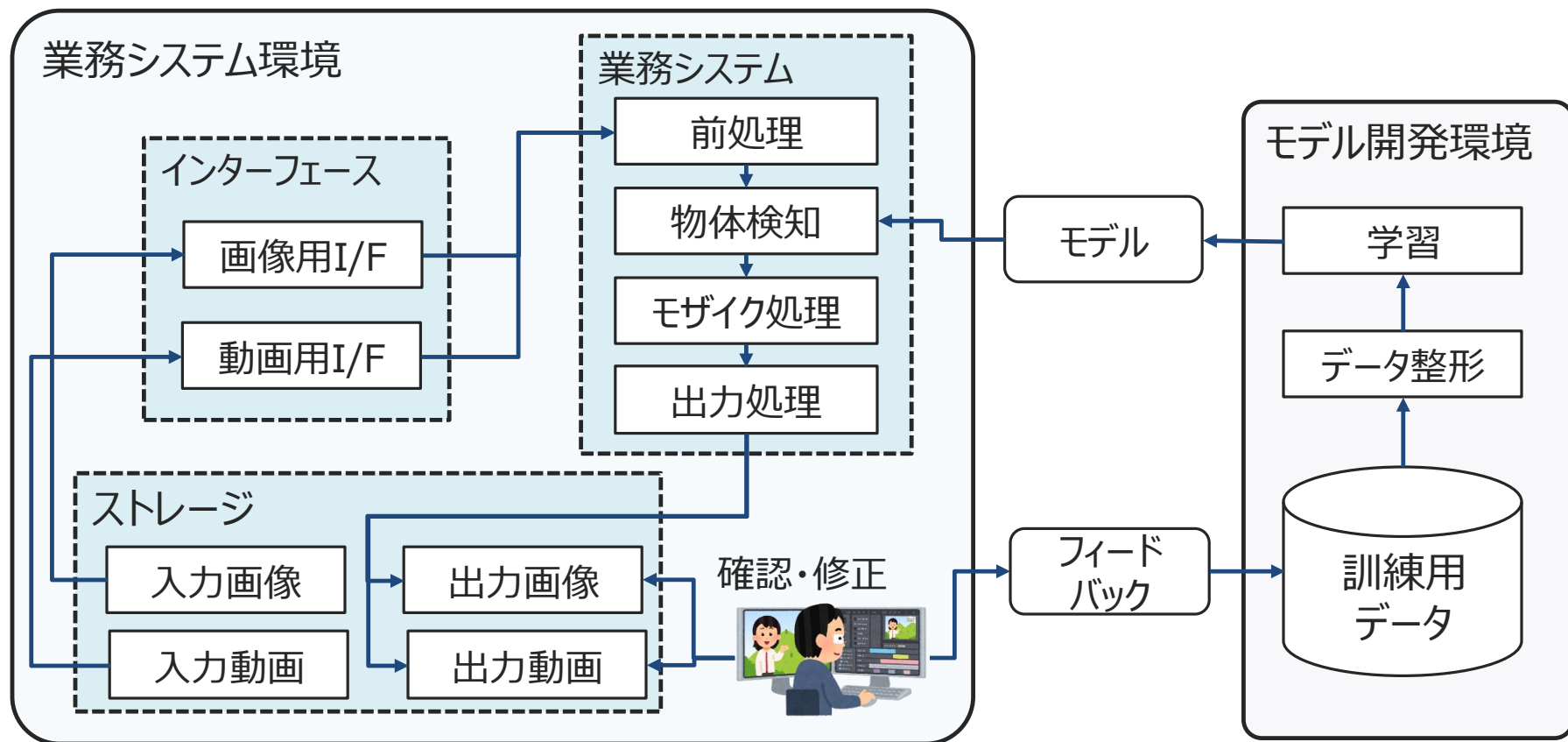
| | | |
|--------|----------|----------------|
| 機能要件 | モザイク処理対象 | 車のナンバープレート |
| | 入力データ | 画像ファイル、動画ファイル |
| システム要件 | HW構成 | 個人PC |
| | SW構成 | python、OpenCV |
| 性能要件 | 録画データ | 動画時間と同時間での処理 |
| | 画像 | 録画データの処理時間に準ずる |
| | ストリームデータ | 動画時間の8割の時間での処理 |
| 運用要件 | 運用 | 運用はスコープ外とする |

