製造 IoT データ・生体データ などのセンサデータ分析最前線

株式会社NTTデータ数理システム
数理工学部
大場　拓慈

oba@msi:~$ whoami

ホーム / 数理システムのつよみ

人類の英知をフル活用し、AIプロジェクトを生産的に推進する

数理工学部

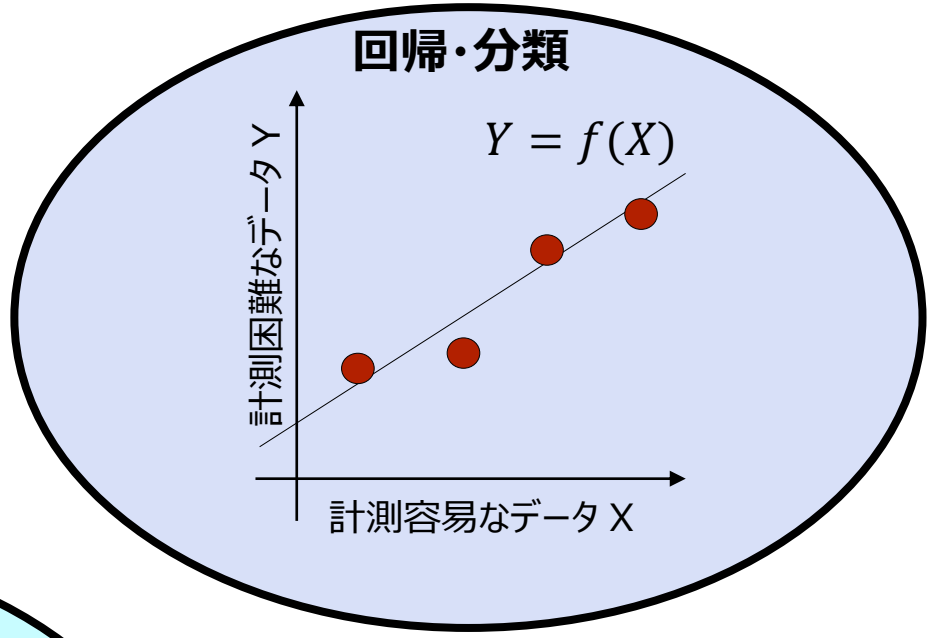
大場　拓慈

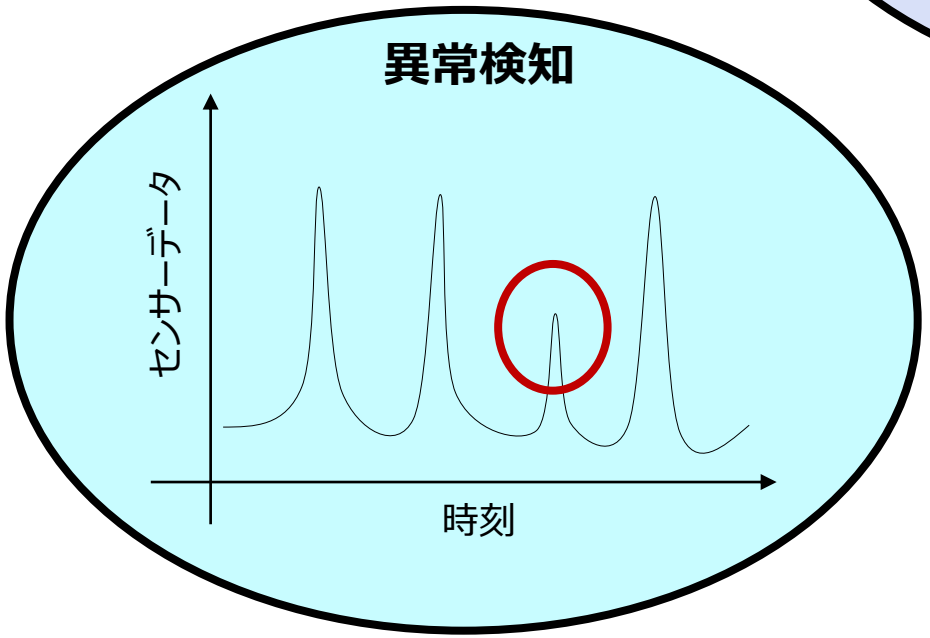
学生時代の専攻は数学（学部）と数理生物学（修士～博士）。プログラミングはほとんど未経験で2019年にNTTデータ数理システムへ入社。入社後は数理工学部の時系列データ分析を扱うチームにて、お客様の課題解決に向けた手法の提案から実際の分析業務まで、一連の流れとして担当している。最近の技術的興味は不定期計測データに対する分析手法。

