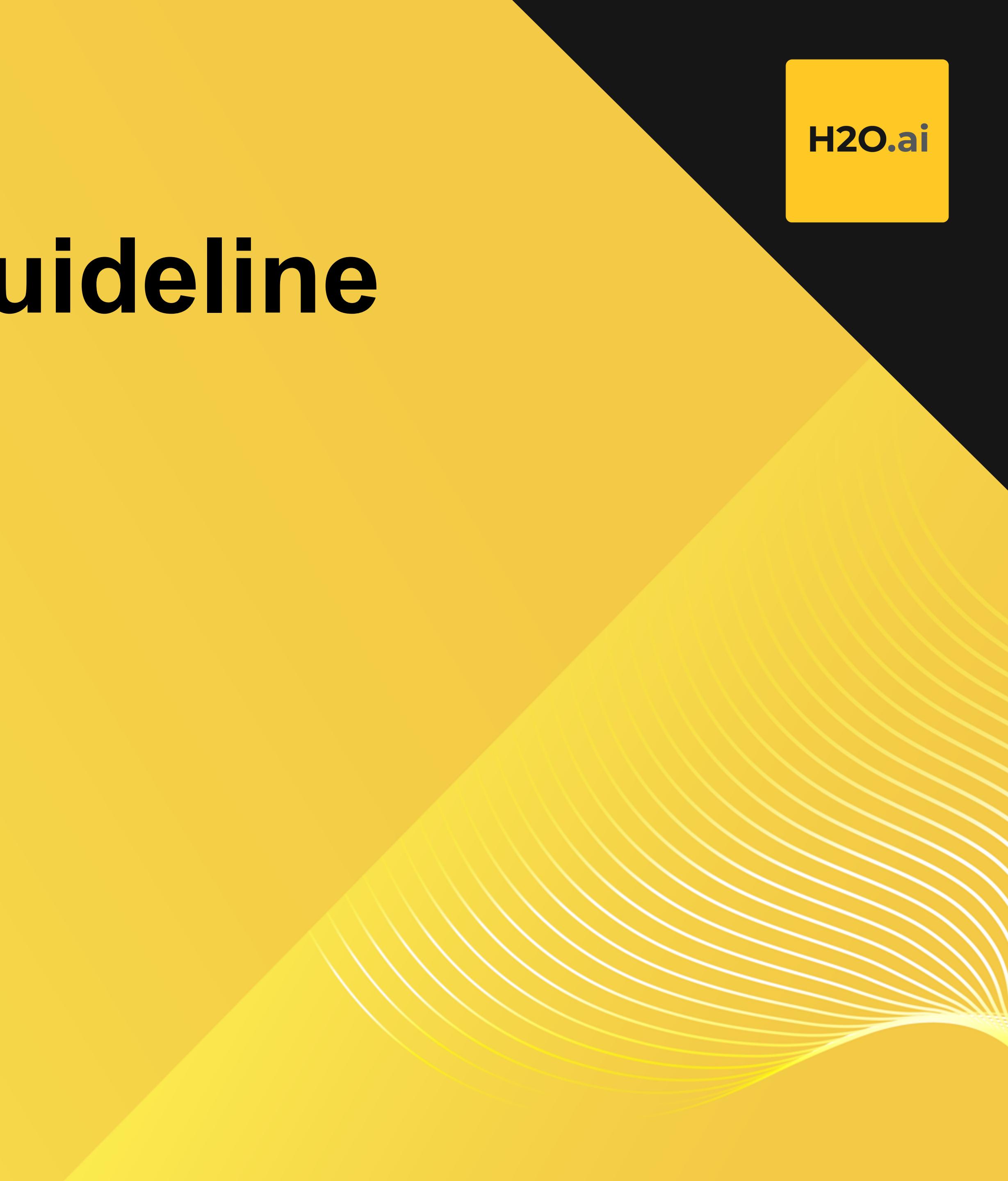


Data Analysis Guideline

製造業編



Agenda

H2O.ai

機械学習とそのユースケース

機械学習プロジェクトの実施

ツールとしてのAutoML

Case Study – 予知保全/Predictive Maintenance

機械学習と そのユースケース

機械学習とは

機械学習は、データから予測のための数学的なルールを、アルゴリズムに学習させる技術

データからパターンを学習し、予測などに利用

用途

機械学習（Deep Learning除く）

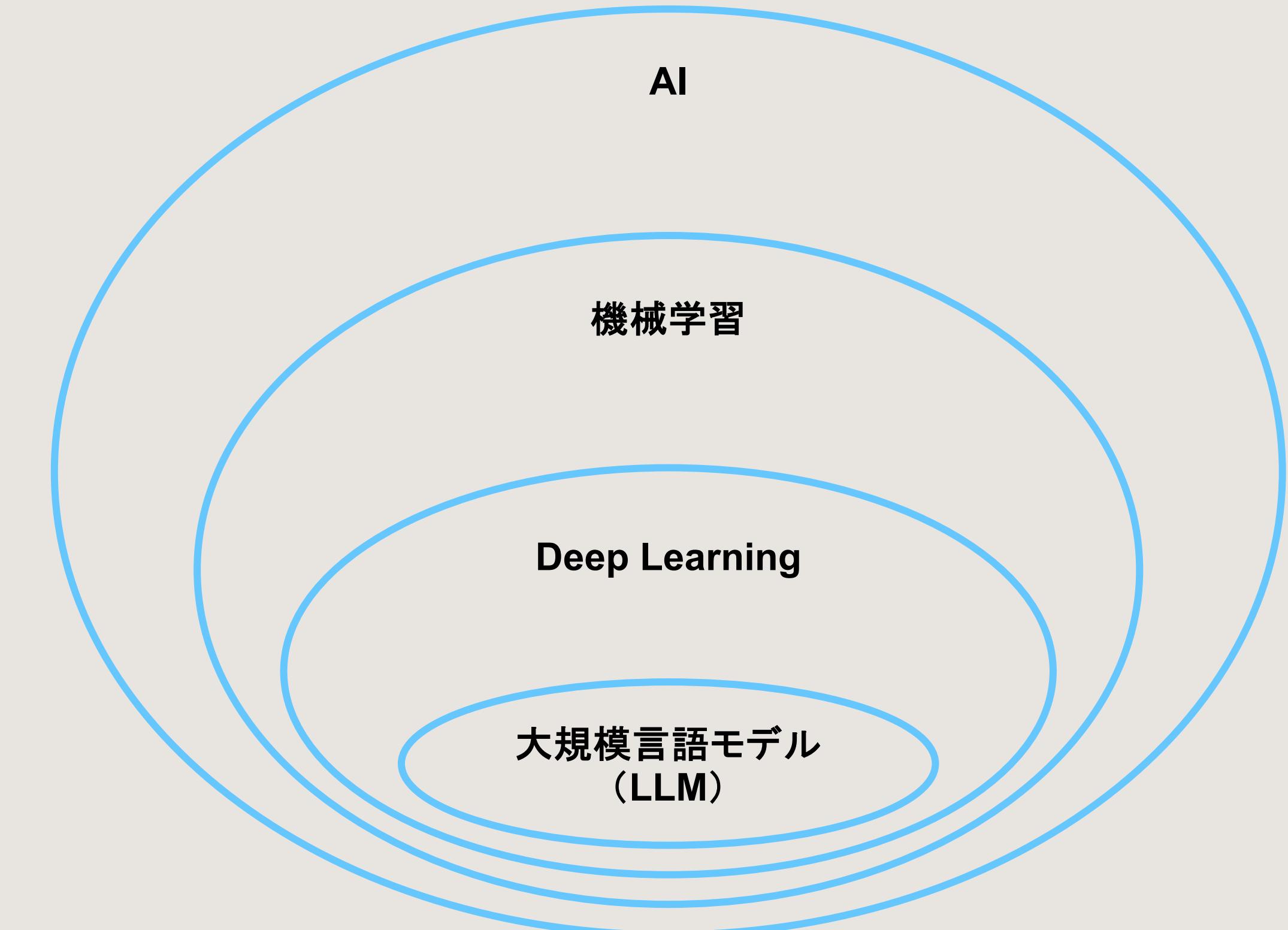
- 構造化データ（データベース）や半構造化データ（センサーログ）を用いた予測

Deep Learning（大規模言語モデル除く）

- 非構造化データ（画像、テキスト、音声）を用いた予測

大規模言語モデル

- 文章の生成を目的とする場合



- 機械学習はAI技術の一部
- Deep LearningはNeural Networkと呼ばれる機械学習アルゴリズム
- 大規模言語モデルは、自然言語を扱う大規模なDeep Learningモデル

機械学習と教師あり学習

機械学習は、教師あり学習、教師なし学習、強化学習に分類される

教師あり学習では、正解情報を含むデータから、その正解を予測するモデルを作成

教師あり学習は予測対象（目的変数のデータ型）によって、回帰と分類に分かれる

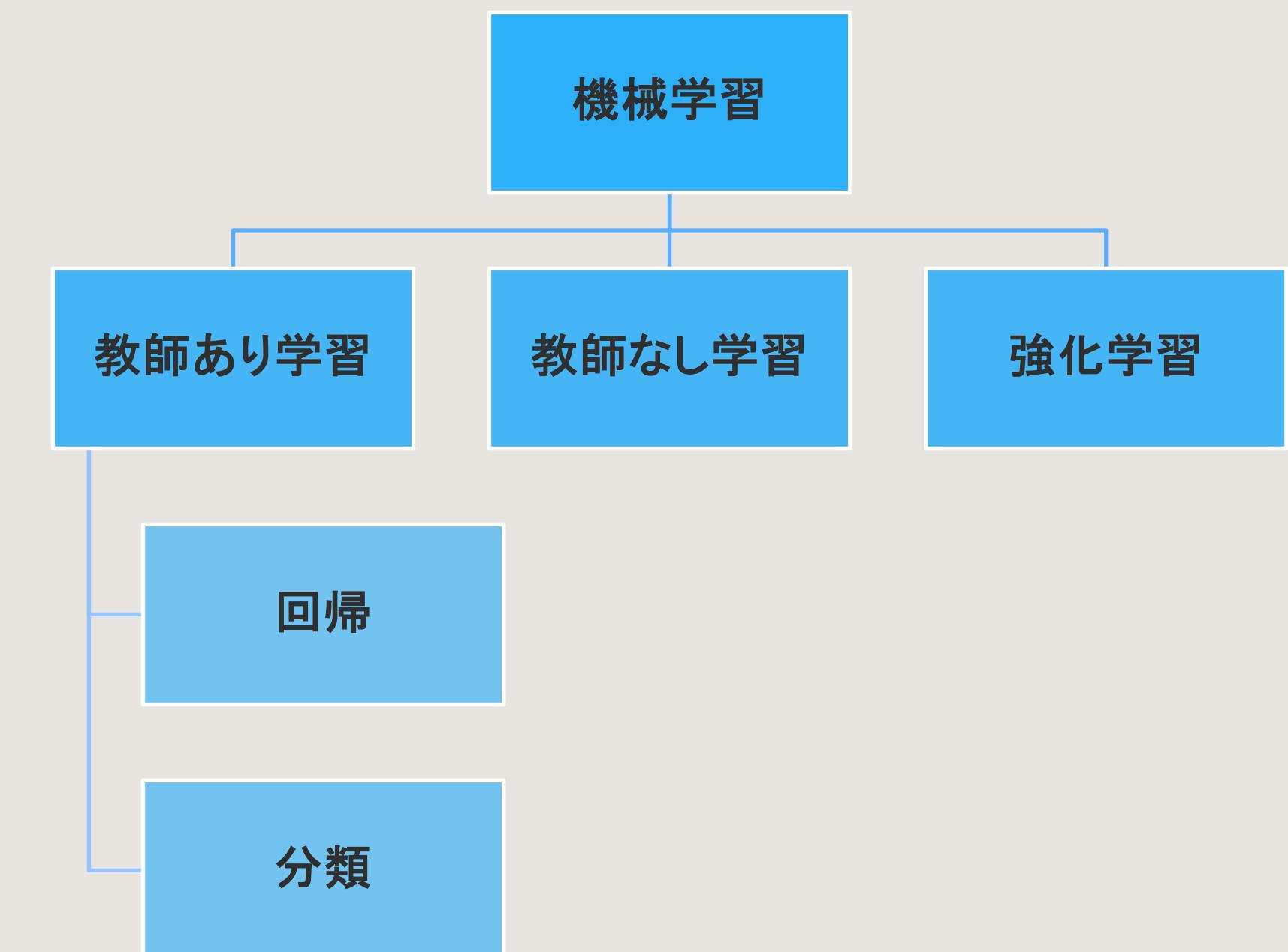
教師あり学習は正解が定義できるので、実務上取り組みやすい特徴がある

“回帰”（目的変数が数値）の例

- 製造条件のデータから、製品特性値（数値で表現）を予測
- 来月の製品需要（製品の出荷数）を予測

“分類”（目的変数がカテゴリ）の例

- 製品の外観画像から、良品か不良品かを分類
- 顧客セグメント（優良、通常、休眠予備軍など）の予測



製造業におけるユースケース

サプライチェーン

- 需要予測
- 在庫最適化
- 配送最適化

予知保全

- 不具合検知
- 寿命予測

品質管理

- 不良品検知
- 工程管理

製品開発

- マテリアルズインフォマティクス
- CAE代理モデル

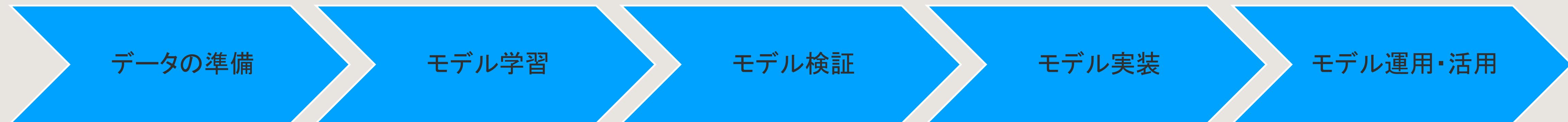
機械学習モデルを以下のようなことを目的として活用し、開発、生産、オペレーションを効率化する

- 人手で作業している業務を、機械学習モデルによる予測でサポート（判断精度の向上や省力化）
- 製造方法から、製品の特性値を機械学習モデルで予測することによって開発を効率化

機械学習プロジェクトの実施

機械学習モデルの開発（AI開発）ステップ

H2O.ai



モデル学習用データの準備

- 生データからのETL
- 前処理
- データの確認 (EDA)

予測モデルの作成

- 特徴量エンジニアリング
- モデル選択
- パラメータチューニング

作成した予測モデルの検証（ビジネス適用可能かの判断）

- 予測精度の確認
- モデルの解釈

予測モデルの利用準備

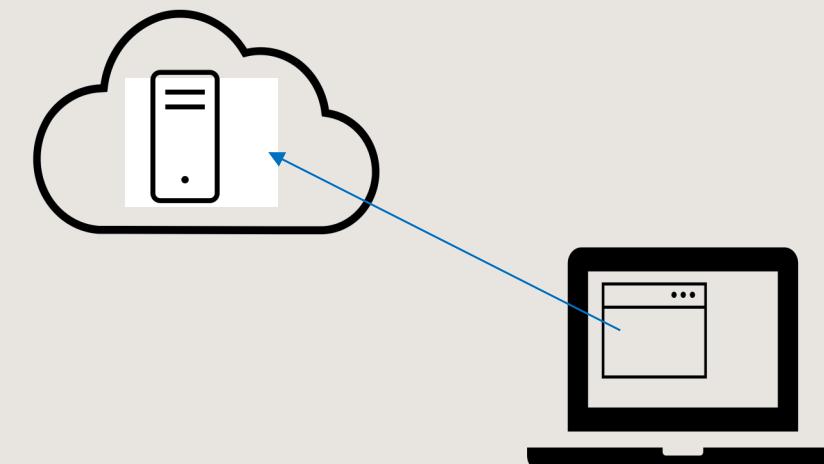
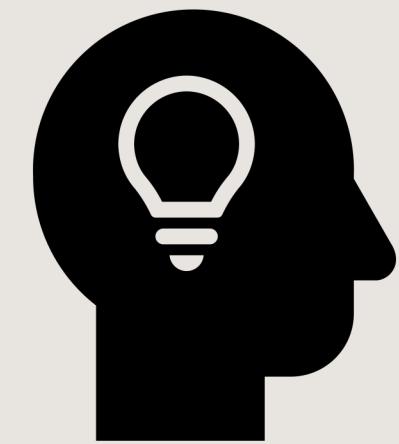
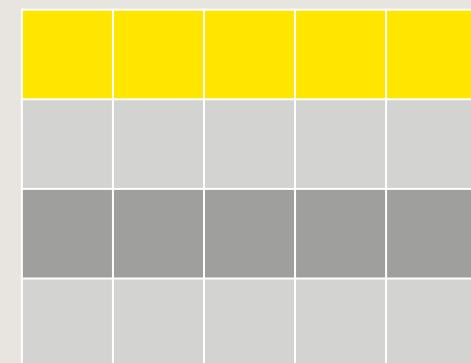
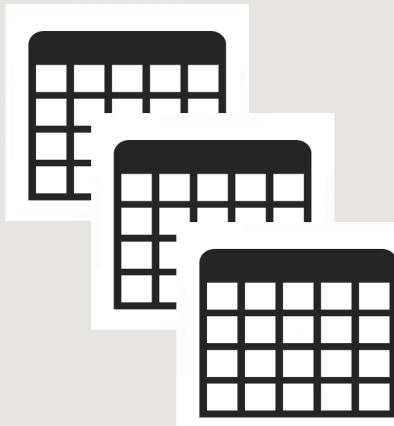
- スコアリング実行モジュールの作成
- スコアリングAPIの作成

予測モデルの運用 (MLOps)

- モデルのモニタリング
- モデルの管理
- モデルの再学習

モデルの活用

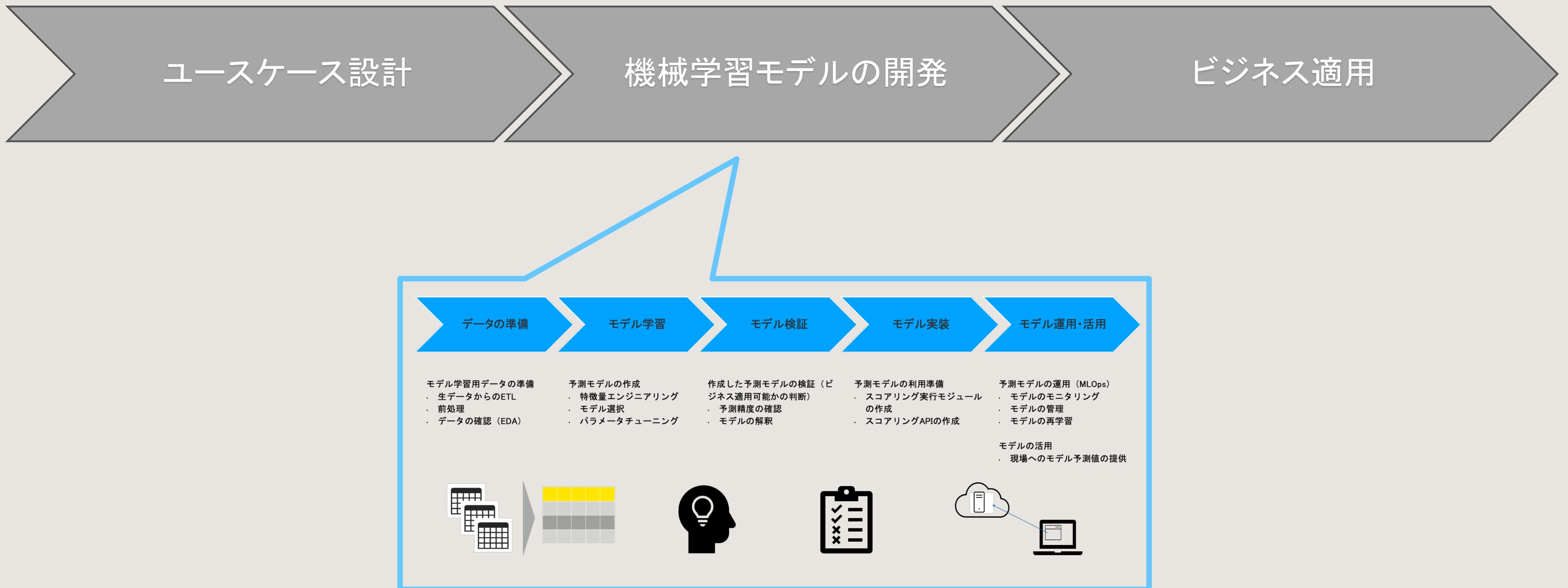
- 現場へのモデル予測値の提供



機械学習プロジェクト

H2O.ai

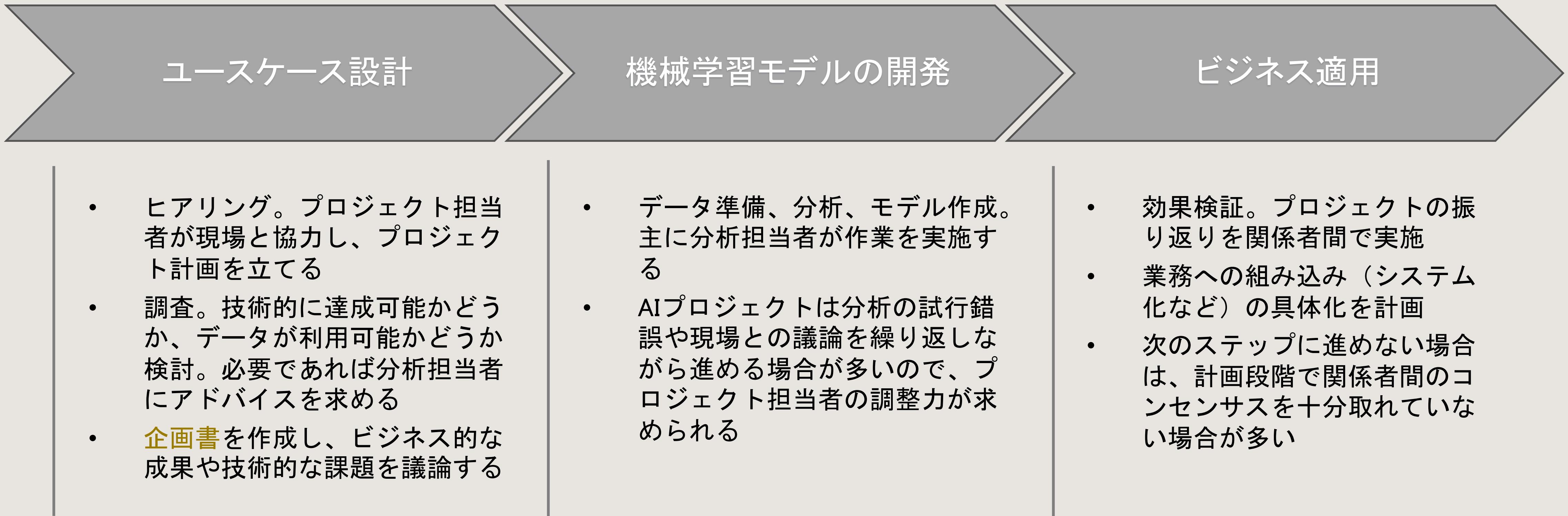
- AI開発はプロジェクトの一部
- ユースケース設計とビジネス適用はAIではなく、人間が主体