基礎検討：アーキテクチャ

□ アーキテクチャ

Point

- ✓ データの流れに注意する

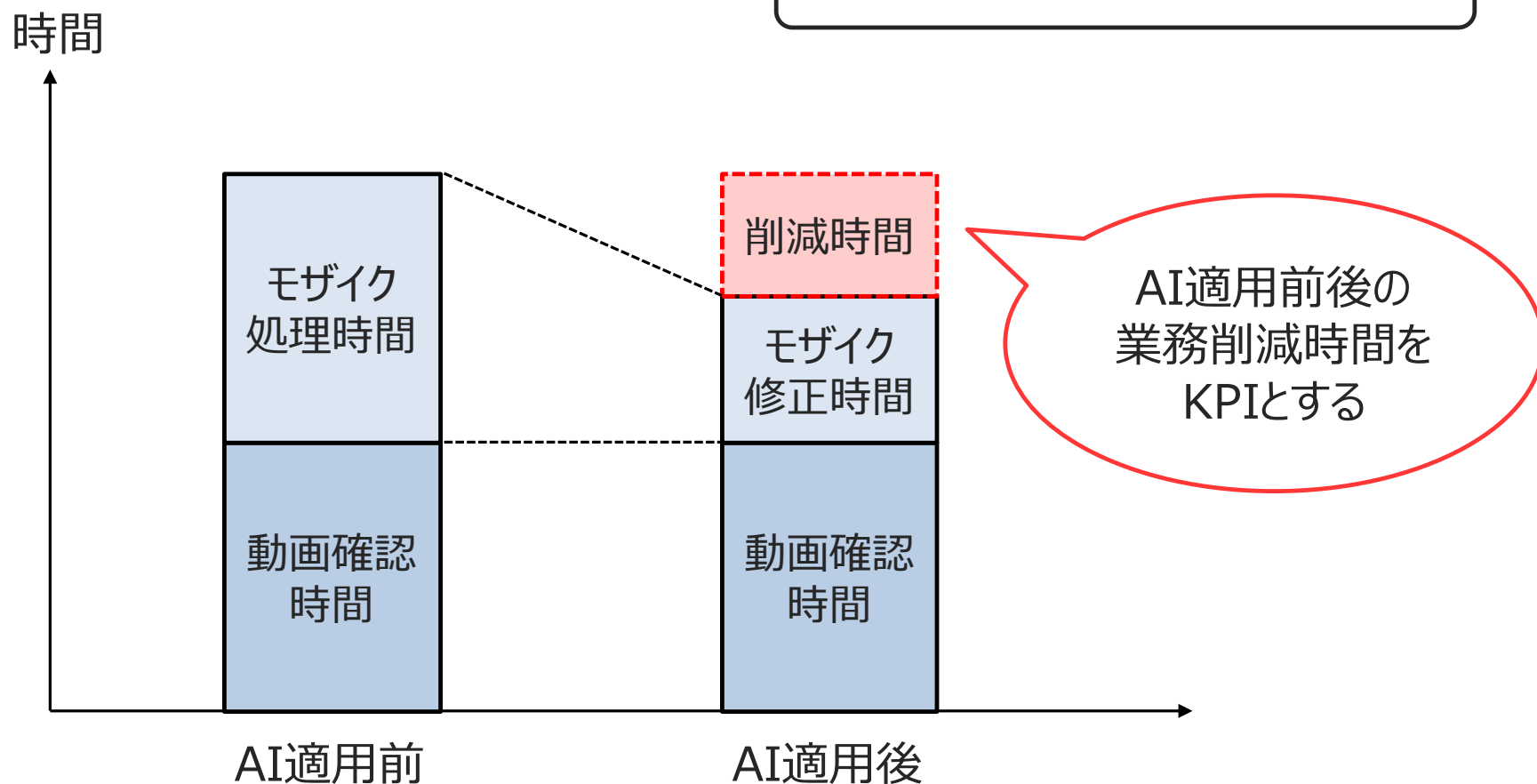


基礎検討：KPI設定

□ KPIを設定

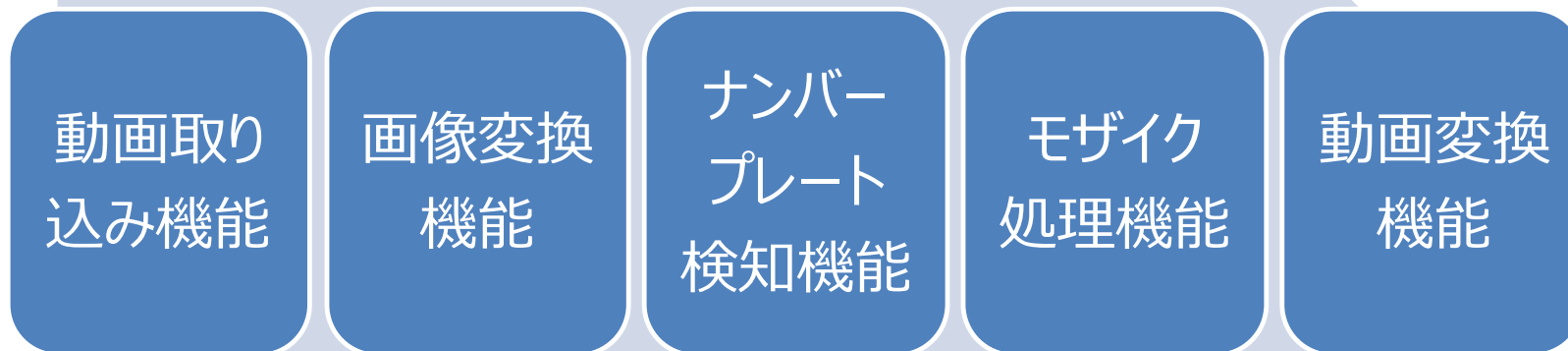
Point

- ✓ 業務目線のKPIを設定する



業務システム開発

□ 今回開発した機能は以下の通り



※人の手による修正機能や教師データ収集機能は、
実運用がないためスコープ外とした

業務システム開発

□ 開発環境

- 個人PC上で開発
- 利用ソフトは以下の通り



• python

オープンソースの汎用プログラミング言語
データ分析や機械学習分野にて広く利用されている



• OpenCV

オープンソースの画像処理ライブラリ
標準で物体検知の学習機能を搭載

業務システム開発：動画取り込み・画像変換機能

動画取り
込み機能

画像変換
機能

ナンバー
プレート
検知機能

モザイク
処理機能

動画変換
機能

読み込んだ動画を1フレームごとの画像に
分解する機能

動画



分解



1フレーム目：



2フレーム目：



3フレーム目：



⋮

Nフレーム目：



別々の
画像

業務システム開発：ナンバープレート検知機能

動画取り
込み機能

画像変換
機能

ナンバー
プレート
検知機能

モザイク
処理機能

動画変換
機能

モデルを読み込んで画像に適用し、
ナンバープレートの座標を検知する機能



モデルの適用



業務システム開発：モザイク処理機能

動画取り
込み機能

画像変換
機能

ナンバー
プレート
検知機能

モザイク
処理機能

動画変換
機能

検知した座標にモザイク処理を施す機能



業務システム開発：動画変換機能

動画取り
込み機能

画像変換
機能

ナンバー
プレート
検知機能

モザイク
処理機能

動画変換
機能

モザイク処理が完了した画像を統合し、
動画へ変換する機能

1フレーム目：



2フレーム目：



3フレーム目：



⋮

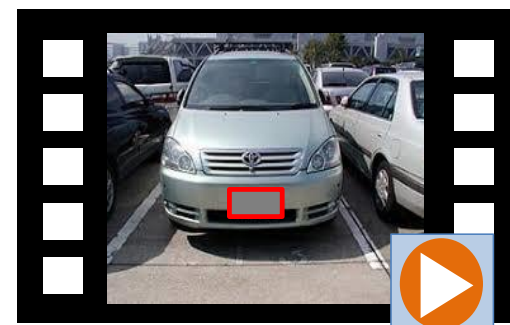
Nフレーム目：



別々の
画像

統合

動画



モザイク済み動画の完成！

モデル開発

- モデル作成プロセスに従って、
ナンバープレートを検知するモデルを作成

データ
収集

データ
整形

モデル
生成

モデル
評価

チューニ
ング

モデル開発：データ収集

□ 訓練用データを収集

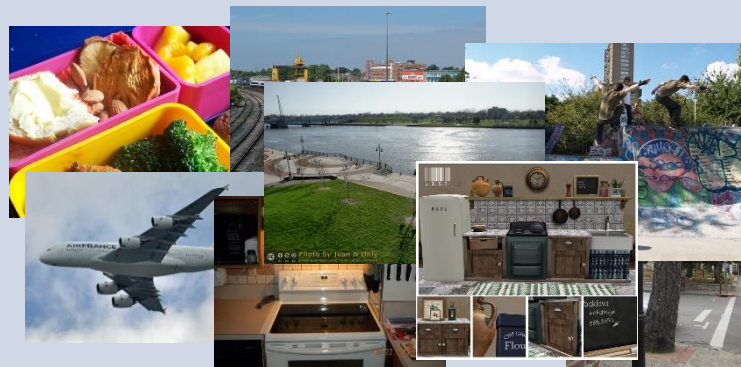
正解データ



- スマホで撮影
- メンバーで手分けして収集

976枚

不正解データ



- オープンデータを利用
- ナンバープレートが映っていない画像を収集

1500枚

データ
収集

データ
整形

モデル
生成

モデル
評価

チューニング

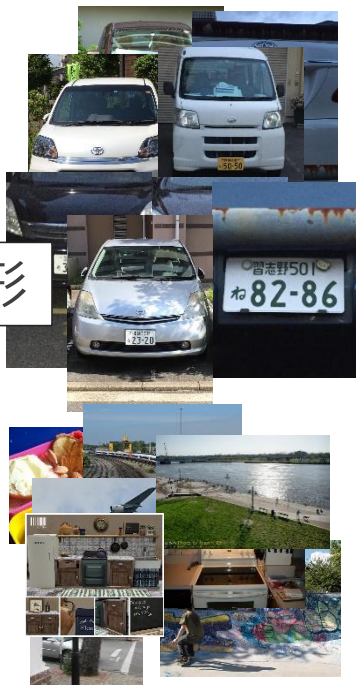
モデル開発：データ整形

□ 学習用にデータを整形

元データ



整形データ

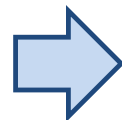


データ整形



画像形式、画像サイズ、
exifデータの削除等

ラベリング



ツールを利用し、
ナンバープレートの
位置をラベリング

教師データ（画像＋座標）



(12, 65, 45, 32)



(22, 35, 21, 18)



(19, 27, 22, 20)



(20, 26, 23, 19)

...