- ツールの利用によって作業負担の軽減が実施しやすい

機械学習プロジェクト

H2O.ai



以下の項目を具体的に書き出してみる

1	テーマ	・取り組み(分析、AIモデル作成)のテーマ
2	業務の課題(AS-IS)	・現在の業務課題
3	AI導入後の業務の姿(TO-BE)	・分析結果の活用、AIの導入により業務がどのように変わるか ・活用、導入イメージが現場と共有されているか確認
4	期待効果	・業務における定量的効果(削減金額や量、工数など)
5	利用データ	・利用可能、利用予定のデータをリストアップ ・必須にものオプショナルなものを整理 ・データのオーナー
6	アプローチ	・教師あり学習のアプローチで対処することが望ましい
7	KPIと予測対象	・何を予測の目的変数とするか ・”4.期待効果“へ関連する必要あり
8	目標精度	・”7.KPIと予測対象”がどの程度の予測精度を達成できれば良いか仮決めする
9	体制	・実施に必要な担当者が揃っているか確認
10	スケジュール	・データ準備期間や分析、モデル作成期間のスケジュールを作成 ・データ分析PJはアジャイルで進める場合が多いので、厳密なスケジュールの計画は難しい場合が多い

企画書（例）

予知保全/Predictive Maintenanceでの例

1	テーマ	エンジンの交換時期予測による、作業コスト削減
2	業務の課題(AS-IS)	エンジンの交換判断にはベテラン整備士の知見を必要としている。毎回の確認にそれら人手をかけて実施するのに大きなコストがかかっている
3	AI導入後の業務の姿(TO-BE)	エンジンの交換時期を予測することにより、交換の確認作業の省力化を実施(明らかに大丈夫な場合は確認作業を簡略化する)したい
4	期待効果	交換確認作業コストの削減。また、安全管理(どれくらいで交換が必要となるか作業員に注意喚起を実施)などにも有用と考えられる ただし、安全管理に関わる内容なので、リスクに関する考慮する必要がある
5	利用データ	エンジンの各種センサー値を運用開始から交換時期まで時系列で記録したデータ
6	アプローチ	運用開始後、特定時点における交換までの期間を予測(回帰問題) 予測期間算出後、どれくらい予測期間を基準とし、確認作業の実施有無を判断するか検討
7	KPIと予測対象	KPIは、確認作業の削減回数 予測結果を元に確認作業の実施有無のエンジンを判断。同時に、作業の削減によって発生するリスクの定量化を実施
8	目標精度	過去に予測を実施したことがなく現時点での目標精度はなし。コスト削減とリスクのバランスを考慮し、後ほど検討
9	体制	— 省略 —
10	スケジュール	— 省略 —

企画書（例）

H2O.ai

「予測対象」とはAIモデルの目的変数。数字で表現できなくてはならない

KPIと予測対象の作成の例

やりたいこと	KPI(ビジネス目線)	予測対象(分析目線)
交換確認作業コストの削減	確認作業の削減回数 作業の削減によって発生するリスクの定量的評価	特定時点における交換までの予測期間をセンサーデータを用い予測。予測結果を元に確認作業の実施有無を判断((例)予測時点より20期末満で交換が必要とされるエンジンは確認を実施し、それ以外は確認作業を実施しない) 確認が必要ないと判断されたエンジンのうち、実際は確認が必要だったエンジンが安全管理上のリスクとなり得る

* 予測対象は分析や議論を繰り返し、試行錯誤で決めていく場合も多い

AIモデルによって予測対象を予測した結果にアクション可能かどうかも重要な判断材料（予測はできるが使い所がないモデルは、役に立たない場合が多い）

手持ちのデータの利用法から考えるのでなく、出口（AI導入後の業務の姿）から考えることも重要

ツールとしてのAutoML

機械学習モデルの開発ステップ

(再掲)

H2O.ai



モデル学習用データの準備

- 生データからのETL
- 前処理
- データの確認 (EDA)

予測モデルの作成

- 特徴量エンジニアリング
- モデル選択
- パラメータチューニング

モデル検証

- 作成した予測モデルの検証 (ビジネス適用可能かの判断)
・ 予測精度の確認
・ モデルの解釈

モデル実装

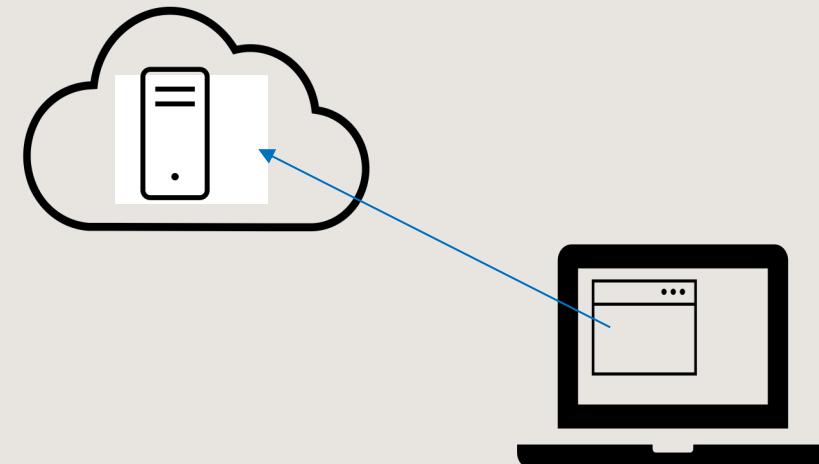
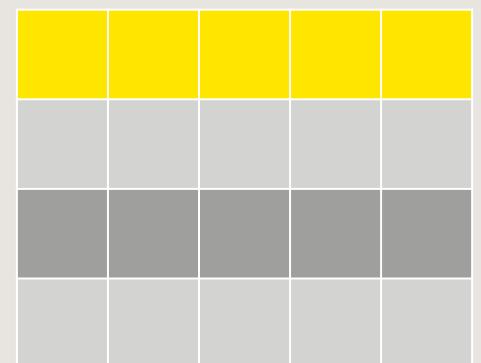
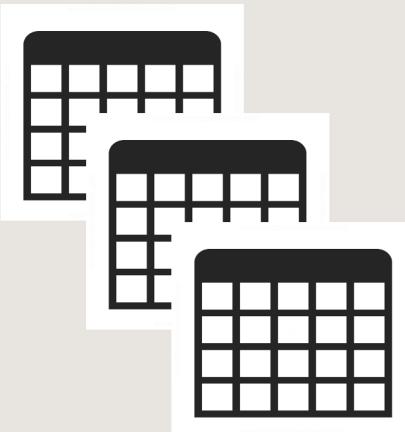
- 予測モデルの利用準備
・ スコアリング実行モジュールの作成
・ スコアリングAPIの作成

モデル運用・活用

- 予測モデルの運用 (MLOps)
・ モデルのモニタリング
・ モデルの管理
・ モデルの再学習

モデルの活用

- 現場へのモデル予測値の提供



AutoMLを用い、“機械学習モデルの開発ステップ”を省力化

H2O.ai

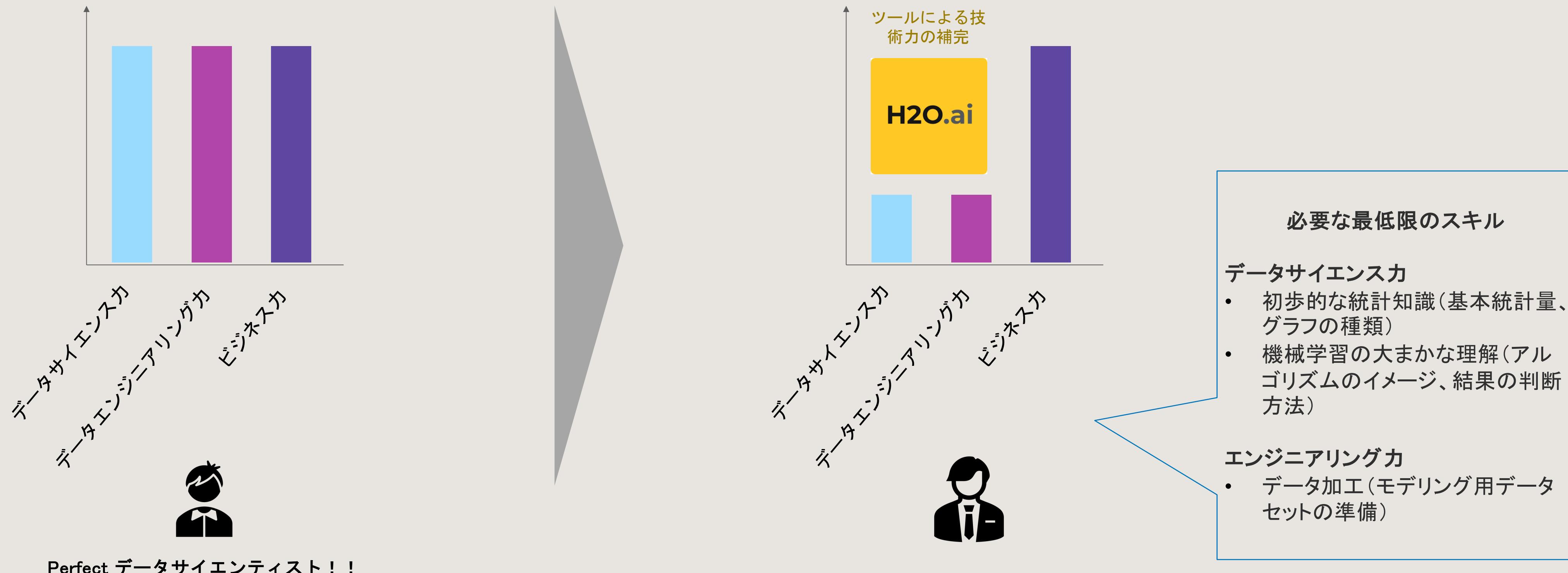


AutoML (Automated Machine Learning・自動化された機械学習) を利用することにより、機械学習モデル作成（モデル学習）を省力化。機械学習の専門家でなくても精度の高いモデルの作成が可能 よって、分析者はビジネス・研究の課題に集中することができる

ツールによるスキルセットの補完

H2O.ai

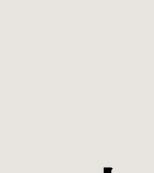
- ✓ AutoMLといったツールの発展により、技術領域の要求スキルを低くすることができる（もしくは、元々技術力の高いデータサイエンティストの時間効率化ができる）
- ✓ ツールでビジネス力の補完はできない



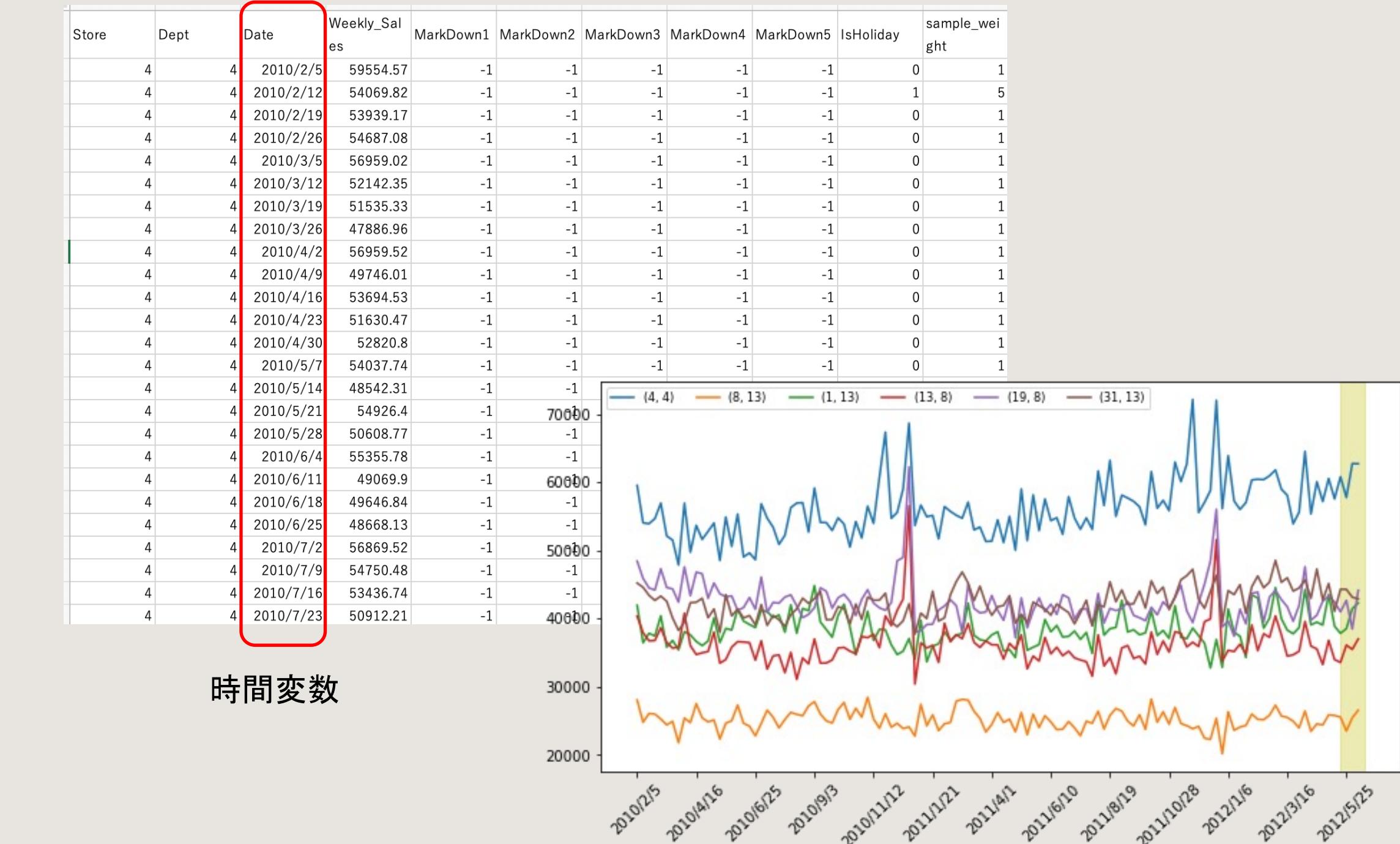
Driverless AIに投入するデータと対応する問題

H2O.ai

テーブル形式データ

co...	view	modality	date	location	folder	filename	doi	url	license	clinical_notes
PA	X-ray	January 22, 2020	Cho Ray Hospital, ...	images			... 10.1056/nejmco2001...	https://www.nejm...	On January 22, 20...	
PA	X-ray	January 25, 2020	Cho Ray Hospital, ...	images			... 10.1056/nejmco2001...	https://www.nejm...	On January 22, 20...	
PA	X-ray	January 27, 2020	Cho Ray Hospital, ...	images			... 10.1056/nejmco2001...	https://www.nejm...	On January 22, 20...	
PA	X-ray	January 28, 2020	Cho Ray Hospital, ...	images			... 10.1056/nejmco2001...	https://www.nejm...	On January 22, 20...	
PA	X-ray	January 25, 2020	Changhua Christia...	images			... 10.1056/NEJMco2001...	https://www.nejm...	diffuse infiltrates i...	
PA	X-ray	January 30, 2020	Changhua Christia...	images			... 10.1056/NEJMco2001...	https://www.nejm...	progressive diffuse...	
PA	X-ray	2017		images			... CC BY-SA	Severe ARDS. Pers...		

時系列データ

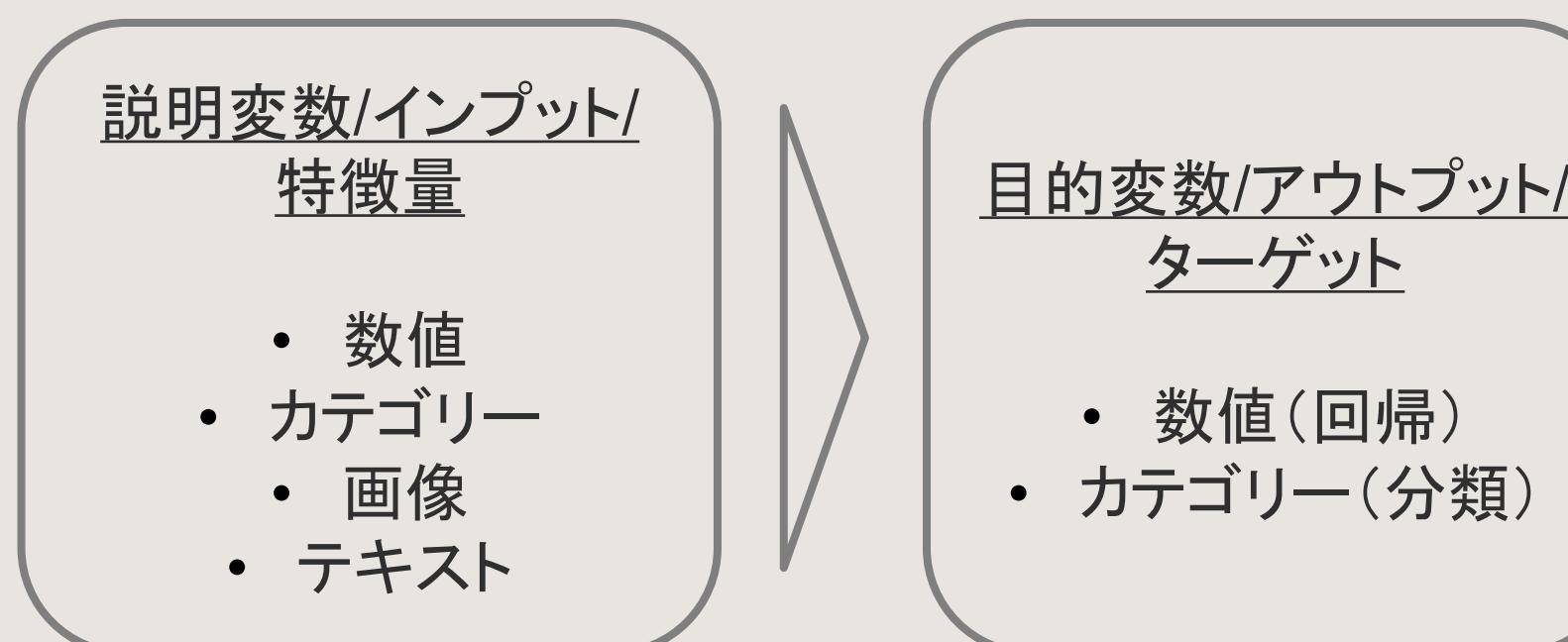


以下データをインプットとしての回帰/分類

- 数値/カテゴリー
- 画像（※ データでは画像ファイルへのパスを指定）
- テキスト（日本語などの文章）
- 一変量時系列の予測
- 多変量時系列の予測（多数の時系列を同時に予測）

データの例

問題	ユースケース	サンプルデータ	サンプルデータの説明変数データタイプ
回帰	予知保全:故障までの期間の予測を実施し、メンテナンスの効率化に利用	https://jp-public.s3.ap-southeast-1.amazonaws.com/data/RUL_Nasa/cmapss_rul_ws.csv (Small Sample) https://jp-public.s3.ap-southeast-1.amazonaws.com/data/RUL_Nasa/cmapss_rul_ws_MIN.csv	数値 カテゴリー
回帰(時系列)	需要予測:製品の出荷数を予測し、販売計画や在庫の最適化に利用	(Train data) https://jp-public.s3.ap-southeast-1.amazonaws.com/data/demand_forecast_walmart-fake/train_demand_forecast_fwm.csv (Test data) https://jp-public.s3.ap-southeast-1.amazonaws.com/data/demand_forecast_walmart-fake/test_demand_forecast_fwm.csv	数値
分類	異常検知:製品の異常を検知するシステム	(Small Sample) https://jp-public.s3.ap-southeast-1.amazonaws.com/data/Product_defect_classification/Product_defect_classification_SAMPLE.zip	画像
分類	文書分類:問い合わせなどの分類の自動化	https://jp-public.s3.ap-southeast-1.amazonaws.com/data/Livedoor_news/livedoor_news_v2.csv (Small Sample) https://jp-public.s3.ap-southeast-1.amazonaws.com/data/Livedoor_news/livedoor_news_v2_3CategorySample.csv	テキスト