データ
収集

データ
整形

モデル
生成

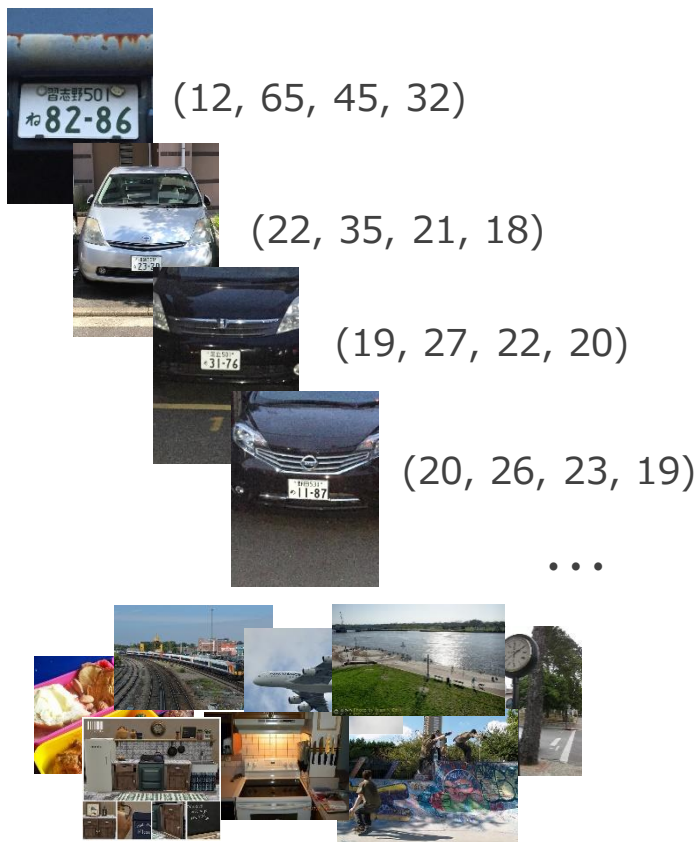
モデル
評価

チューン
グ

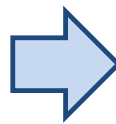
モデル開発：モデル生成

□ 学習用データを投入し、モデルを作成する

学習用画像



投入



学習



モデル

OpenCVの標準学習機能
でカスケード分類器を作成

データ
収集

データ
整形

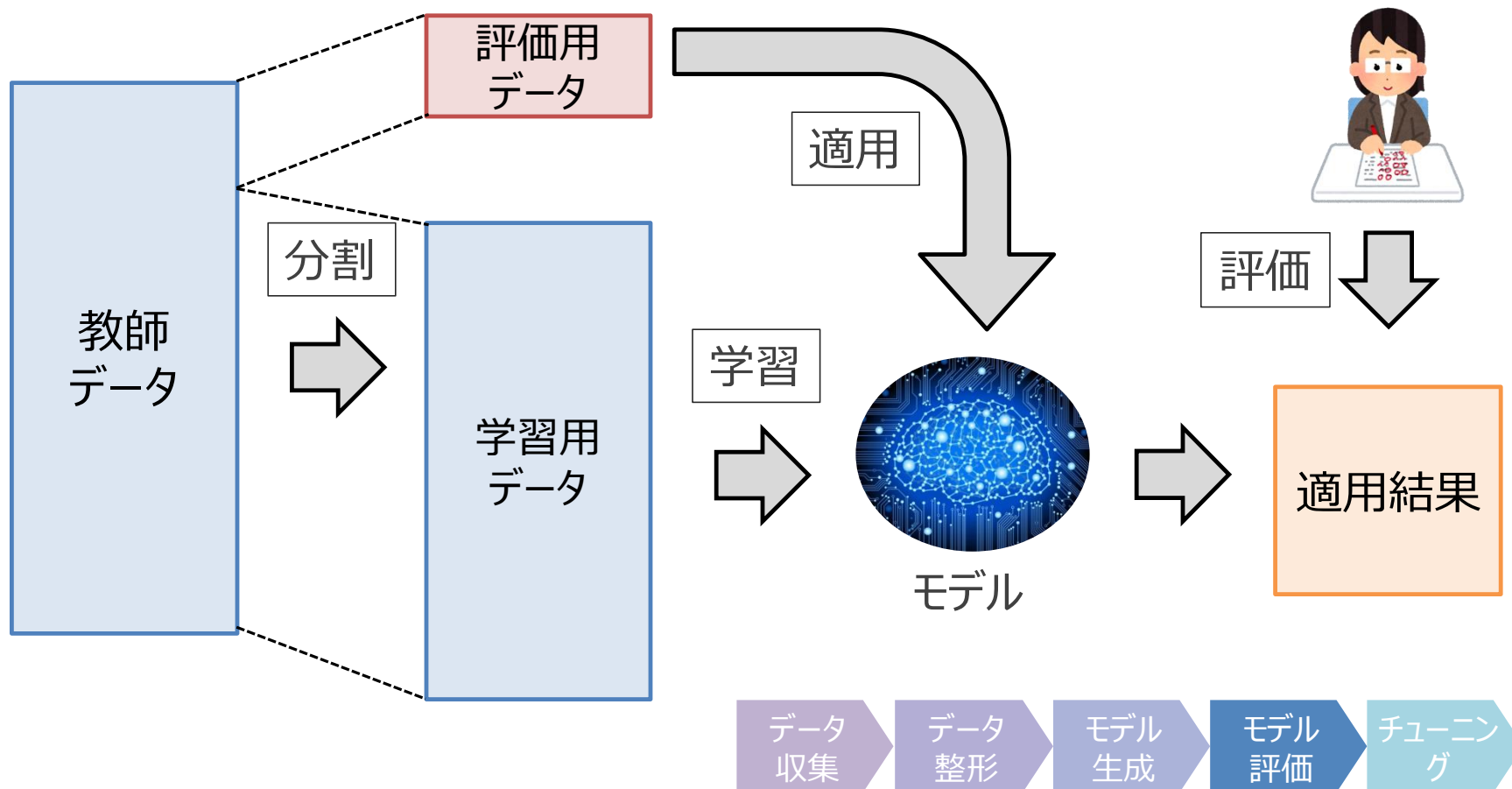
モデル
生成

モデル
評価

チューニン
グ

モデル開発：モデルの評価

□ テストセット法で評価



モデル開発：モデル評価

□ 評価基準

テストデータ



モデル
適用



適用結果



| | 検知成功 | 未検知 | 誤検知 |
|-----------|------|-----|-----|
| 結果 (n=50) | 27枚 | 23枚 | 3枚 |
| 割合 | 54% | 46% | 6% |

データ
収集

データ
整形

モデル
生成

モデル
評価

チューニン
グ

モデル開発：チューニング

- 3回のチューニングを実施し、モデルの精度を向上

| 実施項目 | 事象 | チューニング内容 | 効果 |
|-----------------|----------------|------------------------------|------------|
| 正解データ 増強 | 検知率が 著しく低い | 正解データ数： 100枚→976枚 | 検知率 ↑ ↑ |