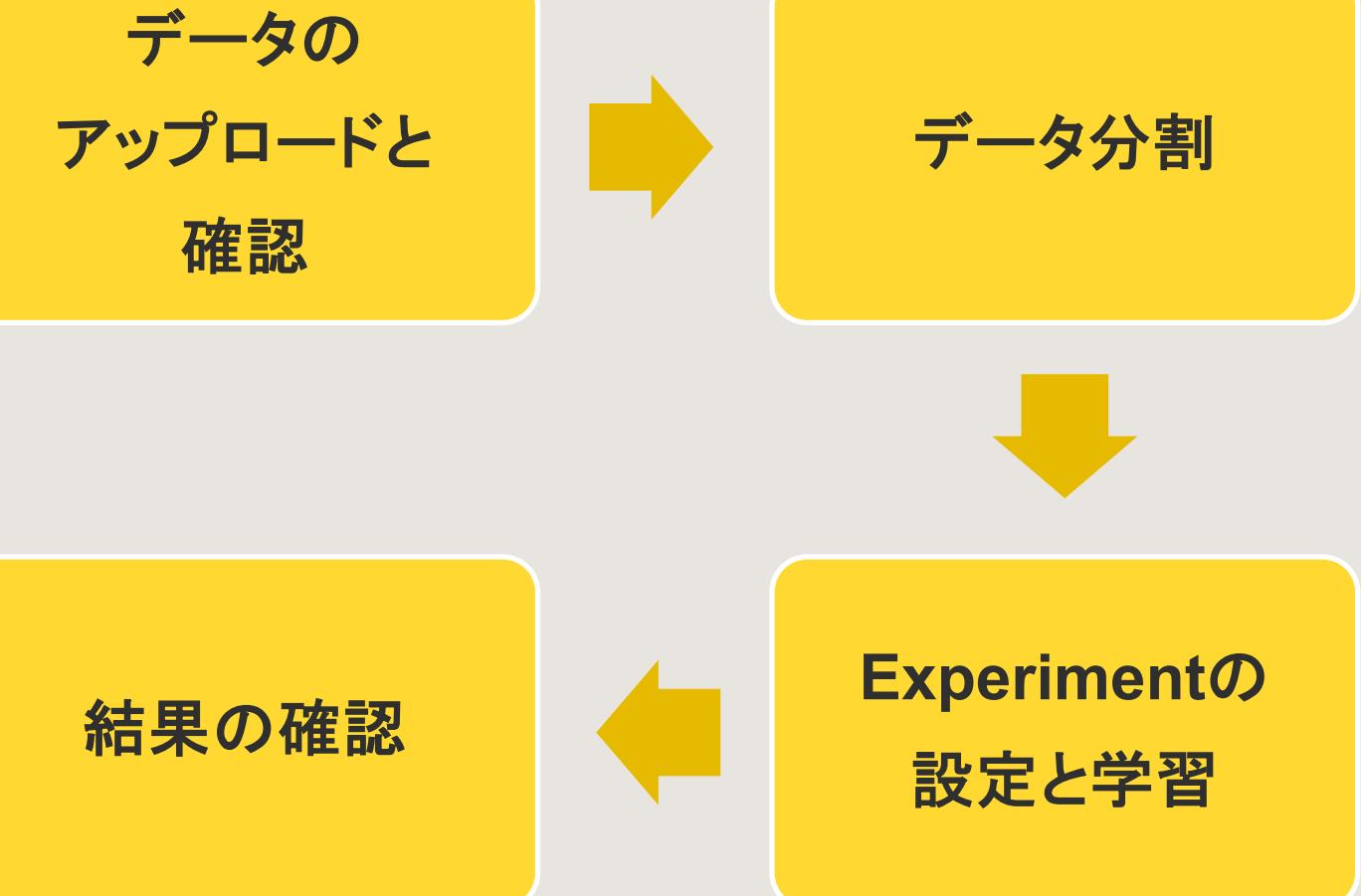


- 説明変数で目的変数を予測
- 説明変数側のデータは数値やカテゴリー型のデータの他に、画像やテキストも利用可能

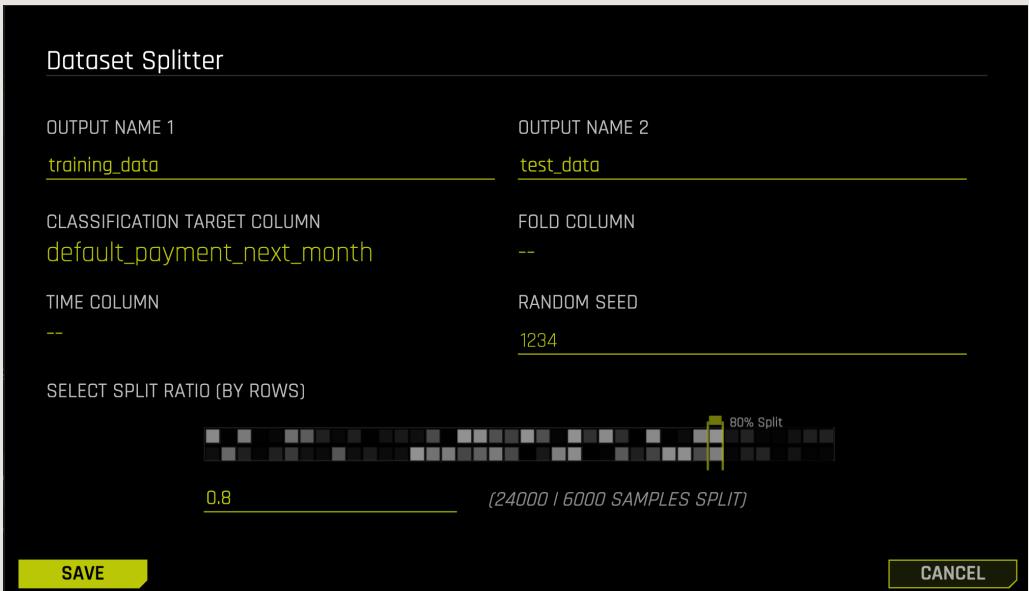
操作の流れ



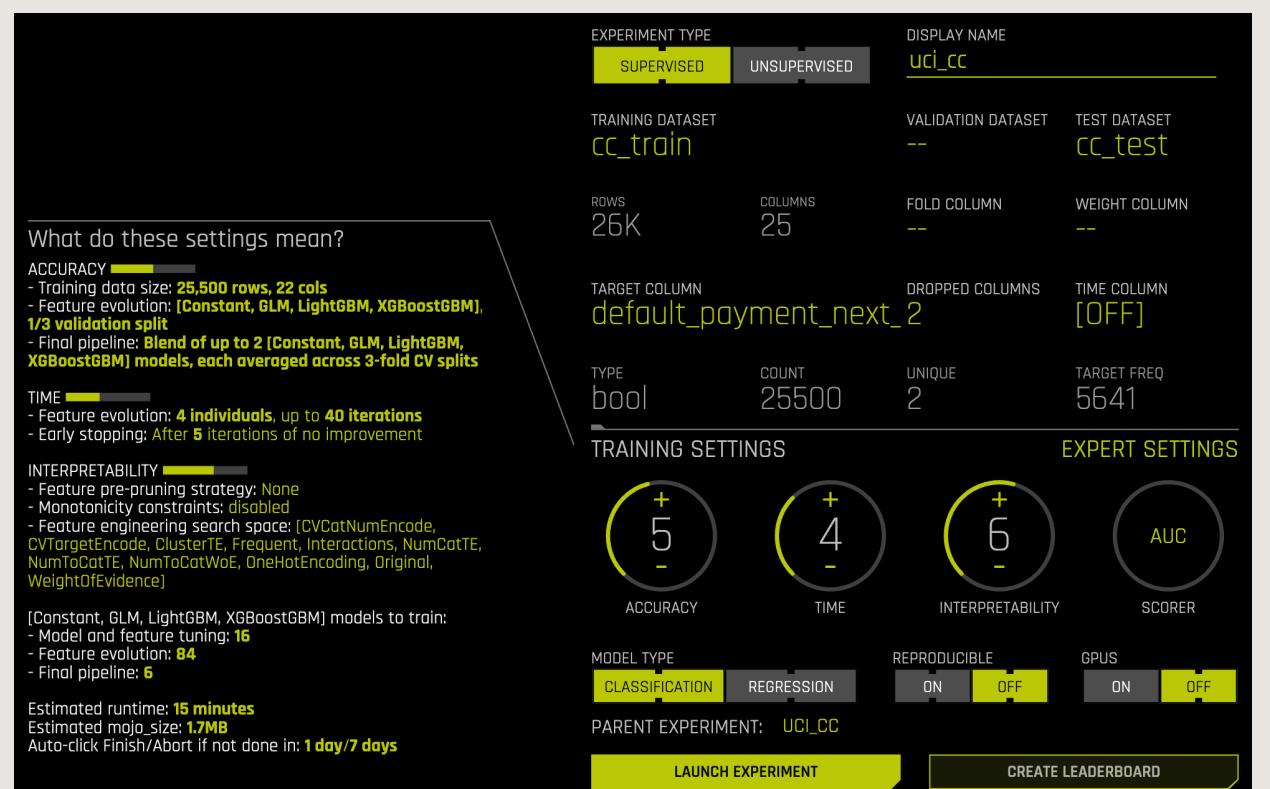
- データのアップロード（さまざまなストレージやDB連携も可能）
- 集計やプロットによるデータ確認



- 予測精度の確認
- モデルの理解（機械学習の解釈可能性）



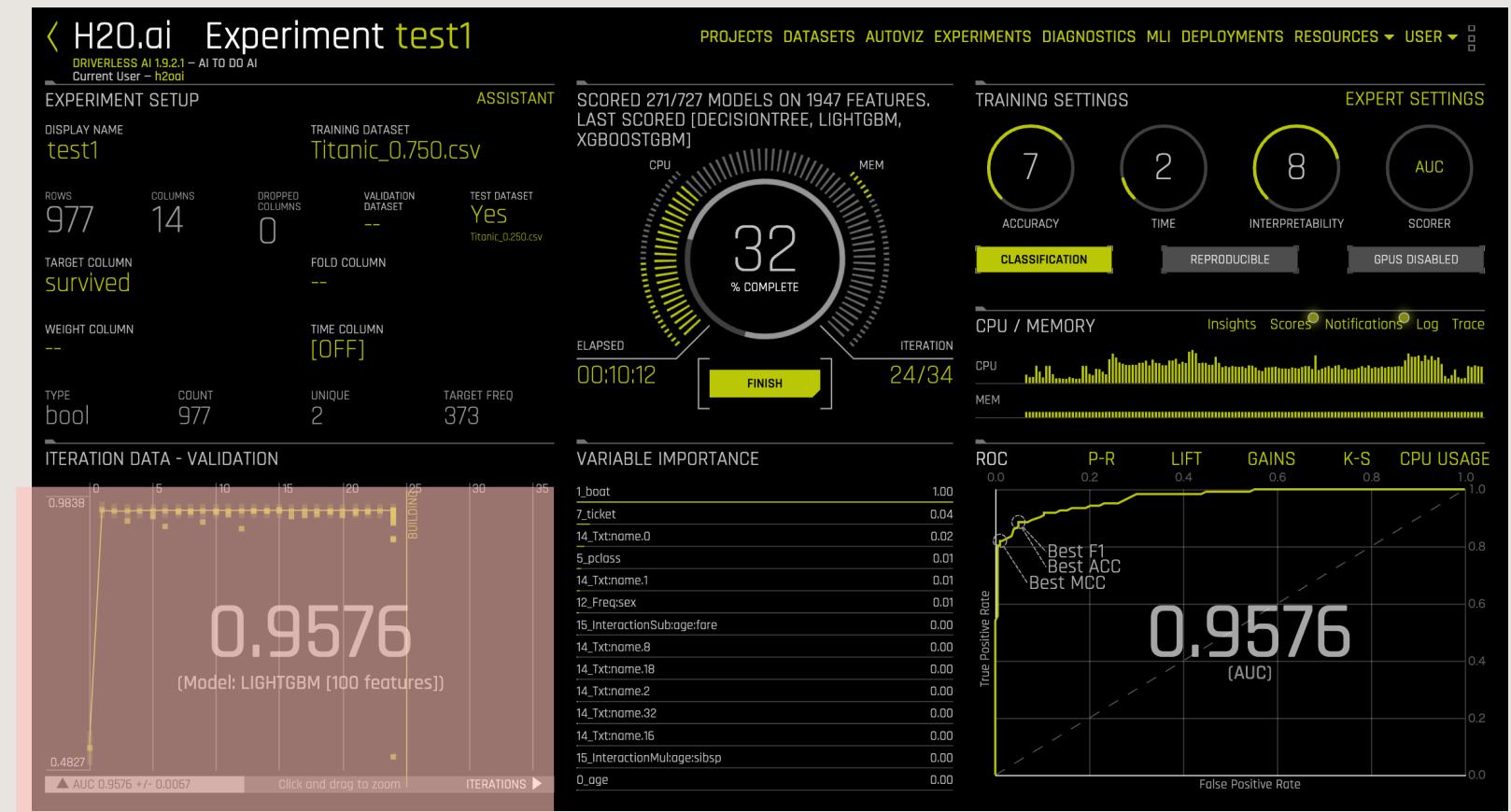
- 学習とテストデータへの分割（モデル学習プロセスにおいて学習データに対しk分割交差検証が自動で実施される）



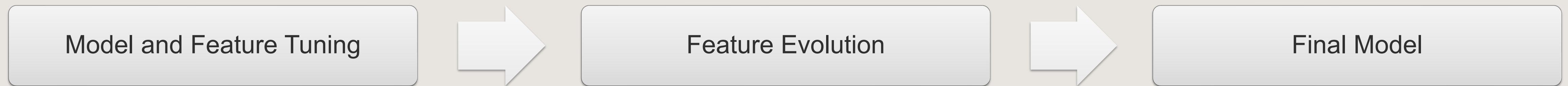
- モデルの設定（精度や学習時間を考慮した設定を自動で提案。マニュアルで詳細設定（利用するモデルの指定）することも可能）
- モデル学習（設定やデータサイズに応じて学習時間が変化）

Driverless AIのモデル作成のプロセス

H2O.ai



Driverless AIでは、大きく分けて以下の3ステップでモデル作成が実行される



データ、モデル、ハイパーパラメータの事前評価

- データのシフト検知、ターゲット変数の変換
- 異なるアルゴリズム、そのハイパーパラメータ探索（ランダムサーチ）、シンプルな特徴量エンジニアリングの試行
- 特徴量事前選択の実施
- 精度と計算効率の高い複数のモデルをFeature Evolutionステージに渡す

特徴量エンジニアリング

- Model and Feature Tuningステージから渡されたモデルやハイパーパラメータをもとに、さらに複雑な特徴量エンジニアリングを試す
- 遺伝的アルゴリズムによって、膨大な特徴量空間を繰り返し探索する
- Final Modelステージへ渡すモデルの決定

アンサンブル等を実施し、最終モデルを作る

- Feature Evolutionステージから渡されたモデルを用いた、全量データによるパラメータの推定、アンサンブル（スタッキング）の実施

AutoML (Driverless AI) 利用のコツ

H2O.ai

- ◆ データさえ準備できれば、いい感じで精度の良いモデルを自動で作成してくれる便利なツール、と考えておく
- ◆ どのようなデータが扱え、どんな予測ができる（対応する問題）かをまず理解しておく
- ◆ Driverless AIの細かい仕様や設定に最初はこだわらない（デフォルト設定である程度うまくいくように設計されている）。まずは予測を実施してみて、疑問に対して調査（モデル評価、解釈、予測ロジック etc.）を進めていくのがお勧め
- ◆ どのようなデータを準備・作成すればどのような課題が解決できるか、予測結果をどう実務に適用していくかにこだわる

とりあえず知っておこう – Keywords

アルゴリズム

- **GLM/Generalized Lenoir Model** – 重回帰モデル
- **決定木モデル**
- **勾配ブースティング** – 沢山の決定木モデルを組み合わせることにより予測精度を向上。テーブル型のデータに対してはデファクトスタンダードで用いられる。**XGBoost**や**LightGBM**といった勾配ブースティングを実施できるライブラリが有名
- **Neural Network** – Neural Networkの一種としてDeep Learningがある。数値やカテゴリデータに対してNeural Networkを適用することも可能

モデルのアンサンブル

- 予測精度の向上を狙ってモデルを組み合わせて予測すること（多数決を実施し間違った判断を減らすといったイメージ）

特徴量エンジニアリング

- 元データから予測に役立つと考えられる説明変数/特徴量を作成すること

Deep Learning

- 画像やテキストデータを扱うのに長けているNeural Network。一般的に大きなNeural Networkを指す
- **Tesorflow**や**PyTorch**といったライブラリが有名

学習、検証、テストデータセット

- データを分割し予測モデルの作成と精度確認を実施する。学習データでモデルを学習。検証データはその学習が上手く行くようにモデル学習をチューニング。テストデータで学習を終えたモデルの精度の最終確認を実施

モデル評価指標/Scorer

- モデルの予測精度の良さを測る指標。モデル作成前に指標を指定し、その指標が最大化/最小化するようにモデルを調整する

機械学習の解釈可能性/Machine Learning Interpretability

- モデルの解釈技術。どのような特徴量が効いているか、どのように効いているか、などを分析

学習（Training）と推論（Scoring）

- “学習（Training）”は、モデルを作成する工程。“推論（Scoring）”は、作成されたモデルを用いて予測を実施する工程

参考



特徴量エンジニアリングとは

学習データの特徴量から、予測に有用な新しい特徴量を作成する

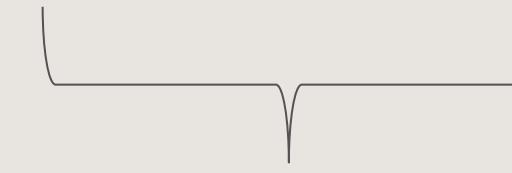
元データ

	A	B	E
1	X1	X2	Target
2	3.282	1.174	11.183
3	3.569	3.324	-44.186
4	4.748	4.810	-126.023
5	2.082	4.136	-20.757
6	2.274	0.967	13.986
7	4.345	4.438	-98.629
8	4.616	2.054	-21.466
9	4.194	0.926	18.597
10	1.229	3.428	5.336
11	2.894	4.084	-43.805

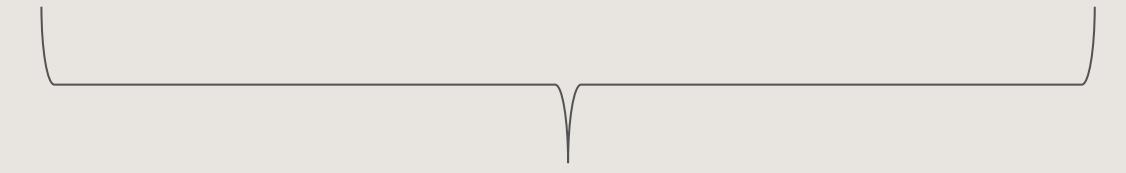
特徴量エンジニアリングを実施したデータ

	A	B	C	D	E
1	X1	X2	X1*X2	log(X1)	Target
2	3.282	1.174	3.854	0.516	11.183
3	3.569	3.324	11.865	0.553	-44.186
4	4.748	4.810	22.837	0.676	-126.023
5	2.082	4.136	8.613	0.319	-20.757
6	2.274	0.967	2.199	0.357	13.986
7	4.345	4.438	19.284	0.638	-98.629
8	4.616	2.054	9.480	0.664	-21.466
9	4.194	0.926	3.882	0.623	18.597
10	1.229	3.428	4.213	0.090	5.336
11	2.894	4.084	11.820	0.462	-43.805

元の特徴量



元の特徴量と、特徴量エンジニアリングで生成した特徴量



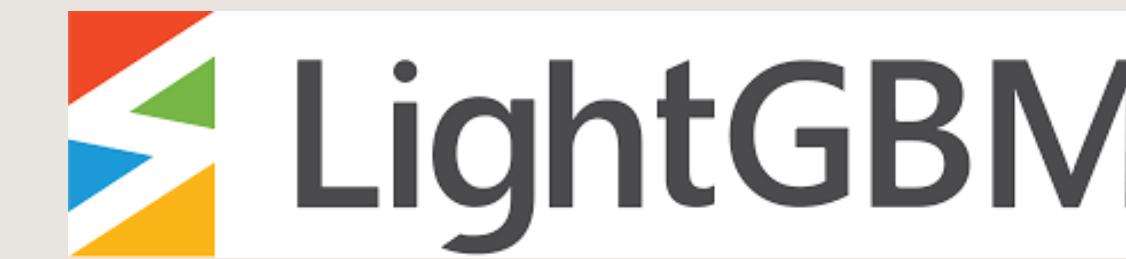
特徴量とターゲット変数間の複雑な関係をより反映し、予測精度の向上が期待できる

Driverless AIには40種類以上の特徴量エンジニアリングが実装されており、データ型に合わせて自動で実施される

Driverless AIがサポートするアルゴリズム

H2O.ai

- XGBoost
- Light GBM
- Multi Layer Perceptron
- Decision Tree
- Random Forrest
- Generalized Lenoir Model
- その他Deep Learning
など



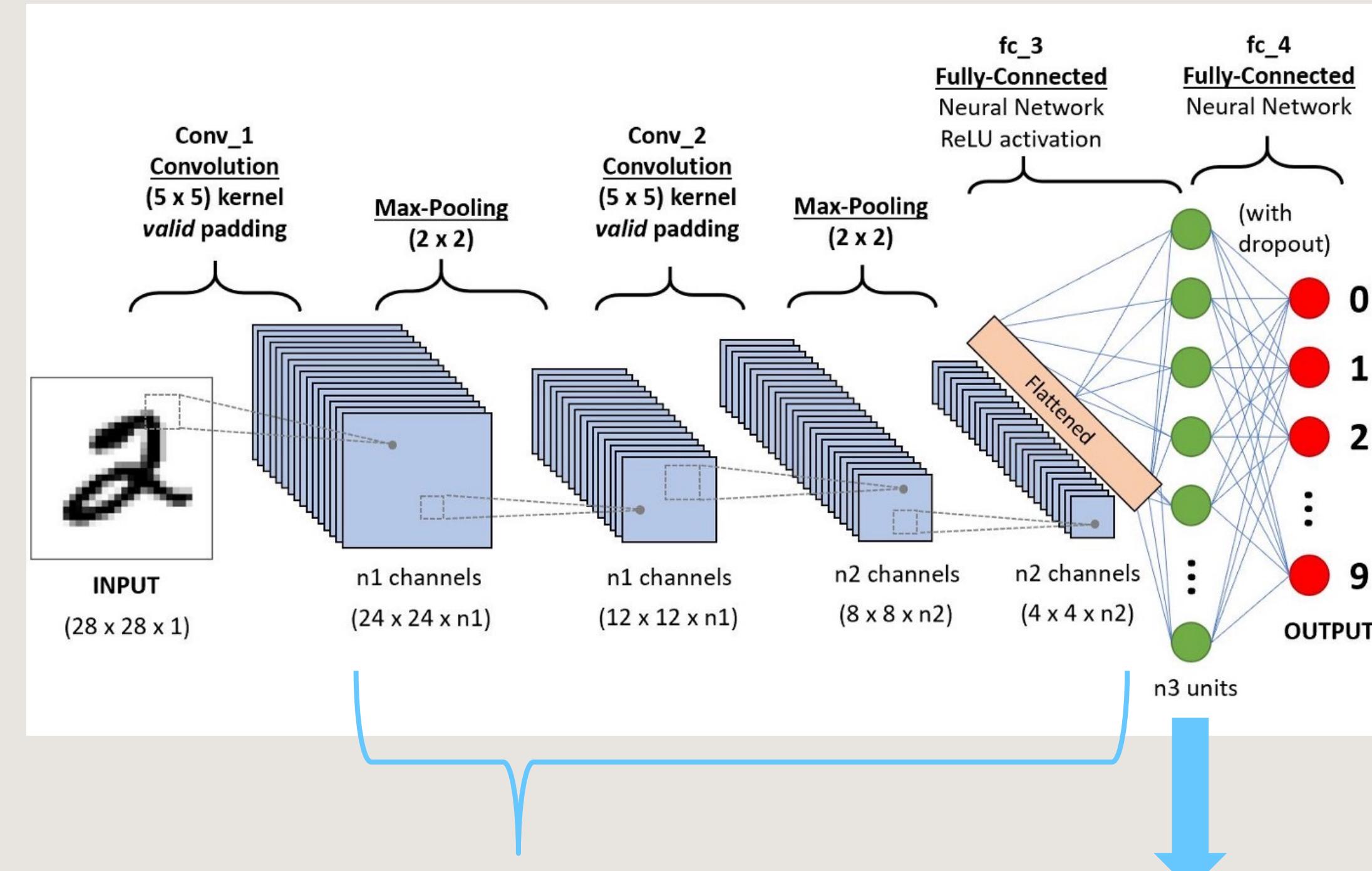
Driverless AIで利用できるアルゴリズム : <https://docs.h2o.ai/driverless-ai/latest-stable/docs/userguide/supported-algorithms.html>

H2O.ai Confidential

画像データを利用したモデル作成

畳み込みニューラルネットワーク (CNN) を用い、画像から特徴量を抽出する

さまざまなデータ拡張の
手法の利用が可能



転移学習、もしくはFine Tuningの実施

- 転移学習：学習済みモデルの重みを初期値として学習を実施
- Fine Tuning：学習済みモデルを特徴量抽出器として利用

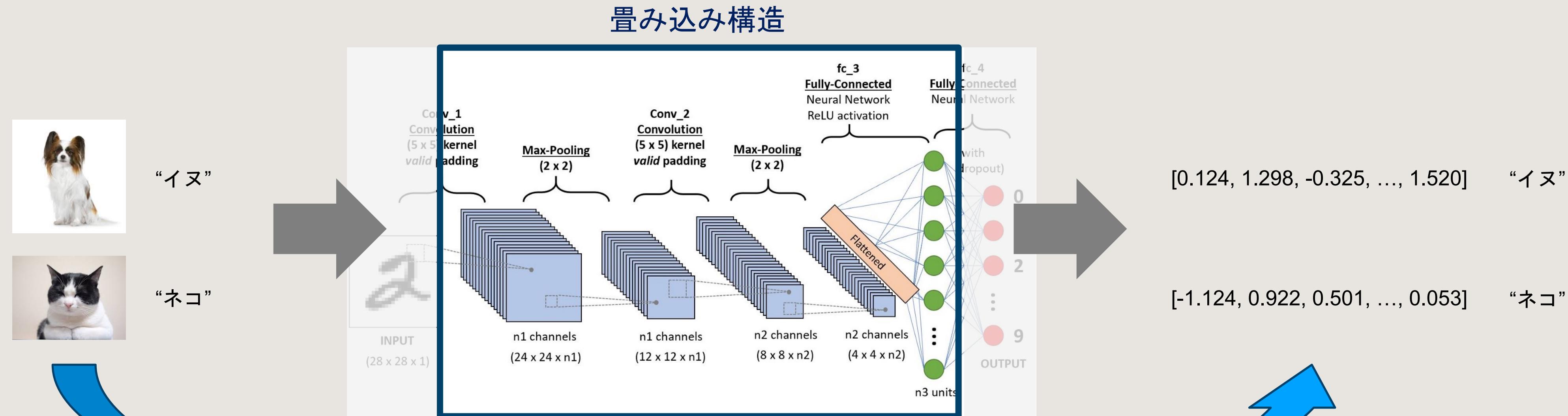
Global Average Pooling層からの出力を、Driverless AI
のモデル学習プロセスの入力とする

利用できる事前学習CNNモデル

- densenet121
- efficientnetb0
- efficientnetb2
- inception_v3
- mobilenetv2
- resnet34
- resnet50
- seresnet50
- seresnext50
- xception

画像データを利用したモデル作成

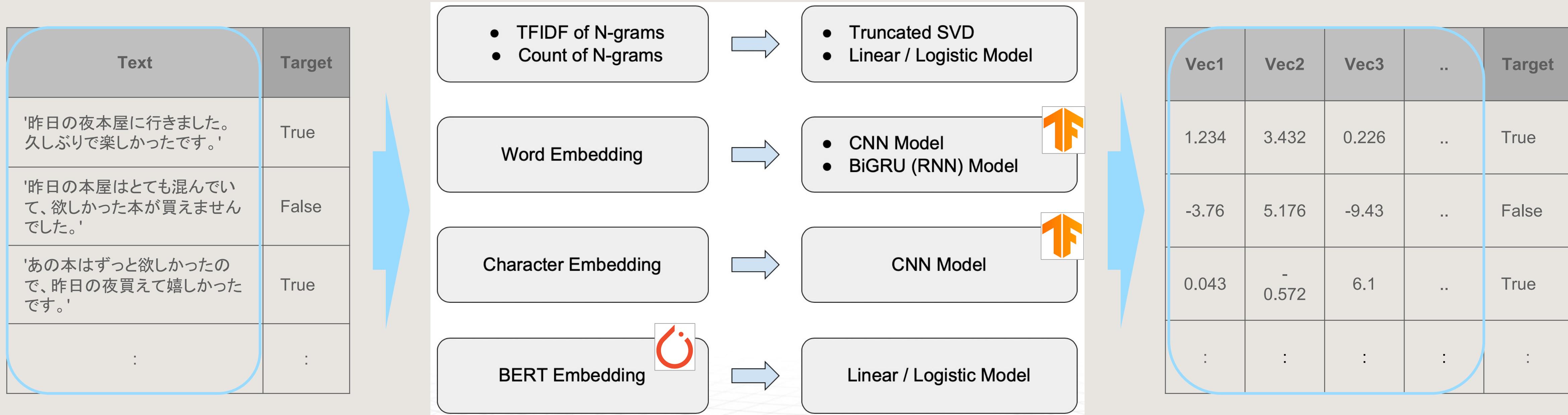
畳み込みニューラルネットワーク (CNN) を用い、画像から特徴量を抽出する



畳み込み構造により、画像がその特徴量を表すベクトルに変換される

テキストデータを利用したモデル作成

テキストデータを、Deep Learningなどさまざまな方法で、ベクトルへ変換する



- Bag of Words, TFIDF + SVD, GLM
- Word/Character Embedding + CNN, RNN
- BERT

時系列データに対するアプローチ

Driverless AIでの時系列予測は、古典的時系列モデル（指数平滑化、ARIMA、VAR、など）を用いたアプローチではなく、特徴量エンジニアリングを工夫し（Lag特徴量など）、時系列データをテーブル型データへ変換し、機械学習モデルをあてはめるアプローチで実施

例)

- ターゲット変数からLagを取り、特徴量として利用。また、Lag特徴量から平均を取ったりとさらに複雑な特徴量エンジニアリングを実施
- 日付変数からも特徴量エンジニアリング（月や曜日の抽出）を実施する
- その他の説明変数を特徴量として利用することも可能



Case Study

Case Study – 予知保全/Predictive Maintenance

H2O.ai

エンジンの交換判断にはベテラン整備士の知見を必要としている。毎回の確認にそれら人手をかけて実施するのには大きなコストがかかる。

エンジンの交換時期を予測することにより、交換の確認作業の省力化を実施（明らかに大丈夫な場合は確認作業を簡略化する）したい。

また、そのような予測が実施できることは、安全管理（どれくらいで交換が必要となるか作業員に注意喚起を実施）にも非常に有用。

エンジン各所に取り付けられたセンサーデータを、交換時期まで取得し続けたデータがあり、それらを利用して交換必要時期を予測するモデルを開発したい。

また、本件は安全管理に関わることから、確認作業の省力化によるメリットとリスクを比較して分析する必要がある。



モデル作成のステップ

H2O.ai

1. ターゲット変数の分析
2. 予測要件の定義
3. 説明変数を考える
4. モデリング用データの作成
5. 予測モデルの作成
6. 精度確認やモデルの理解
7. 予測結果のビジネス的解釈

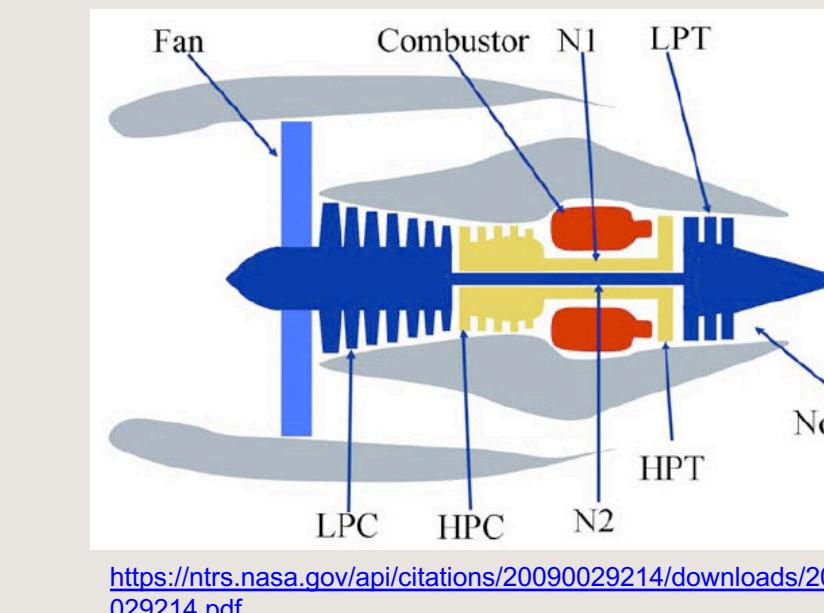
データについて

260のエンジン (Unit) の各種センサー値を運用開始から交換時期まで時系列で記録したデータ

	unit_ID	cycles	time_to_deterioration	setting_1	setting_2	setting_3	T2	T24	T30	T50	...	phi	NRf	NRc	BPR	farB	htB
0	1	1	148	34.9983	0.8400	100.0	449.44	555.32	1358.61	1137.23	...	183.06	2387.72	8048.56	9.3461	0.02	
1	1	2	147	41.9982	0.8408	100.0	445.00	549.90	1353.22	1125.78	...	130.42	2387.66	8072.30	9.3774	0.02	
2	1	3	146	24.9988	0.6218	60.0	462.54	537.31	1256.76	1047.45	...	164.22	2028.03	7864.87	10.8941	0.02	
3	1	4	145	42.0077	0.8416	100.0	445.00	549.51	1354.03	1126.38	...	130.72	2387.61	8068.66	9.3528	0.02	
4	1	5	144	25.0005	0.6203	60.0	462.54	537.07	1257.71	1047.93	...	164.31	2028.00	7861.23	10.8963	0.02	
...	
53754	260	312	4	20.0037	0.7000	100.0	491.19	608.79	1495.60	1269.51	...	314.05	2389.02	8169.64	9.3035	0.03	
53755	260	313	3	10.0022	0.2510	100.0	489.05	605.81	1514.32	1324.12	...	371.22	2388.42	8245.36	8.7586	0.03	
53756	260	314	2	25.0041	0.6200	60.0	462.54	537.48	1276.24	1057.92	...	163.74	2030.33	7971.25	11.0657	0.02	
53757	260	315	1	25.0033	0.6220	60.0	462.54	537.84	1272.95	1066.30	...	164.37	2030.35	7972.47	11.0537	0.02	
53758	260	316	0	35.0036	0.8400	100.0	449.44	556.64	1374.61	1145.52	...	183.09	2390.38	8185.35	9.3998	0.02	

↓
Unit=1のみ抽出

	unit_ID	cycles	time_to_deterioration
0	1	1	148
1	1	2	147
2	1	3	146
3	1	4	145
4	1	5	144
...
144	1	145	4
145	1	146	3
146	1	147	2
147	1	148	1
148	1	149	0



<https://ntrs.nasa.gov/api/citations/20090029214/downloads/2009029214.pdf>

データ上149 Cycle存在する
= 149 Cycleで交換が発生した

unit_ID :

- エンジン (Unit) ID

cycle :

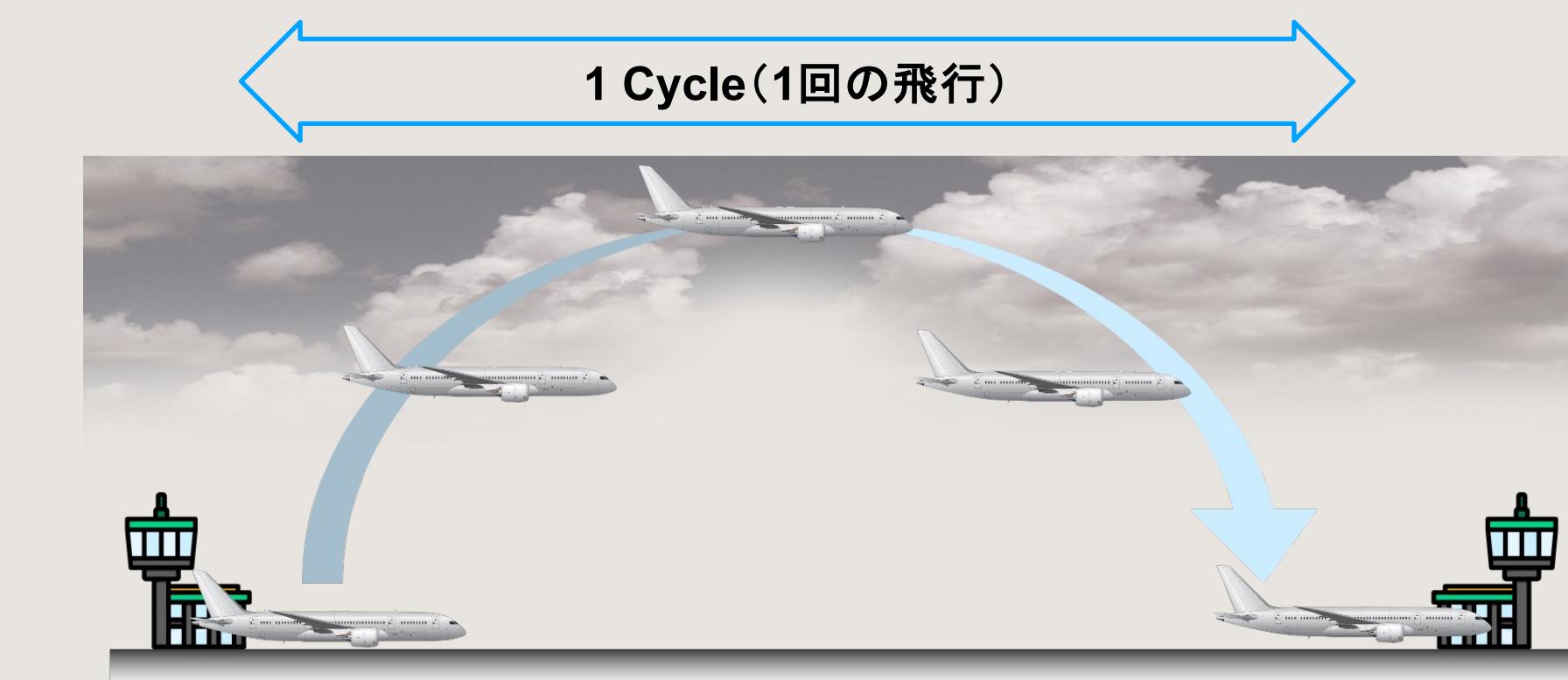
- (現時点での) 飛行回数

time_to_deterioration :

- 交換までの残り飛行回数 (cycleの逆)

Setting_1以降 :