| 不正解データ 導入 | 誤検知率が 著しく高い | ナンバープレートが映ってい ない不正解データを準備 | 誤検知率 ↓↓ |
| パラメータ チューニング | 検知率が悪い | 学習用のパラメータを調整 | 検知率 ↑ |

データ
収集

データ
整形

モデル
生成

モデル
評価

チュー
ニング

業務システムへの組み込み

□ 作成したモデルを業務システムへ適用し、動作確認

• 動作確認環境

| OS | CPU | メモリ | フレーム幅 | フレーム高 | フレーム率 | 動画時間 |
|-----------|---------------------|------|-------|-------|----------------|-------|
| Windows10 | Intel Core i7 7700K | 16GB | 1920 | 1080 | 30 (フレーム/秒) | 6 (秒) |

□ 確認結果

| | |
|--------------|-----|
| 動画取り込み機能 | OK |
| 画像変換機能 | OK |
| ナンバープレート検知機能 | OK |
| モザイク処理機能 | OK |
| 動画変換機能 | OK |
| 処理時間（5回平均） | 26秒 |

6秒の動画の
モザイク処理に
26秒！

処理時間内訳

□ 画像 1 枚当たりの処理時間の内訳（秒）

| | |
|------------|---------------|
| グレースケール変換 | 0.0017 |
| ナンバープレート検出 | 0.6497 |
| モザイク処理 | 0.0123 |
| 次フレーム読み込み | 0.0043 |
| 全体 | 0.06735 |