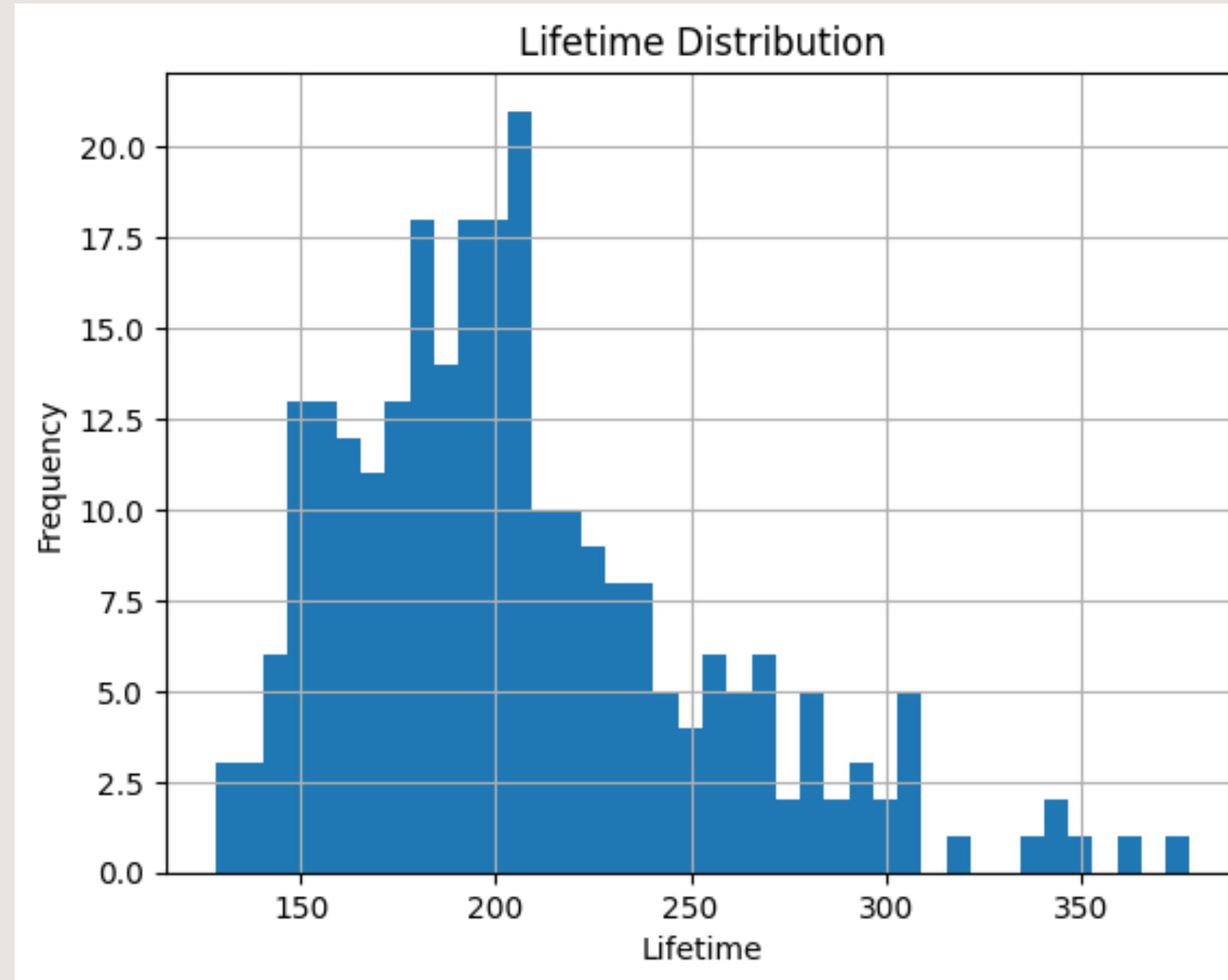
- センサーデータ。各Cycleにおける代表値。24個



1. ターゲット変数の分析

どれくらいのCycleで交換が発生しているかを確認

260Unitの交換時期

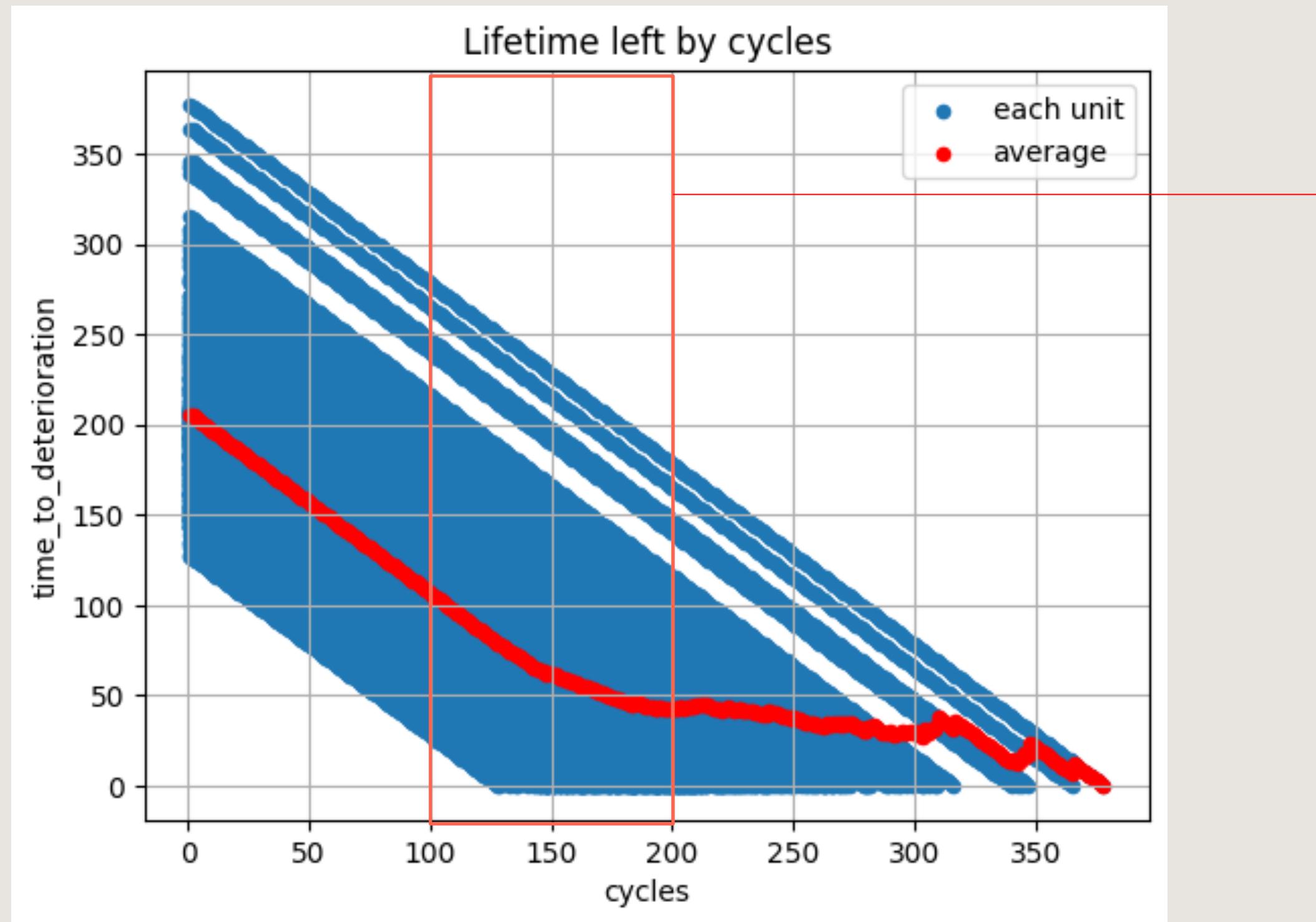


mean	206.765385
std	46.782198
min	128
25%	174
50%	199
75%	230.25
max	378

- ✓ 260 Unitの内、最も交換時期が短かったUnitはCycleが128
- ✓ Cycleが200前後で交換を迎えるUnitが多い

2. 予測要件の定義

あるCycle時点における、残存交換時間を予測したい



260 Unitの各Cycleにおける残存交換時間 (time_to_deterioration)

100 Cycle以前に交換を迎えたUnitは無い、200 Cycle前後で交換を迎えるUnitが多い

- Cycle<100 : 検査の実施無し
- 100<=Cycle<200 : 予測を実施し検査の実施有無を判断
- 200<=Cycle : 全て検査を実施

100<=Cycle<200において、

- 予測により残存交換時間が短いUnitは検査を実施、それ以外は実施無しとする
- 予測残存交換時間を閾値とし、短く設定した場合、長く設定した場合を分析

短く設定すると、検査を実施しなくて良いUnitが増える（すぐ交換が必要そうなものののみ検査）ので大きくコスト削減できる。検査を実施しないUnitが多いのでリスクは高くなる？

長く設定すると、検査実施数は多く（しばらく交換が必要なさそうなものまで検査）なる。多くを検査することになるので、リスクは低くなる？

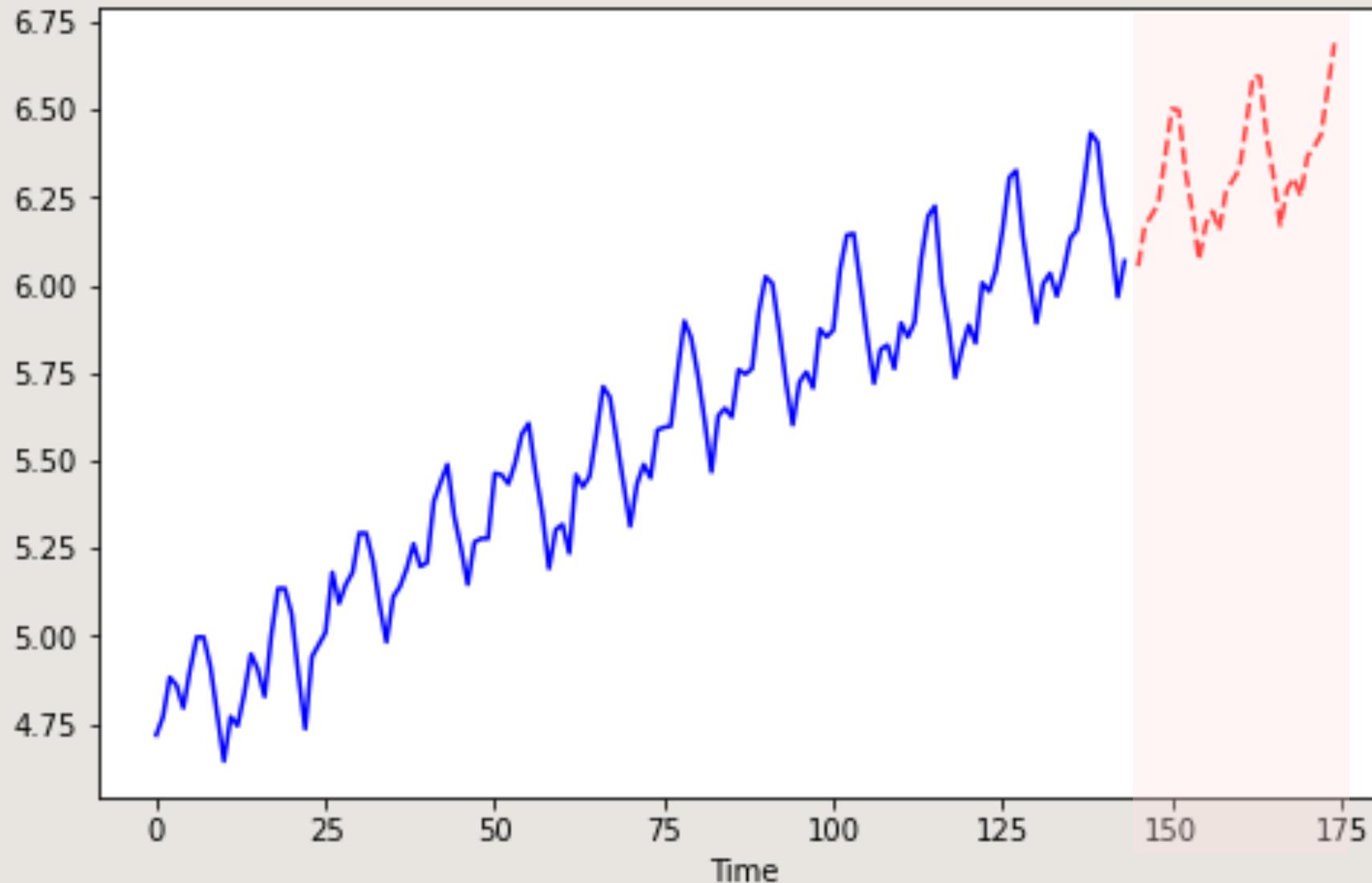
センサーデータに対するアプローチ：予測方法

H2O.ai

センサーデータ（時系列データ）をどのように扱うか？モデリング手法は？

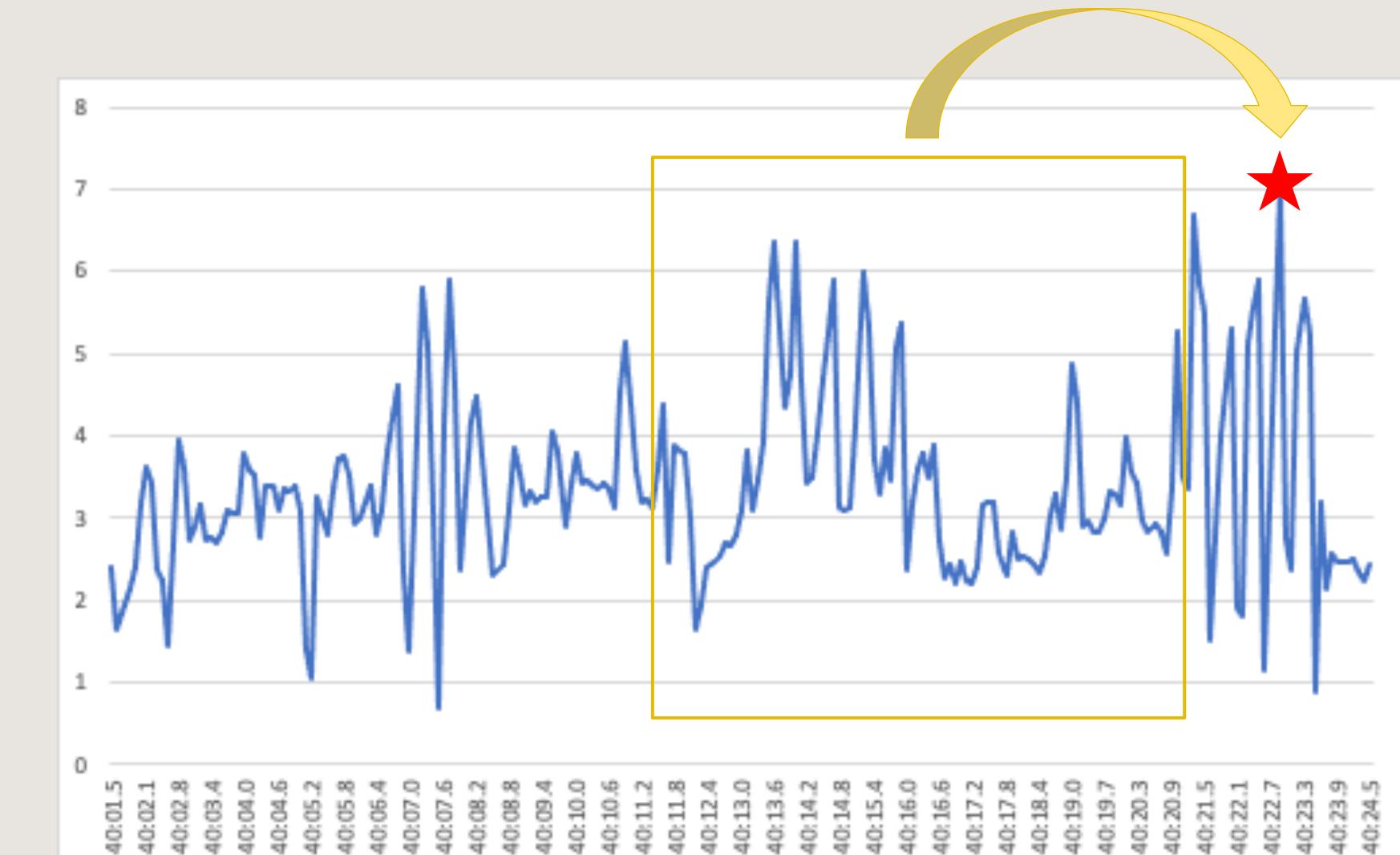
時系列予測による方法

- ・ 時系列の将来の値そのものを予測



非時系列に持ち込む方法

- ・ 将来発生する何らかの事象の発生を、事前の情報から予測



データ準備：

- ・ 時間を示す変数と、予測対象の変数を準備（時間等間隔で観測されているデータが前提）

モデリング手法：

- ・ 時系列予測

データ準備：

- ・ 予測を実施したい事象のラベル付け（教師あり学習の場合）
- ・ 事象発生前の情報から、特徴量を作成

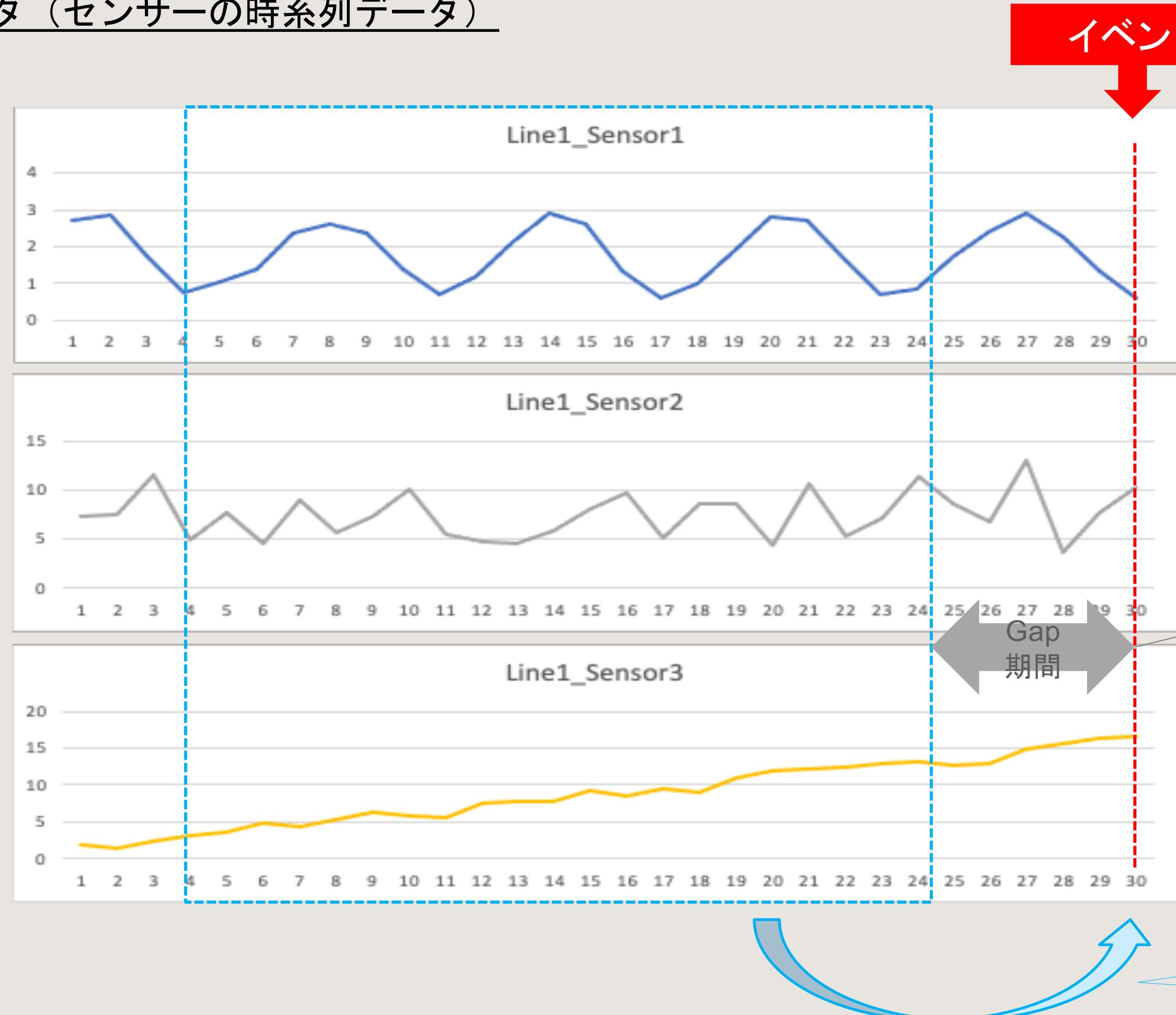
モデリング手法：

- ・ 教師なし学習
- ・ 教師あり学習（非時系列）

センサーデータに対するアプローチ： 非時系列に持ち込む方法（1）

教師あり学習（分類もしくは回帰問題）として実施する異常検知

データ（センサーの時系列データ）



～実施条件(データ面の整備)～

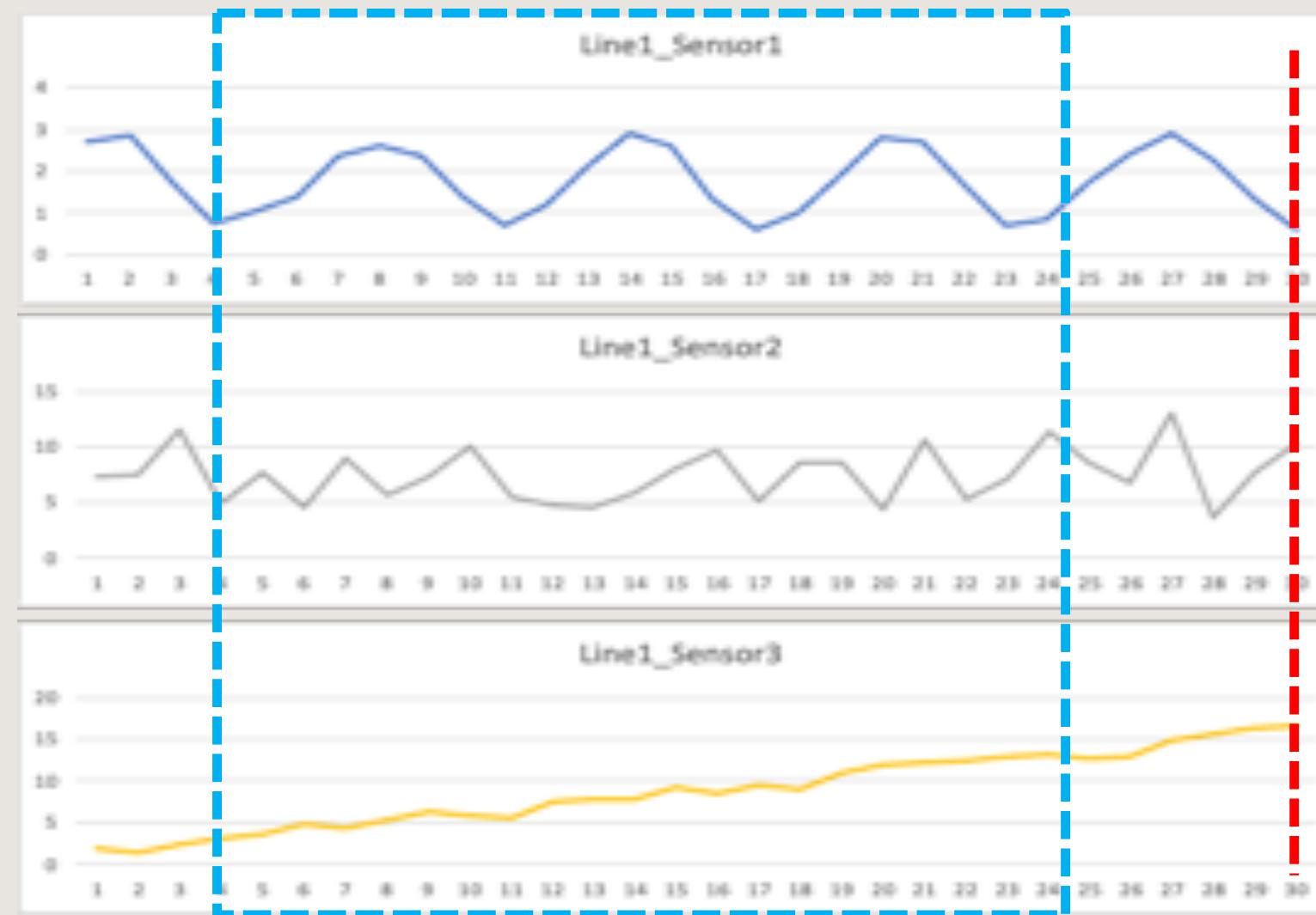
- ✓ イベント発生データ(ラベル)がある程度取得できている
- ✓ その異常に影響がある(相関)と考えられる設備のセンサーデータが取得できている
- ✓ 異常発生前に何らかの兆候が現れている（兆候なしの突発異常でない）

Gap期間：
データを取得し、予測し、アクションを実施できる期間を想定

イベント発生前の各センサーデータをインプットとして、イベント発生を予測

センサーデータに対するアプローチ： 非時系列に持ち込む方法（2）

モデリング用データの作成



インプットとして用いる期間の
 • 平均値
 • ばらつき(標準偏差)
 • 範囲(Max-Min)
 • 平均変化率 etc.
 など各統計量を特徴量とする

期間自体複数作成(例:全体10分、直近5分)

モデリング用データ

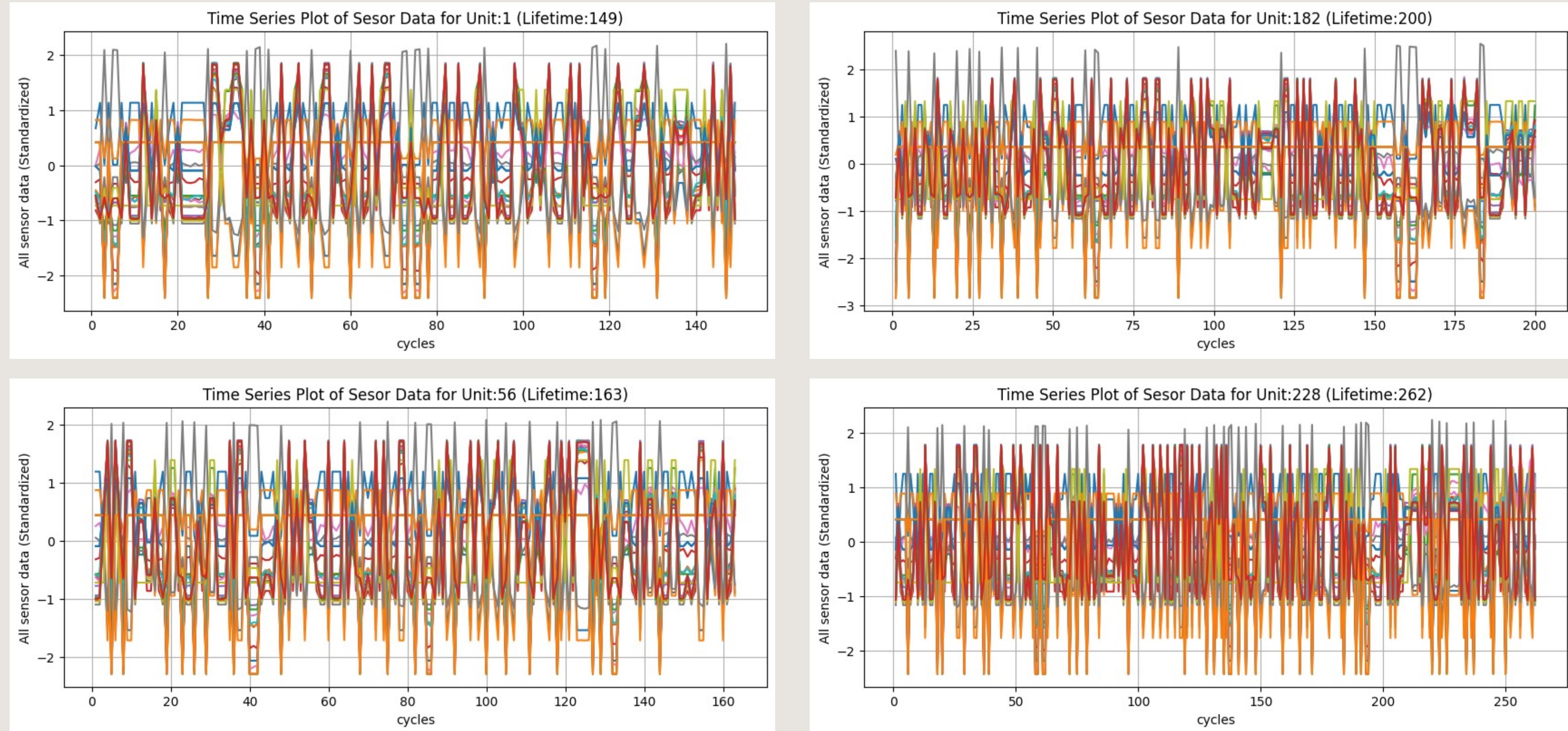
Sensor 1 平均	Sensor 1 標準偏差	Sensor 2 平均	Sensor 2 標準偏差	..	イベント 情報
10.3	2.9	5.3	7.3	..	X
11.2	1.4	10.2	2.4	..	Y
13.4	1.5	9.3	4.4	..	Z
:	:	:	:		:

- 1つのイベント発生とインプット期間から作成した特徴量を1レコードに集約する

3. 説明変数を考える

ポイント：時系列データは、まずはプロットを作成してみる

Unit_IDが1, 56, 182, 228のエンジンの各センサー値を、運用開始から交換まで時系列でプロット
(値の単位が異なるため各センサー値に標準化を実施してからプロット)



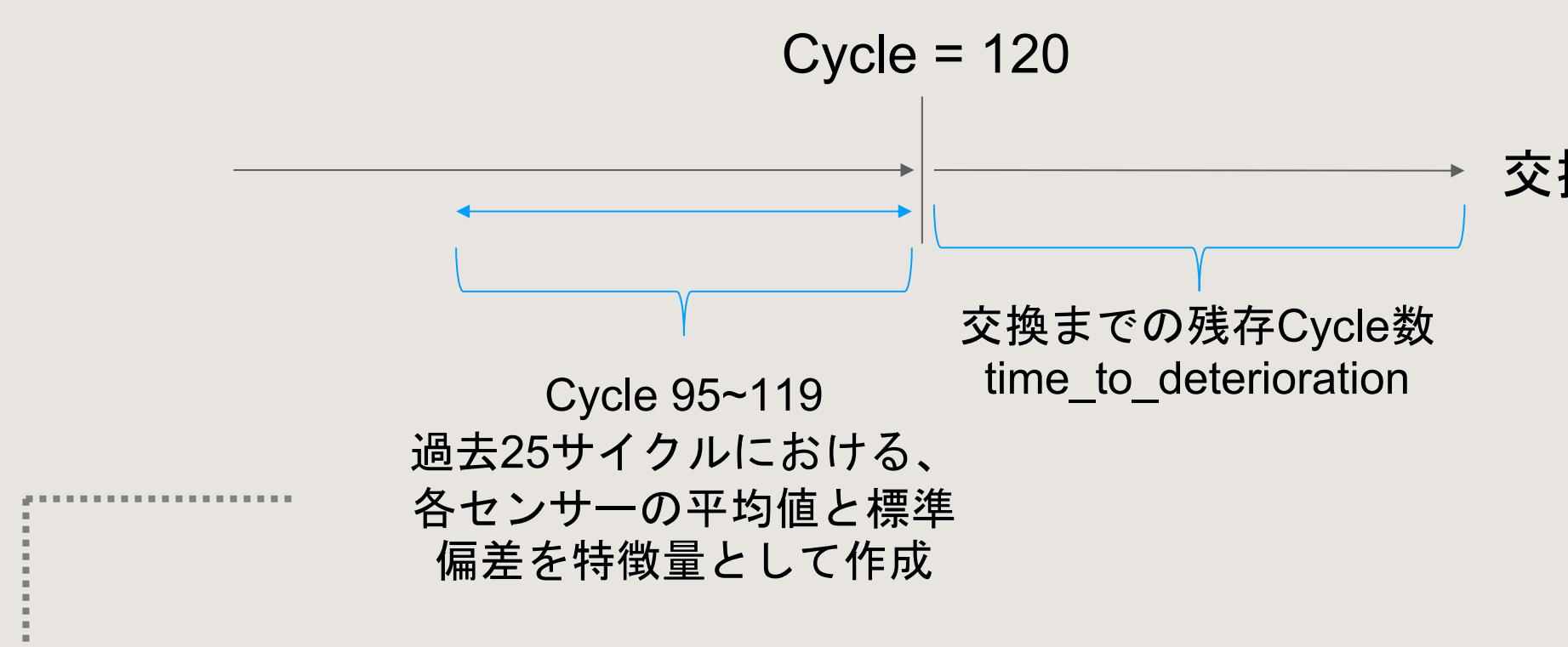
ここでは一旦、（ドメイン知識なども元にし）交換時期の前25 Cycleに兆候が発生すると仮定

4. モデリング用データの作成

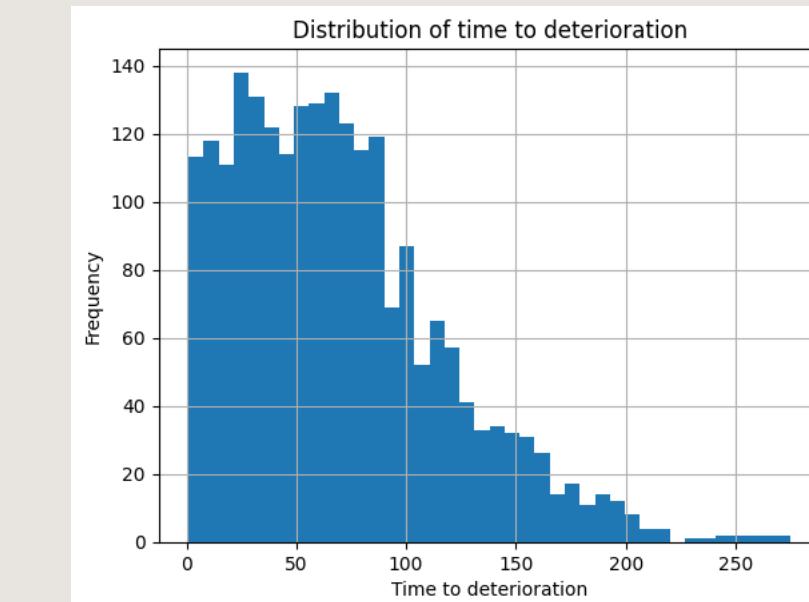
各Unitにおいて、あるCycle時点における、交換までの残りの残存Cycle数がターゲット変数 (time_to_deterioration) となり、その時点の前25 Cycleを各センサーから特徴量を作成する期間とする

各Unitにおける全てのCycle時点の情報を利用すると、Unitへのオーバーフィットが考えられるため、サンプリングを実施
作成したモデリング用データは学習とテストデータに分割

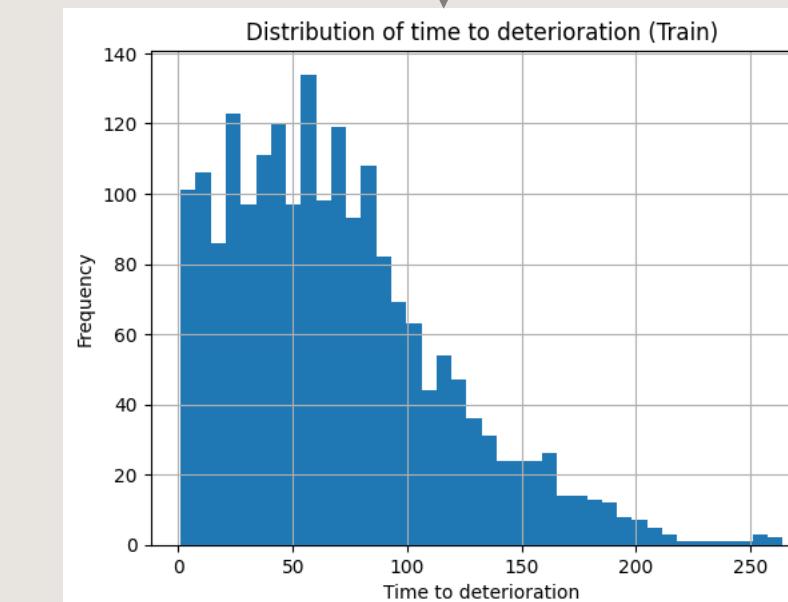
例) unit_ID=1 at cycles=120



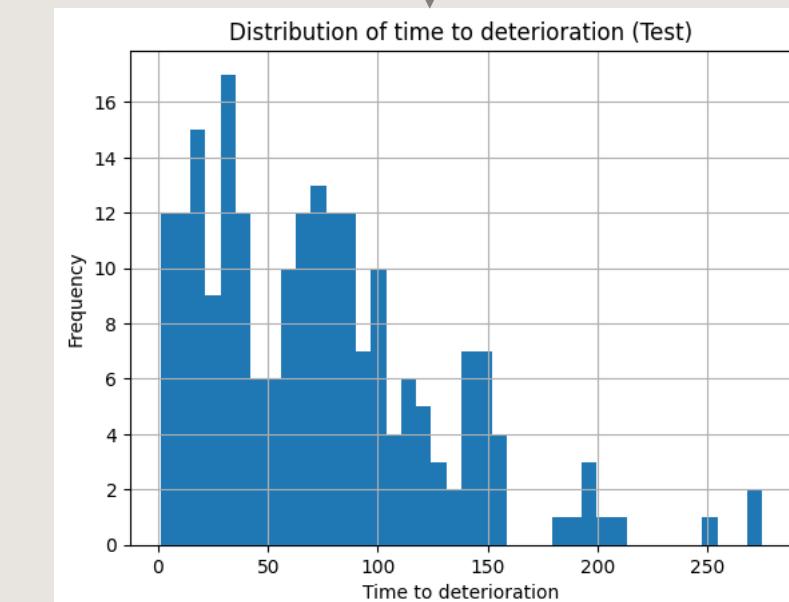
モデリング用データ
N=2,216



学習データ
N=2,003



テストデータ
N=213



unit_ID	cycles	time_to_deterioration	setting_1-mean	setting_2-mean	setting_3-mean	T2-mean	T24-mean	T30-mean	T50-mean	phi-sd	NRf-sd	NRc-	
1	120	29	25.883372	0.628304	95.2	470.9576	578.3988	1418.8880	1202.9252	...	138.767936	119.374676	73.8056
1	124	25	23.723044	0.599132	95.2	474.5672	582.9664	1430.5744	1216.2684	...	135.328416	119.396075	74.3692
1	125	24	25.003032	0.622780	95.2	472.8052	580.7724	1424.8056	1209.4468	...	136.629380	119.386880	73.0643
2	153	116	28.482604	0.707784	95.2	467.1748	573.5012	1403.4784	1177.5300	...	113.389744	119.327653	70.9373
2	173	96	24.882896	0.615056	92.0	472.4908	577.3712	1409.7260	1193.0236	...	130.746788	146.879170	91.6240
...	
260	130	186	23.242544	0.537328	92.0	473.4632	578.0828	1411.8896	1196.9112	...	139.838800	146.897945	95.6909
260	156	160	28.923252	0.657436	92.0	462.9084	565.0076	1376.7656	1157.8228	...	110.566501	146.876920	91.7783
260	169	147	21.882824	0.498996	95.2	474.9688	583.0120	1428.7052	1218.3416	...	152.139911	119.310863	80.8120
260	172	144	22.882808	0.531812	93.6	473.9940	580.4292	1418.0300	1205.7976	...	151.877535	134.599987	88.2781
260	186	130	21.043268	0.518572	90.4	476.0600	580.2484	1414.9804	1201.4368	...	139.090953	156.813966	102.6389

モデリング用データ

学習データ : https://github.com/yukismd/SensorData_PdM/blob/main/data_modified/modeling_engine_partial_TRAIN.csv

テストデータ : https://github.com/yukismd/SensorData_PdM/blob/main/data_modified/modeling_engine_partial_TEST.csv

4. モデリング用データの作成

前ページ左のデータ変換詳細

H2O.ai

例) unit_ID=1 at cycles=120

120 Cycleの
直前25 Cycle

unit_ID	cycles	time_to_deterioration	setting_1	setting_2	setting_3	T2	T24	T30	T50	...	phi	NRf	NR
1	95	54	35.0058	0.8403	100.0	449.44	555.64	1369.57	1136.49	...	182.37	2387.76	8058.1
1	96	53	42.0062	0.8402	100.0	445.00	549.65	1358.37	1125.38	...	130.49	2387.55	8067.3
1	97	52	42.0015	0.8413	100.0	445.00	549.43	1354.86	1127.44	...	130.44	2387.63	8071.4
1	98	51	0.0025	0.0015	100.0	518.67	643.26	1593.43	1409.77	...	521.22	2388.17	8124.8
1	99	50	10.0051	0.2502	100.0	489.05	604.97	1501.19	1309.55	...	371.05	2388.25	8119.0
1	100	49	20.0027	0.7000	100.0	491.19	607.60	1488.43	1259.11	...	314.17	2388.06	8048.3
1	101	48	42.0021	0.8400	100.0	445.00	549.99	1349.48	1135.38	...	130.44	2387.62	8066.8
1	102	47	35.0063	0.8400	100.0	449.44	555.90	1376.66	1139.31	...	182.22	2387.77	8059.0
1	103	46	20.0057	0.7010	100.0	491.19	607.32	1495.89	1258.36	...	314.23	2388.13	8045.3
1	104	45	35.0078	0.8400	100.0	449.44	556.29	1362.75	1129.45	...	182.92	2387.77	8056.8
1	105	44	20.0052	0.7000	100.0	491.19	607.04	1487.94	1261.08	...	314.16	2388.10	8045.3
1	106	43	0.0008	0.0019	100.0	518.67	643.12	1590.38	1414.45	...	520.52	2388.12	8122.6
1	107	42	35.0037	0.8411	100.0	449.44	555.85	1365.70	1139.21	...	183.02	2387.80	8064.9
1	108	41	41.9998	0.8414	100.0	445.00	550.09	1362.34	1133.92	...	130.73	2387.70	8066.3
1	109	40	41.9991	0.8419	100.0	445.00	549.88	1360.26	1137.95	...	130.97	2387.66	8063.4
1	110	39	20.0074	0.7019	100.0	491.19	607.79	1489.60	1267.13	...	314.53	2388.03	8049.0
1	111	38	0.0027	0.0017	100.0	518.67	643.25	1595.96	1419.21	...	520.65	2388.18	8117.3
1	112	37	42.0049	0.8400	100.0	445.00	549.91	1360.02	1137.89	...	130.77	2387.58	8069.3
1	113	36	0.0013	0.0000	100.0	518.67	642.60	1592.73	1418.58	...	520.45	2388.21	8119.3
1	114	35	35.0026	0.8400	100.0	449.44	556.07	1364.28	1136.49	...	182.57	2387.83	8057.3
1	115	34	20.0045	0.7013	100.0	491.19	608.28	1481.57	1267.61	...	314.63	2388.13	8047.3
1	116	33	24.9980	0.6200	60.0	462.54	536.47	1267.58	1058.31	...	164.87	2027.97	7864.1
1	117	32	25.0000	0.6219	60.0	462.54	536.52	1263.31	1048.66	...	164.11	2028.00	7866.1
1	118	31	35.0030	0.8400	100.0	449.44	555.76	1373.37	1140.43	...	182.88	2387.77	8050.3
1	119	30	25.0056	0.6200	60.0	462.54	537.29	1266.53	1061.97	...	164.87	2027.95	7861.1
1	120	29	34.9991	0.8409	100.0	449.44	555.99	1369.65	1144.41	...	182.38	2387.85	8058.7

元センサーデータ

直前25 Cycleにおけるセンサー値の平均値と標準偏差を算出

unit_ID	cycles	time_to_deterioration	setting_1-mean	setting_2-mean	setting_3-mean	T2-mean	T24-mean	T30-mean	T50-mean	...	phi-sd	NRf-sd	NRc-
1	120	29	25.883372	0.628304	95.2	470.9576	578.3988	1418.8880	1202.9252	...	138.767936	119.374676	73.8056
1	124	25	23.723044	0.599132	95.2	474.5672	582.9664	1430.5744	1216.2684	...	135.328416	119.396075	74.3692
1	125	24	25.003032	0.622780	95.2	472.8052	580.7724	1424.8056	1209.4468	...	136.629380	119.386880	73.0643
2	153	116	28.482604	0.707784	95.2	467.1748	573.5012	1403.4784	1177.5300	...	113.389744	119.327653	70.9373
2	173	96	24.882896	0.615056	92.0	472.4908	577.3712	1409.7260	1193.0236	...	130.746788	146.879170	91.6240
...
260	130	186	23.242544	0.537328	92.0	473.4632	578.0828	1411.8896	1196.9112	...	139.838800	146.897945	95.6909
260	156	160	28.923252	0.657436	92.0	462.9084	565.0076	1376.7656	1157.8228	...	110.566501	146.876920	91.7783
260	169	147	21.882824	0.498996	95.2	474.9688	583.0120	1428.7052	1218.3416	...	152.139911	119.310863	80.8120
260	172	144	22.882808	0.531812	93.6	473.9940	580.4292	1418.0300	1205.7976	...	151.877535	134.599987	88.2781
260	186	130	21.043268	0.518572	90.4	476.0600	580.2484	1414.9804	1201.4368	...	139.090953	156.813966	102.6389

モダリング用データ

Unit 1の120 Cycleにおける情報

5. 予測モデルの作成

Driverless AIのExperiment設定

The screenshot shows the H2O.ai Experiment setup interface. Key configuration details are highlighted:

- EXPERIMENT TYPE:** SUPERVISED
- DISPLAY NAME:** engine_p_default
- TRAINING DATASET:** modeling_engine_partial
- TEST DATASET:** modeling...
- FOLD COLUMN:** unit_ID
- TARGET COLUMN:** time_to_deterioration
- TYPE:** int
- COUNT:** 2003
- MEAN:** 69.542
- STDEV:** 47.334
- DROPPED COLUMNS:** --
- TIME COLUMN:** Disabled
- TRAINING SETTINGS:**
 - ACCURACY: 6
 - TIME: 3
 - INTERPRETABILITY: 7
 - SCORER: RMSE
- EXPERT SETTINGS:**
 - MODEL TYPE: CLASSIFICATION
 - REPRODUCIBLE: ON
 - GPUS: ON
- PARENT EXPERIMENT:** ENGINE_P_DEFAULT
- Buttons:** LAUNCH EXPERIMENT, CREATE LEADERBOARD

テストデータの指定

ターゲット変数 (time_to_deterioration) の指定

Fold Column (unit_ID) の指定

- Fold Columnは、k分割検証法において、同じIDは同じグループに入るように指定。今回は同じUnit内で複数のオブザーベーションがあるので、リーク（スコアが不意に良く見えてしまいます）を防ぐために指定

6. 精度確認やモデルの解釈

Driverless AIのExperiment完了画面

H2O.ai Experiment engine_p_d...