ナンバープレートの検知に時間がかかる

性能改善のために

□ 圧縮機能を追加



□ 画像圧縮率と処理時間

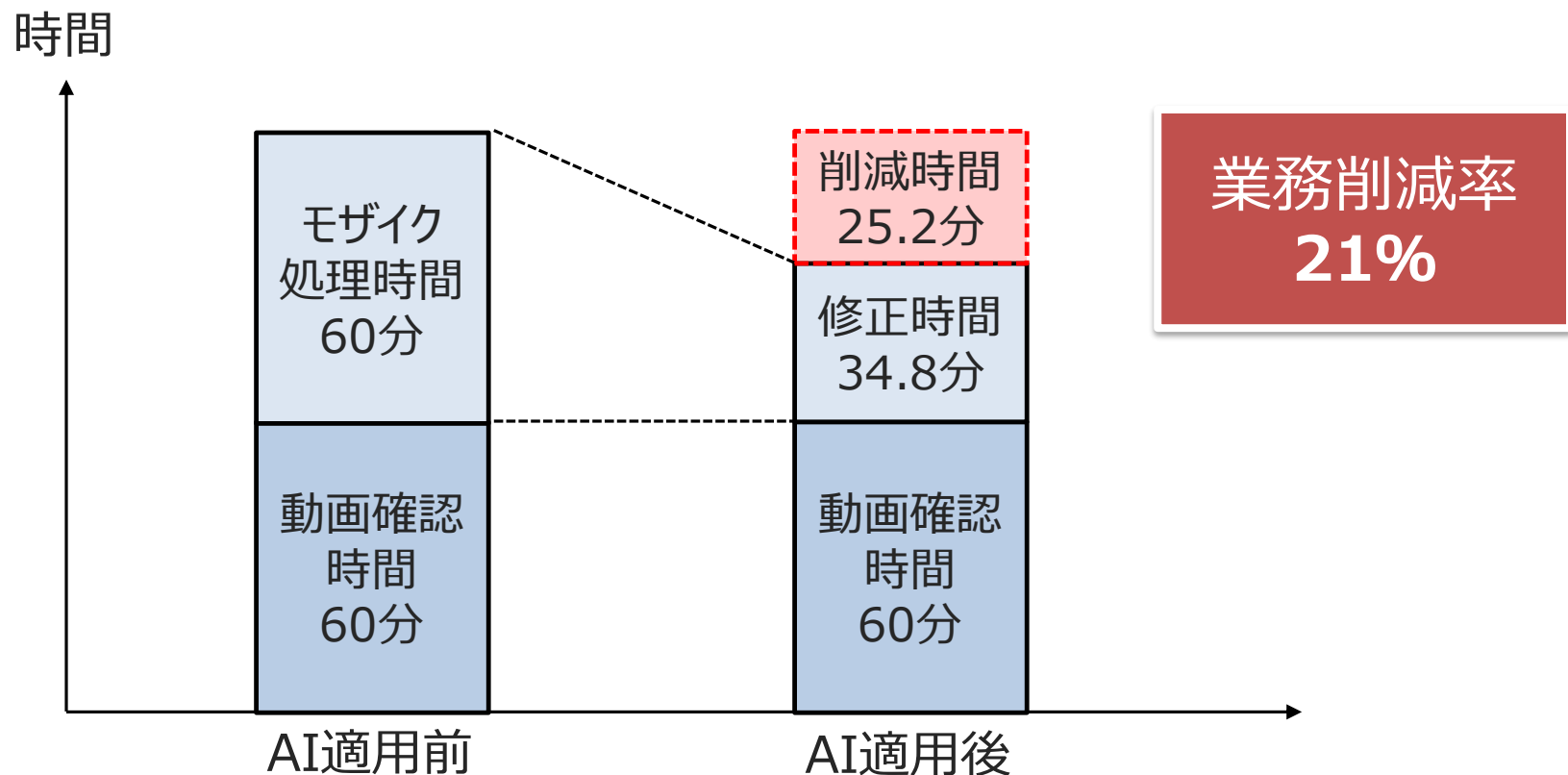
1/4～1/8で
動画時間より短くなる

| 圧縮率 | 1/1 | 1/2 | 1/4 | 1/8 |
|--------|------|-----|-----|-----|
| 処理時（秒） | 26.0 | 9.5 | 5.2 | 3.8 |

※ただし、圧縮率を上げるほど検知率は下がる

KPI評価

- ❑ 動画時間：60分、モザイク処理対象時間30分、
人による1分あたりのモザイク処理時間：2分
と仮定した場合



KPI評価

□ 式で表すと...

作業削減時間合計

$$\left\{ \left(\text{検知率} \times \text{モザイク処理対象時間} \right) - \left(\text{誤検知率} \times \text{動画時間} \right) \right\}$$

× 人による 1 分あたりの
モザイク処理時間

□ 削減時間を増やすには、

- ① 検知率向上
- ② 誤検知率の削減
- ③ モザイク対象時間の増加

ケーススタディ考察

□ 向上の余地はまだまだある



精度向上

- ディープラーニングの導入
- アプリ側の工夫



処理性能

- GPUの導入
- リソース増強



利用シーン拡大

- 処理対象の拡大
- 生放送への適用

演習

- ① 自身の身の回りにある業務を列挙してみる
 - なるべく具体的な業務を列挙すること
 - モザイク処理、ビジネスマッチング、海外送金、カード不正利用監視、システムログ監視、など
 - 最低でも 2 ～ 3 個列挙すること
 - ヒント：
 - 自分が担当しているシステムが行っている機能を列挙し、人の手が残っているものを探す
 - Web検索も活用可能