DRIVERLESS AI 1.10.6.1 (LTS) – AI TO DO AI
Current User – yuki.shimada@h2o.ai

EXPERIMENT SETUP

EXPERIMENT TYPE: SUPERVISED
DISPLAY NAME: engine_p_default

TRAINING DATASET: modeling_engine_partial...
TEST DATASET: modeling_...

ROWS: 2K COLUMNS: 51 FOLD COLUMN: unit_ID WEIGHT COLUMN: --

TARGET COLUMN: time_to_deterioration DROPPED COLUMNS: 0 TIME COLUMN: [OFF]

TYPE: int COUNT: 2003 MEAN: 69.542 STDEV: 47.334

ASSISTANT

STATUS: COMPLETE

- INTERPRET THIS MODEL ▾
- DIAGNOSE MODEL ON NEW DATASET...
- MODEL ACTIONS ▾
- VISUALIZE SCORING PIPELINE
- DOWNLOAD SCORING PIPELINE ▾
- DEPLOY
- DOWNLOAD PREDICTIONS ▾
- DOWNLOAD SUMMARY & LOGS
- DOWNLOAD AUTODOC
- TUNE EXPERIMENT ▾

TRAINING SETTINGS

ACCURACY: 6 TIME: 3 INTERPRETABILITY: 7 SCORER: RMSE

EXPERT SETTINGS

MODEL TYPE: CLASSIFICATION REGRESSION
REPRODUCIBLE: ON OFF GPUS: ON OFF

CPU / MEMORY

Insights Scores Notifications Log Trace

CPU

MEM

VARIABLE IMPORTANCE

53_InteractionMul:BPR-mean:Ps30-mean	1.00
8_NRc-mean	0.16
26_T2-mean	0.09
38_cycles	0.07
125_NRf-mean	0.03
39_epr-mean	0.02
15_Nf-sd	0.02
25_Ps30-sd	0.02
29_T24-sd	0.02
31_T30-sd	0.02
52_CVCatNumEnc:PCNfR_dmd-mean:Ps30-mean.sd	0.02
40_epr-sd	0.02
19_Nc-sd	0.01
28_T24-mean	0.01

RESIDUALS

Experiment: engine_p_default (83b7cd5c-bcb1-11ee-a7bd-2e53304ac579)
Version: 1.10.6.1, 2024-01-27 01:21, GUI
Settings: 6/3/7, seed=248391984, GPUs disabled
Train data: modeling_engine_partial_TRAIN.csv (2003, 50)
Validation data: N/A
Test data: [Test] (213, 50)
Target column: time_to_deterioration (regression, standardize-transformed)
System specs: Docker/Linux, 24 GB, 32 CPU cores, 0/0 GPU
Max memory usage: 1.06 GB, 0 GB GPU, 0.0999 GB MOJO
Recipe: AutoDL (35 iterations, 4 individuals)
Validation scheme: random, 6 internal holdouts (3-fold CV)
Feature engineering: 307 features scored (77 selected)
Timing: MOJO latency 0.2791 millis (4.7MB), Python latency 75.5632 millis (69.3kB)
Data preparation: 6 seconds
Shift/Leakage detection: 1 second
Model and feature tuning: 1 minute 48 seconds (103 of 126 models trained)
Feature evolution: 2 minutes 21 seconds (228 of 384 models trained)
Final pipeline training: 43 seconds (24 models trained)
Python / MOJO scorer building: 32 seconds / 25 seconds
Validation score: RMSE = 47.32249 (constant preds of 69.54)
Validation score: RMSE = 36.17978 +/- 4.04202 (baseline)
Validation score: RMSE = 31.00805 +/- 3.841072 (final pipeline)
Test score: RMSE = 37.88478 +/- 3.841072 (final pipeline)

SUMMARY

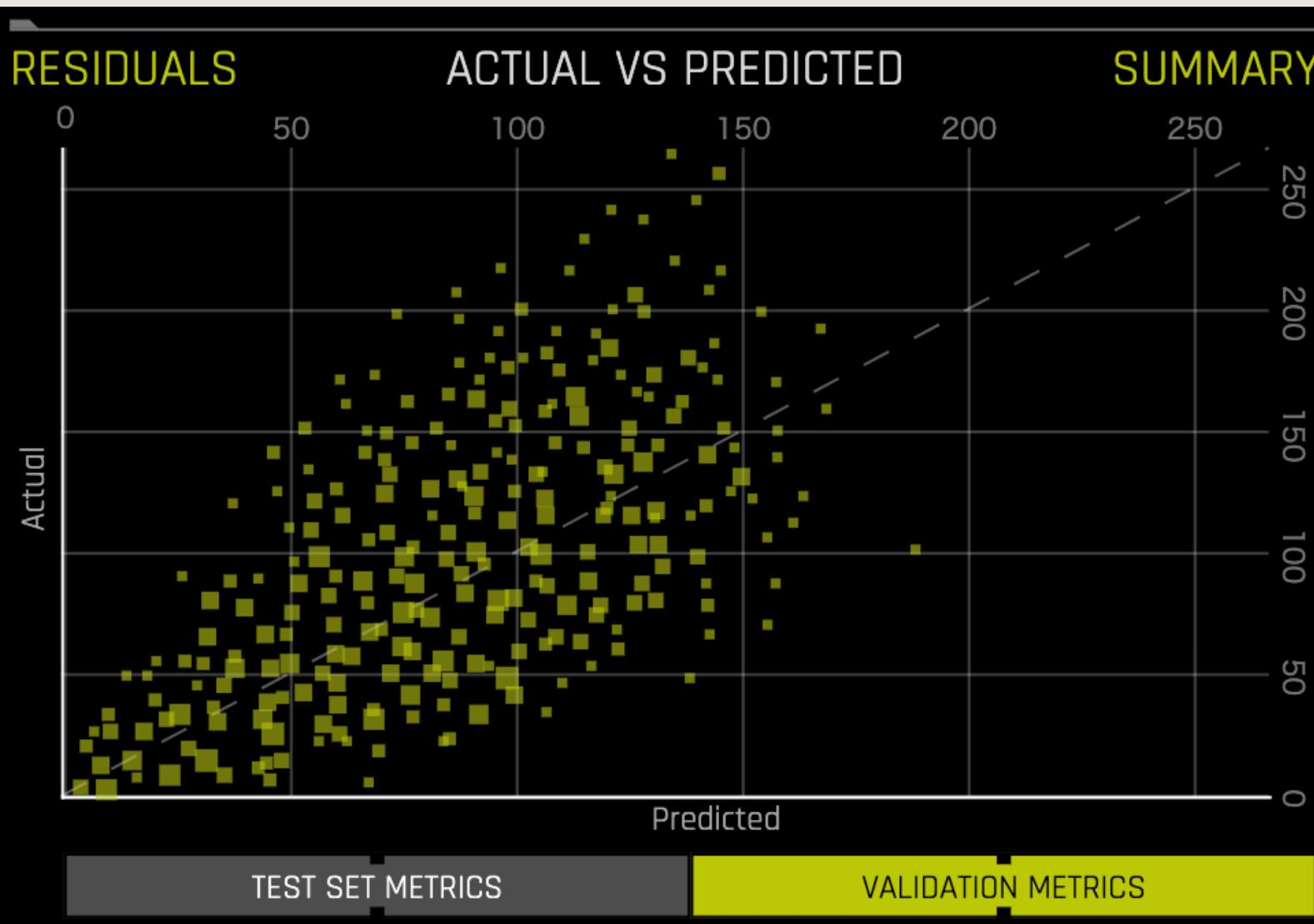
6. 精度確認やモデルの解釈 – 予測精度

代表的な精度評価指標

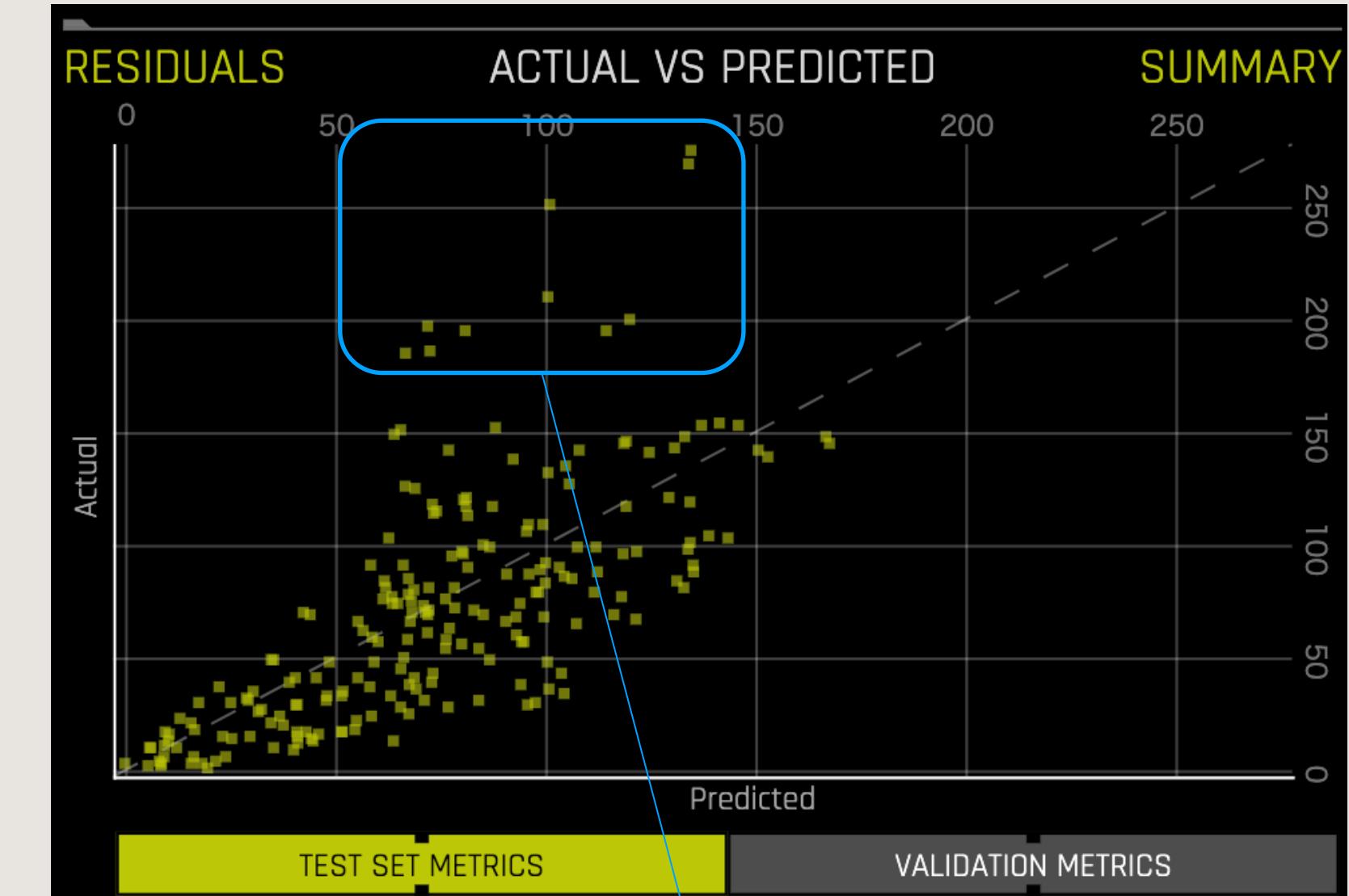
	検証	テスト
RMSE	31.00805	37.88478
MAE	23.41947	26.94281
R2	0.5762772	0.484234

- 今回のExperimentでは、Fold Column (unit_ID) を指定したためか、検証とテストデータの指標間にやや開きがある

予測(横軸) vs 実測(縦軸) – 検証データ



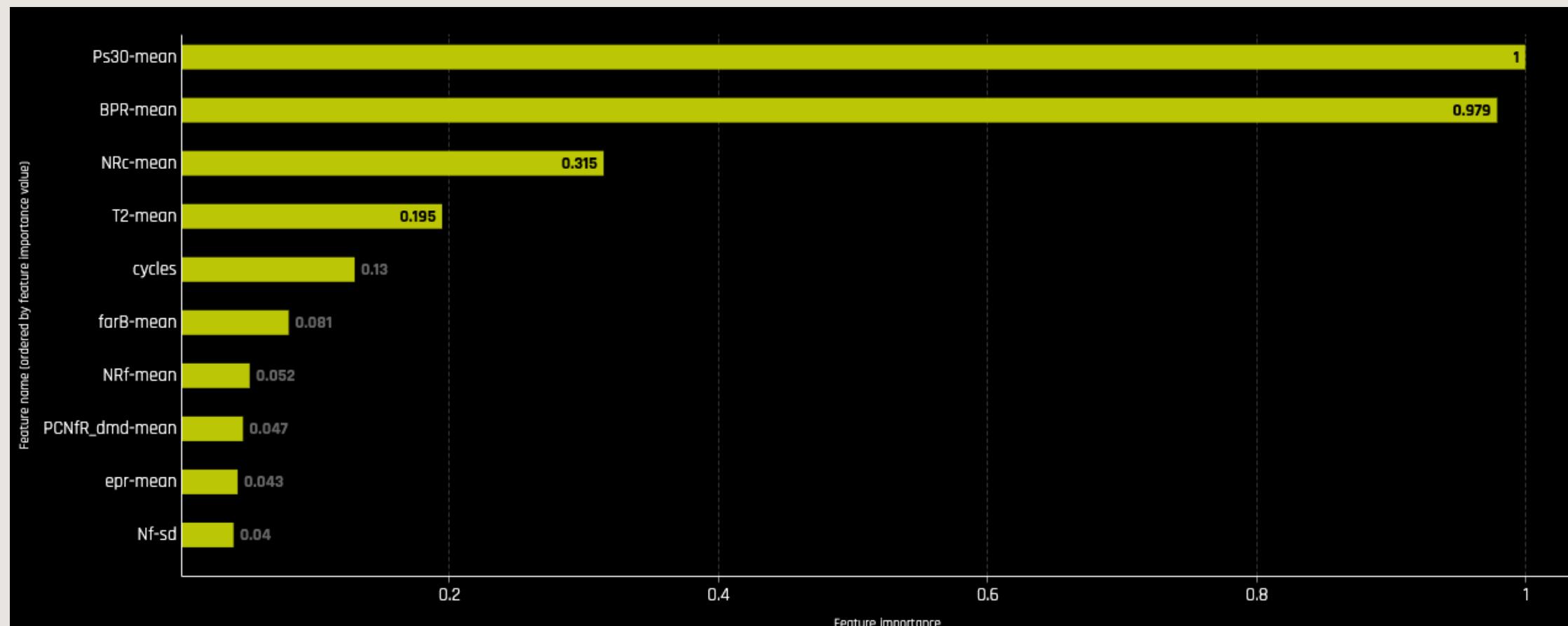
予測(横軸) vs 実測(縦軸) – テストデータ



- 実測値が大きなデータ（実際に寿命が長かったエンジン）の予測がうまくできていない。ロングテール部分の予測を外すのは良くあるパターン

6. 精度確認やモデルの解釈 – モデルの理解

特微量重要度 – オリジナル

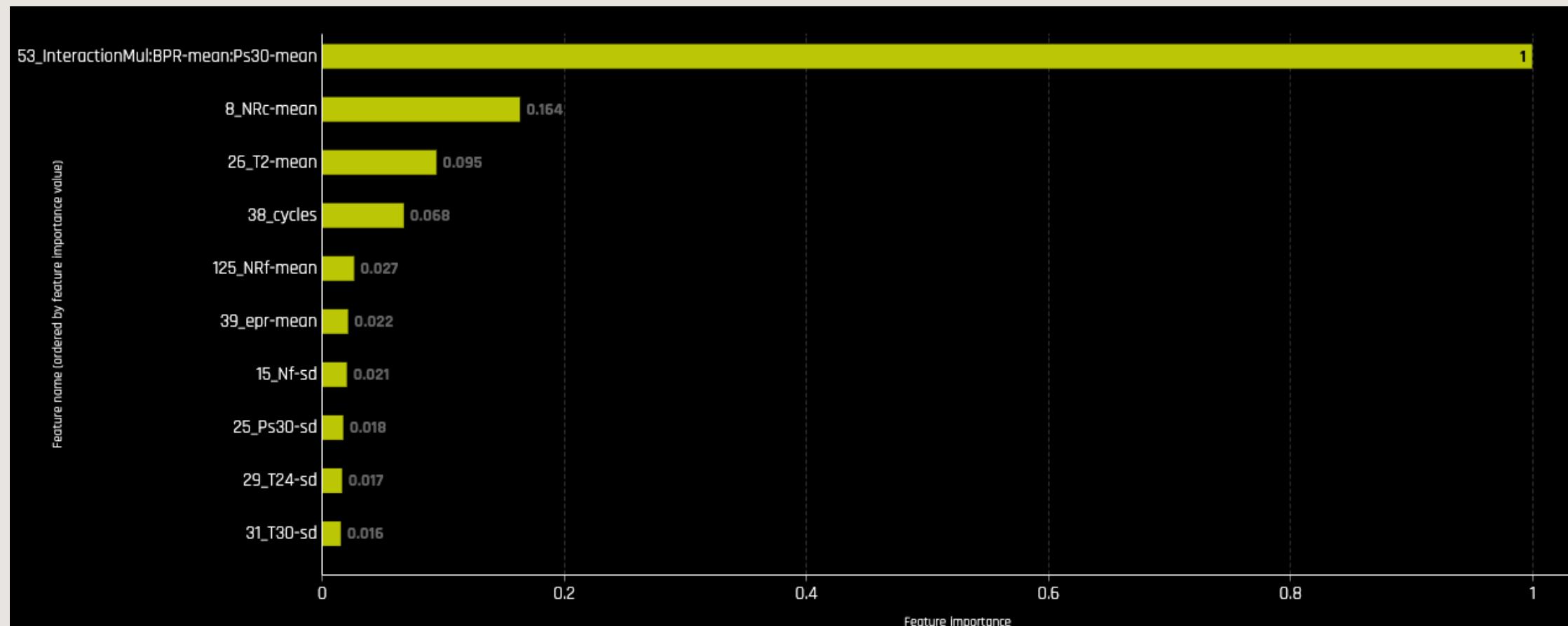


オリジナルの特微量合計は49個

- 24個のセンサー値から平均と標準偏差を作成 (48)
- cycle

- ✓ “Ps30”, “BPR”, “NRc”, “T2”, “cycles”がトップ5寄与度
- ✓ “cycles”より貢献度の高いセンサー値がある
- ✓ 標準偏差(sd)より平均(mean)が重要