演習

- ② 「対象業務選定」の観点に従って、列挙した業務のAI導入可能性を検討する
- データ収集、実現性、精度対効果、KPI設定の各項目について検討すること（3段階評価）
 - ヒント：
 - データ収集
 - オープンデータや公開データがないか探してみる
 - 実現性
 - 類似事例がないか探してみる
 - 精度対効果
 - 人とAIの役割分担をフロー化してみる
 - KPI設定
 - 精度ではなく、業務上のKPIを設定する

演習

③ ①②の検討結果をまとめ、提出する

■ まとめ方は自由

- テキストベース、Excel、PowerPoint、etc

■ 以下の項目を含めること

1. 列挙した業務とその概要
2. データ収集、実現性、精度対効果、KPI設定の各項目についての3段階評価
3. なぜその評価なのかの根拠

■ 提出方法は以下の通り

- 提出先（Teams）

- ファイル > AI概要_提出用フォルダ > AI概論演習

- ファイル名

- AI概論演習_名前.拡張子

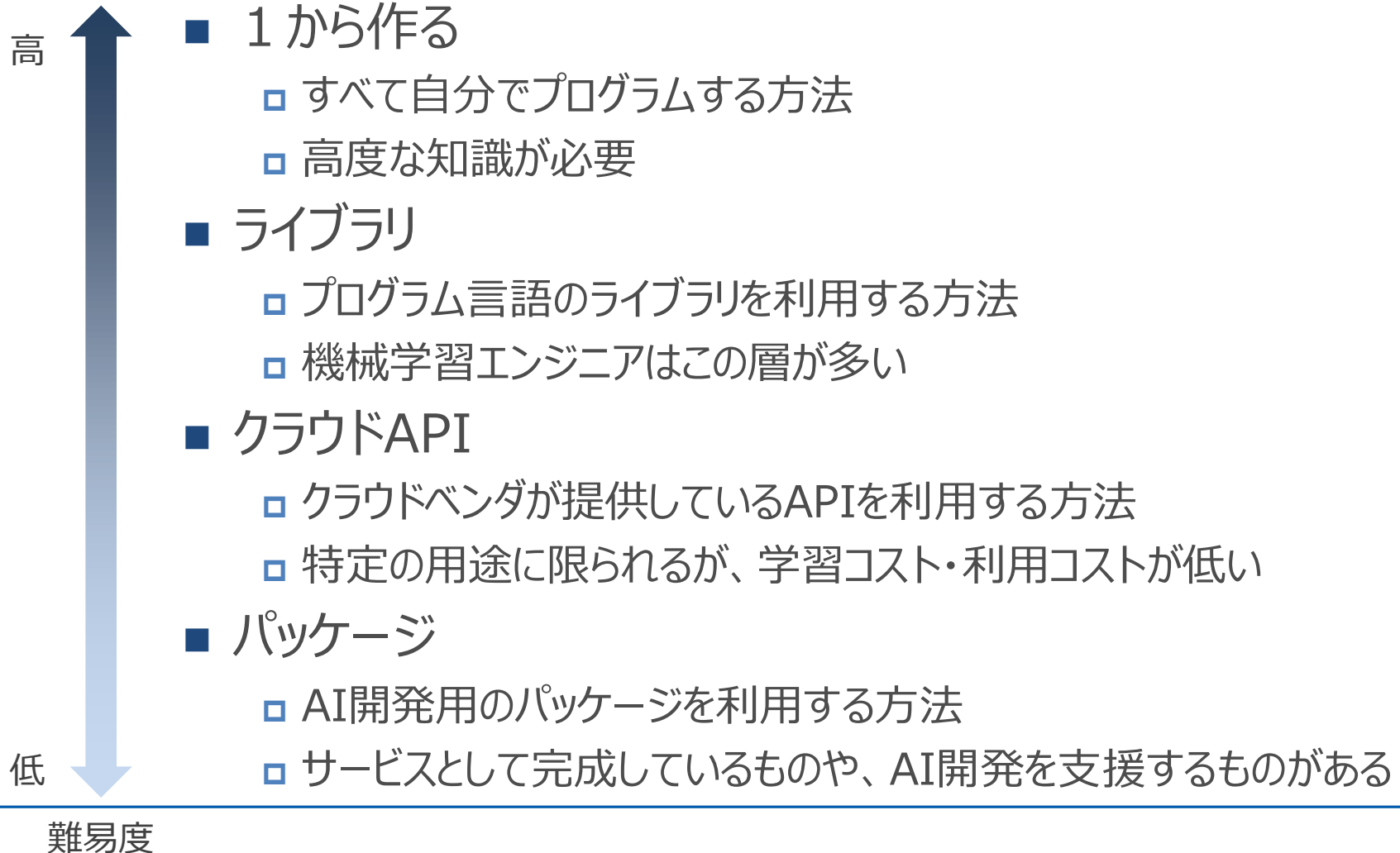


目次

- AIとは
- 機械学習とは
- AI開発フロー
- ツール・ライブラリ
- まとめ

AI開発ツール

□ AI開発の方法は、4つのレベルに分けられる



代表的な機械学習ライブラリ

| 名称 | 言語 | 開発元 | 備考 |
|--------------|--------------------------|---|---|
| TensorFlow | C Java Python | Google | Googleが開発しオープンソースで公開している、機械学習に用いるためのソフトウェアライブラリ |
| Keras | Python | OSS | オープンソースニューラルネットワークライブラリ |
| PyTorch | C++ Python | Facebook | オープンソースの機械学習ライブラリ |
| Spark MLlib | Java Python Scala | Apache | Spark環境下で動作する機械学習ライブラリ 分散処理、リアルタイム処理が可能 |
| libsvm | C++ Python Java | Preferred Networks、 NTTソフトウェア イノベーションセンター | 分散処理、リアルタイム処理が可能 |
| Scikit-learn | Python | David Cournapeau | Pythonで動作するOSS機械学習ライブラリ デファクトスタンダード的存在 |
| OpenCV | C, C++ Java Python | Intel | コンピュータ・ビジョン向けのライブラリ 画像処理・画像解析および機械学習等の機能を持つ |
| gensim | Python | RARE Technologies Ltd | トピックモデリングと自然言語処理のためのオープン ソースライブラリ |

代表的なクラウドAPI (1/2)

| クラウド | サービス | 概要 |
|------|--------------------|---|
| AWS | Amazon personalize | Amazonで実際に使われているレコメンデーションの機能を提供 |
| | Amazon Transcribe | 音声テキストに変換する機能を提供 |
| | Amazon rekognition | 画像・動画の分析サービス 物体検知やテキスト検出、顔の検出・分析等が可能 |
| | Amazon comprehend | テキスト分析サービス 感情分析や構文解析、トピックモデリング等が可能 |
| | Amazon Lex | 音声やテキストを使用して、任意のアプリケーションに対話型インターフェイスを構築するサービス |
| | Amazon textract | 電子化したドキュメントからテキストとデータを自動抽出するサービス |
| | Amazon Polly | 文章をリアルな音声に変換するサービス |
| | Amazon Translate | 高速で高品質な言語翻訳を提供するニューラル機械翻訳サービス |

代表的なクラウドAPI (2/2)

| クラウド | | |
|-------|----------------------|---|
| GCP | Cloud Vision API | 画像検出機能を提供するサービス 画像ラベリング、顔やランドマークの検出、OCR等が可能 |
| | Cloud Speech API | 音声をテキストに変換するサービス |
| | Natural Language API | 非構造化テキストの分析サービス 感情分析、エンティティ分析、コンテンツ分類、構文分析等が可能 |
| | Translation API | 言語翻訳サービス |
| | Dialogflow API | 会話型インタフェース（チャットボット、音声アプリケーションなど）のための開発ツール |
| Azure | 画像処理 | 顔の検出や識別・分析、画像ラベリング、ランドマークの検出、OCR等が可能 |
| | 音声 | 音声をテキストに変換・テキストから音声を生成、話者の識別・検証等が可能 |
| | 検索 | 画像検索やニュース検索、スペルチェック等、検索エンジンで提供されている機能を使用可能 |
| | 言語 | 自然言語の意味解釈や、重要な情報の抽出等が可能 |

代表的なパッケージソフト



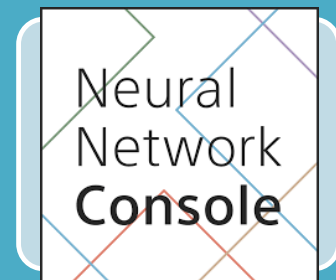
SAS Viya

- 統計解析ソフトウェアのSAS社が、AI・機械学習分野に対して提供しているソフトウェア
- データのロードからデータの確認、前処理、モデル作成・学習、モデルの評価まで、視覚的な操作で実行可能



DataRobot

- DataRobot社が提供する機械学習自動化プラットフォーム
- 超高精度の予測モデルを生成する機械学習の自動化、有用なデータ項目や予測の根拠を可視化する予測モデルのグレーボックス化、システム連携が容易の容易さが特長



Sony Neural Network Console

- SONYが提供するディープラーニングの開発基盤
- ドラッグ & ドロップでの簡単操作、構造自動探索、高速な学習と評価、学習履歴の集中管理が特長



目次

- AIとは
- 機械学習とは
- AI開発フロー
- ツール・ライブラリ
- まとめ

まとめ

- AIとは、人間の知的活動を再現する機械・システム
- 学習とは、「分ける」処理
 - 機械学習では、「分け方」をコンピュータに学ばせる
- AI開発では、
 - AIと人間の役割分担を明確にすること
 - KPIを適切に設定し、運用していくことが重要である