特微量重要度 – 特微量エンジニアリング後



- ✓ BPR(平均)とPs30(平均)の交互作用が大きく寄与
- ✓ 今回利用した特微量は数値のみだったため、複雑な特微量エンジニアリング結果は見られなかった

7. 予測結果のビジネス的解釈

H2O.ai

リスクをどう考える？

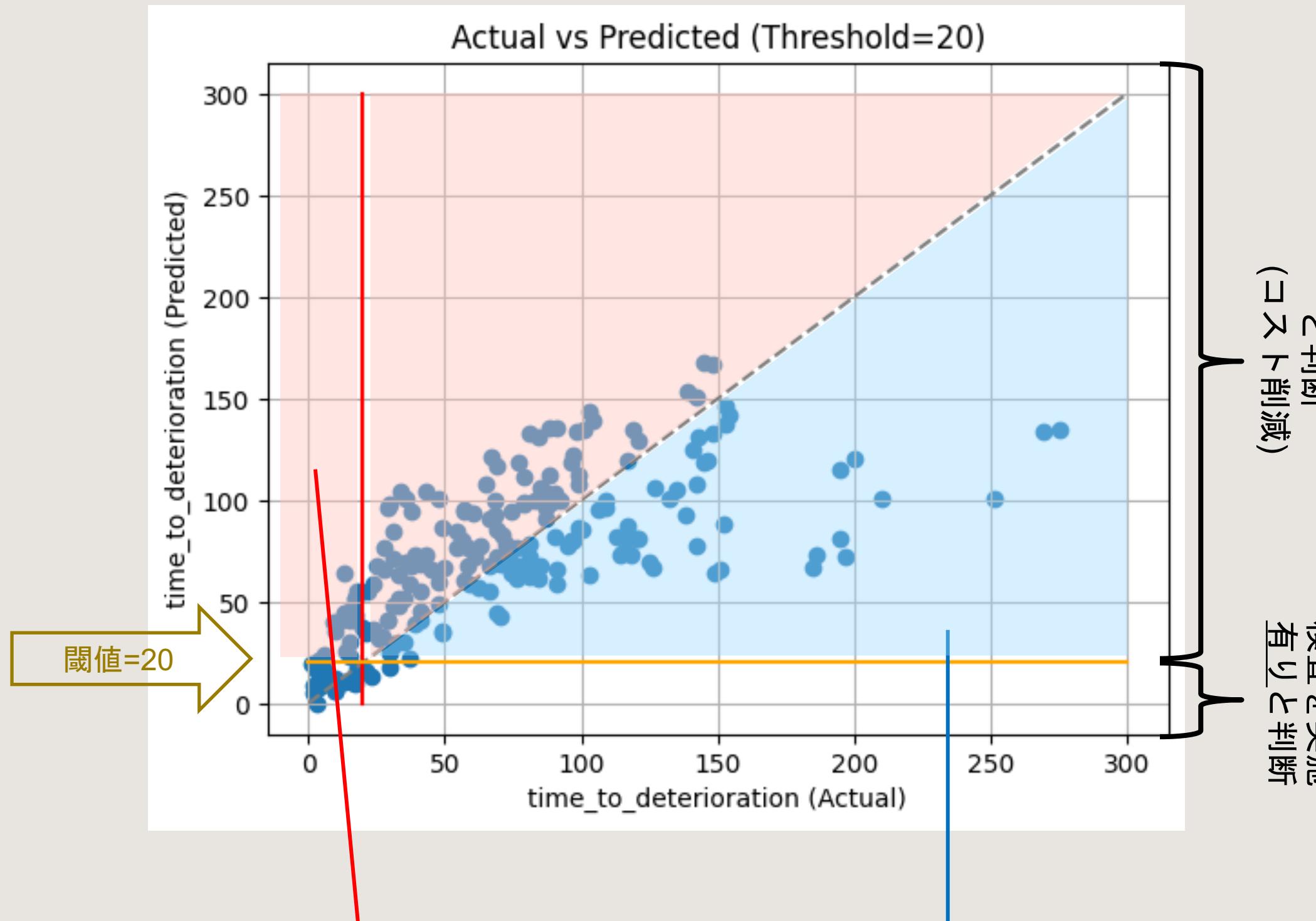
- 検査を実施なかったUnitのうち、「予測>実測」（実際の寿命より、長く予測してしまっている）をリスクと考える場合
- 検査を実施なかったUnitのうち、「予測閾値>実測」（すぐ寿命が来てしまう可能性）をリスクと考えた場合

7. 予測結果のビジネス的解釈

“コスト削減 vs リスク”分析 – 予実プロット：リスクを「予測>実測」と考えた場合

閾値（予想残存交換時期）が20を境界に検査有無を判断した場合

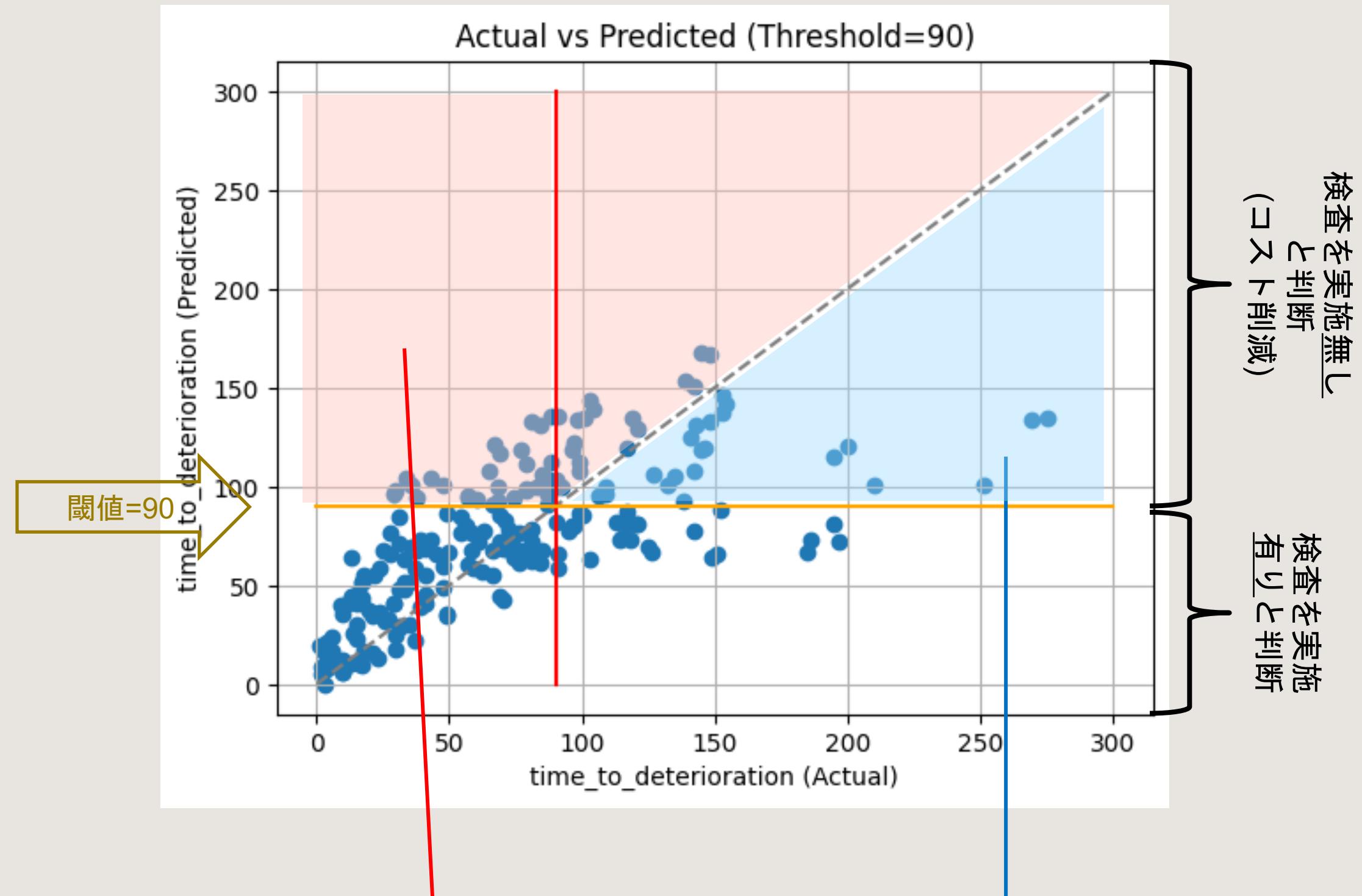
- 予測が20以上は検査の必要なし



閾値：小
➤ コスト削減：大
➤ リスク：？

閾値（予想残存交換時期）が90を境界に検査有無を判断した場合

- 予測が90以上は検査の必要なし



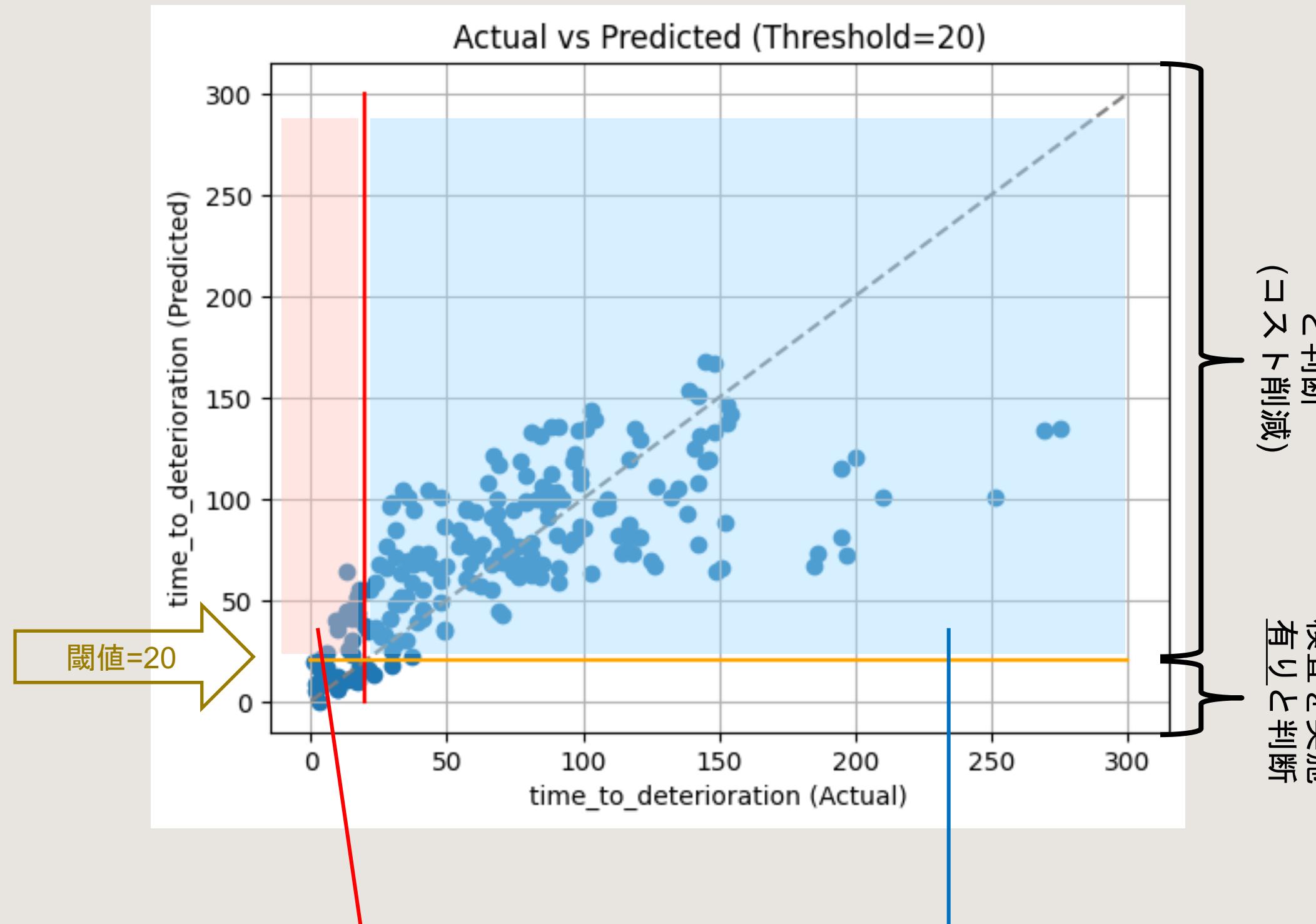
閾値：大
➤ コスト削減：小
➤ リスク：？

7. 予測結果のビジネス的解釈

“コスト削減 vs リスク”分析 – 予実プロット：リスクを「予測閾値>実測」と考えた場合

閾値（予想残存交換時期）が20を境界に検査有無を判断した場合

- 予測が20以上は検査の必要なし

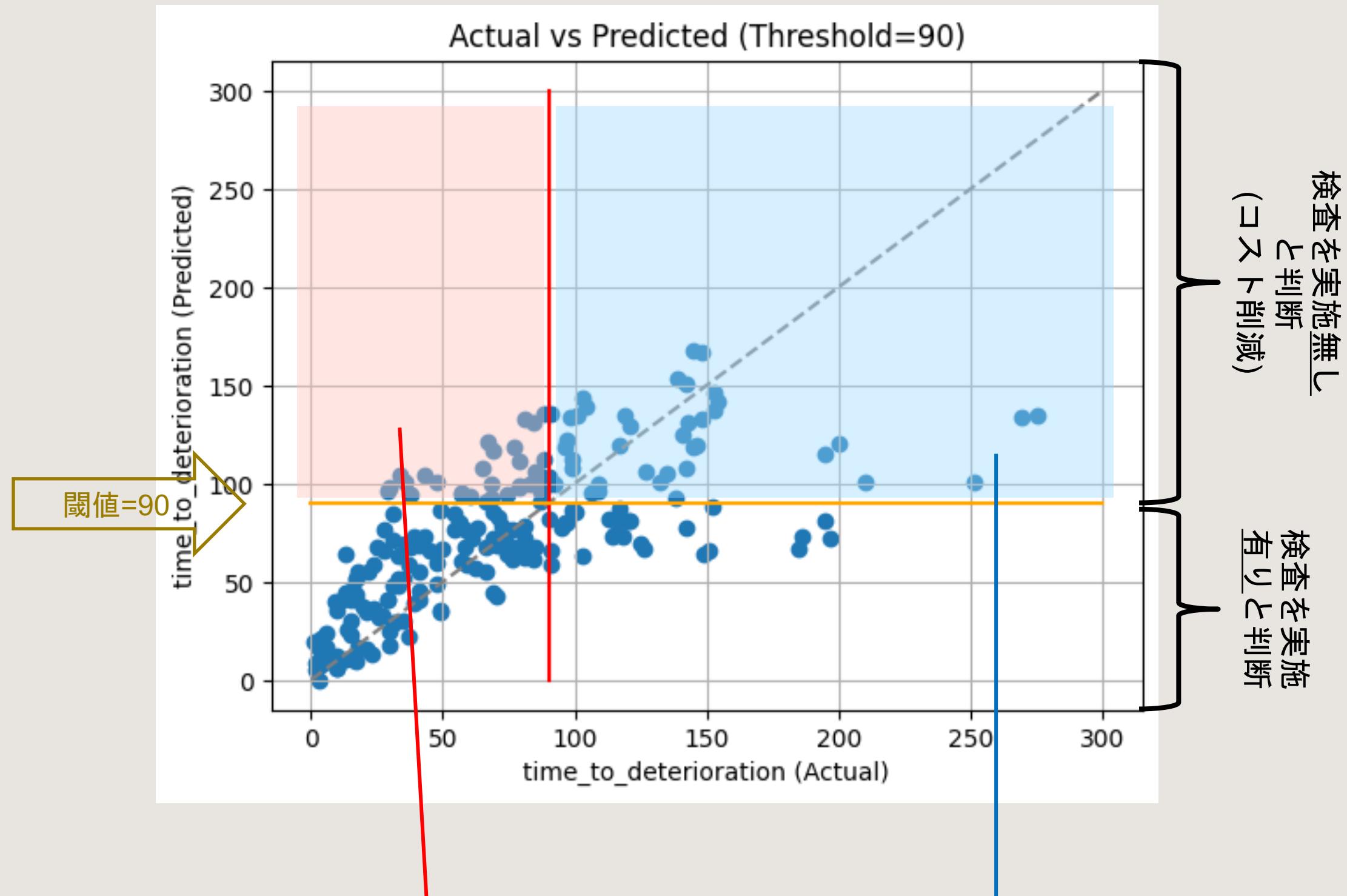


予想閾値より実際の交換
時期が早かった
(リスク)

- 閾値：小**
- コスト削減：大
 - リスク：？

閾値（予想残存交換時期）が90を境界に検査有無を判断した場合

- 予測が90以上は検査の必要なし

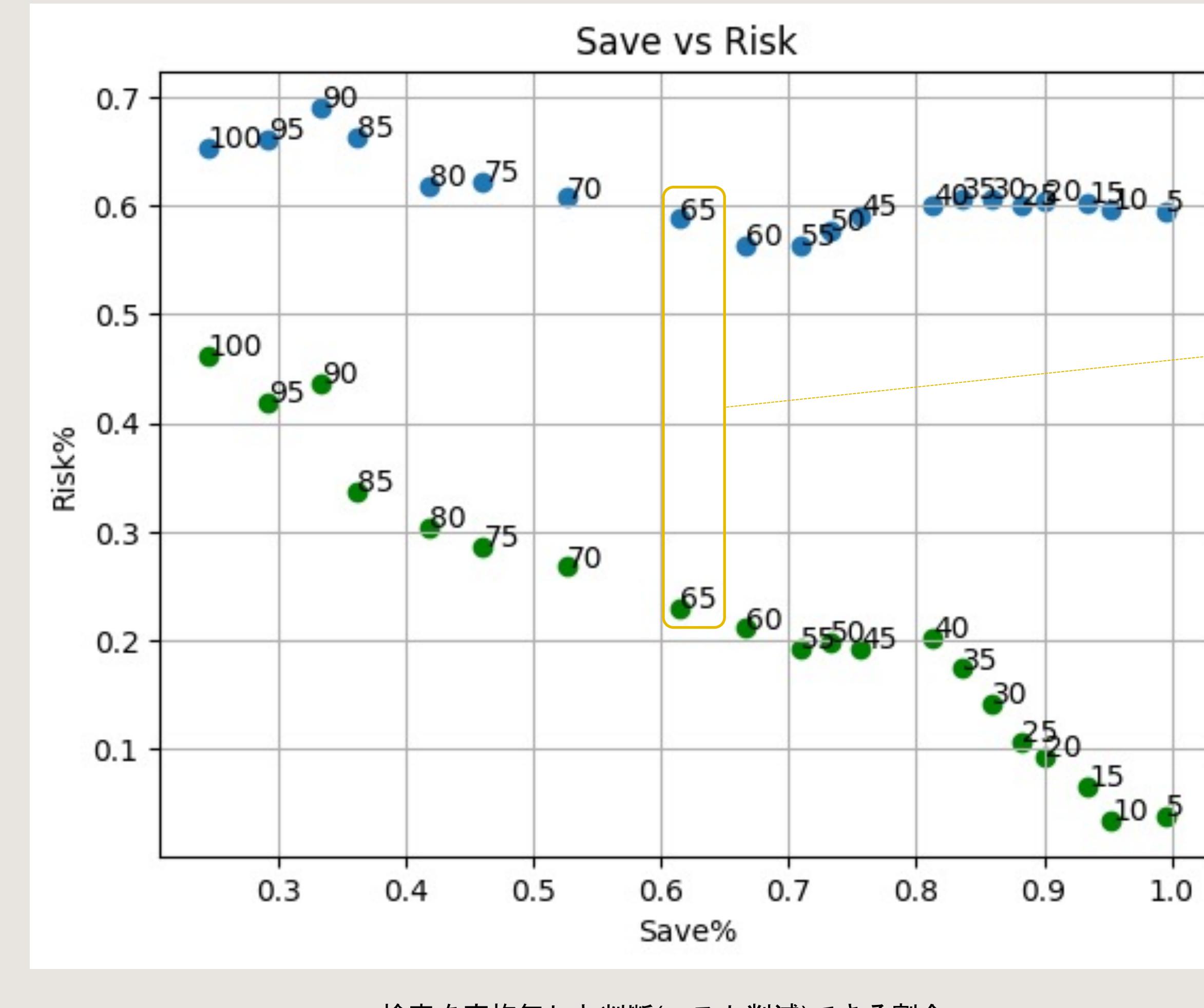


予想閾値より実際の交換
時期が早かった
(リスク)

- 閾値：大**
- コスト削減：小
 - リスク：？

7. 予測結果のビジネス的解釈

“コスト削減 vs リスク”分析 – 閾値を変化させた場合のコスト削減とリスクの散布図



青 : 「予測>実測」リスク
緑 : 「予測閾値>実測」リスク
散布図のラベルは閾値

例えば、65を閾値とした場合

- 61.5%のコスト削減
- 58.1%「予測>実測」リスク
- 23.0%の「予測閾値>実測」リスク

- ✓ 大きなコスト削減（短い閾値）をしたからと言って、リスクが上がるわけではない
- ✓ リスクの最小化ができると、コスト削減の判断が実施しやすい
- ✓ リスクを最小化するのは、モデル予測精度の向上が必要

検討すべき課題

H2O.ai

予測精度の向上

- ✓ 特徴量エンジニアリングの精緻化。今回は過去25 Cycleの各種センサーの平均値と標準偏差のみ使用。ドメイン知識や詳細なデータ分析（MLIなども参考）を踏まえ、その他の特徴量を検討

予測精度の評価

- ✓ 評価指標の選択。今回はRMSEを利用。KPIと直接相関がある指標は？今回の場合、実際より交換時期を短く予測すること（下に予測）はリスク観点から問題なし（ただし、短く予測しすぎるとコスト削減できない）
- ✓ “リスク”の判断方法を最終的にどうするか？

【Appendix】データやコードについて

Original Data Source

- https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data

データ、データ加工や分析のコード

- https://github.com/yukismd/SensorData_PdM/tree/main

参考：結果の表現

予測結果を過去実績含めダッシュボードや動的なアプリケーションとして提供することで、ユーザー（ビジネスサイド）とのコミュニケーションを円滑化



各エンジンの状況と予測結果のモニタリング



特定エンジンの予測詳細

各センサー値の時系列プロット