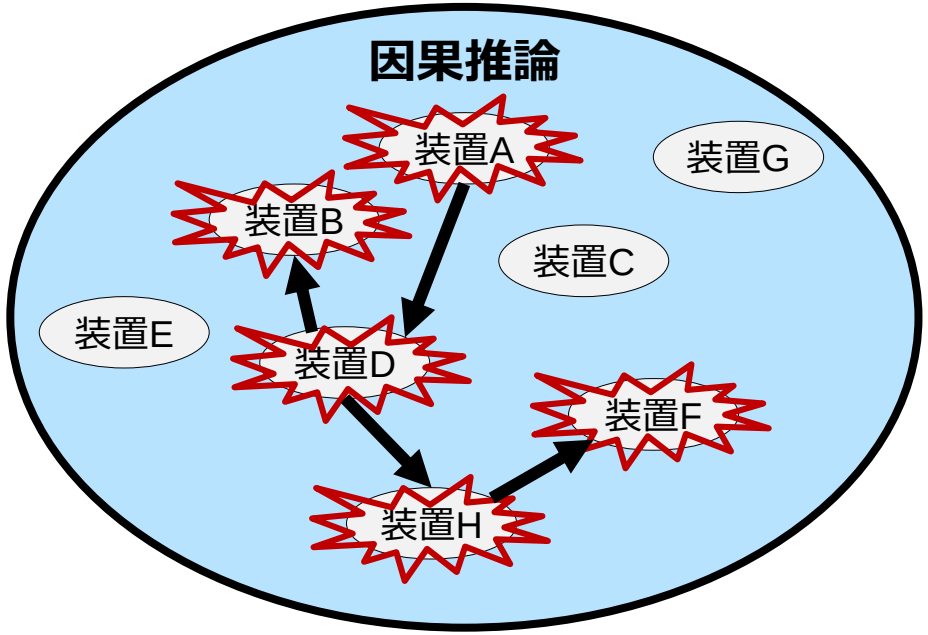
・・・だそうです[1]。

[1] “NTTデータ数理システム 企業サイト/数理システムのつよみ/データサイエンティスト紹介” より引用
https://www.msi.co.jp/speciality/data-scientist06.html

センサーデータの分析事例





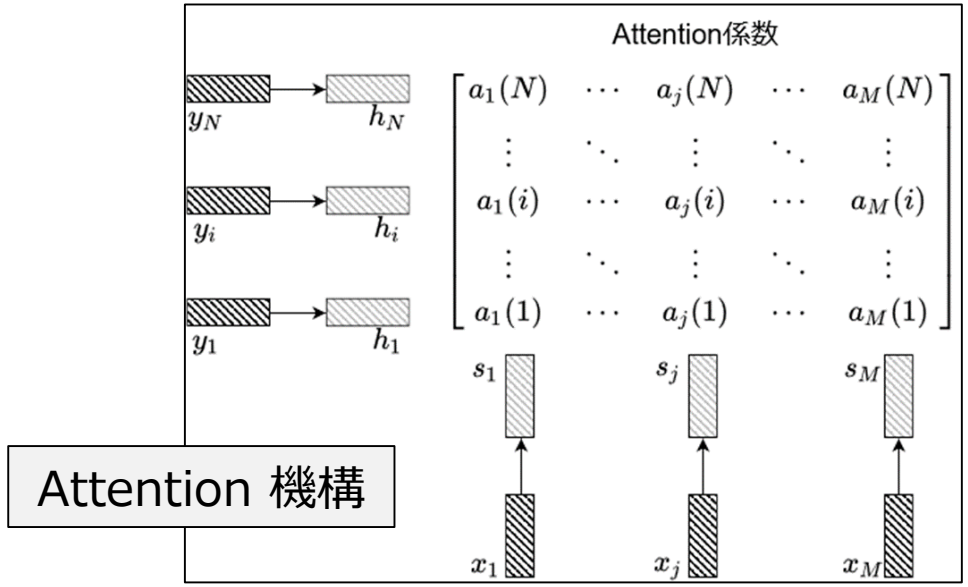


目次

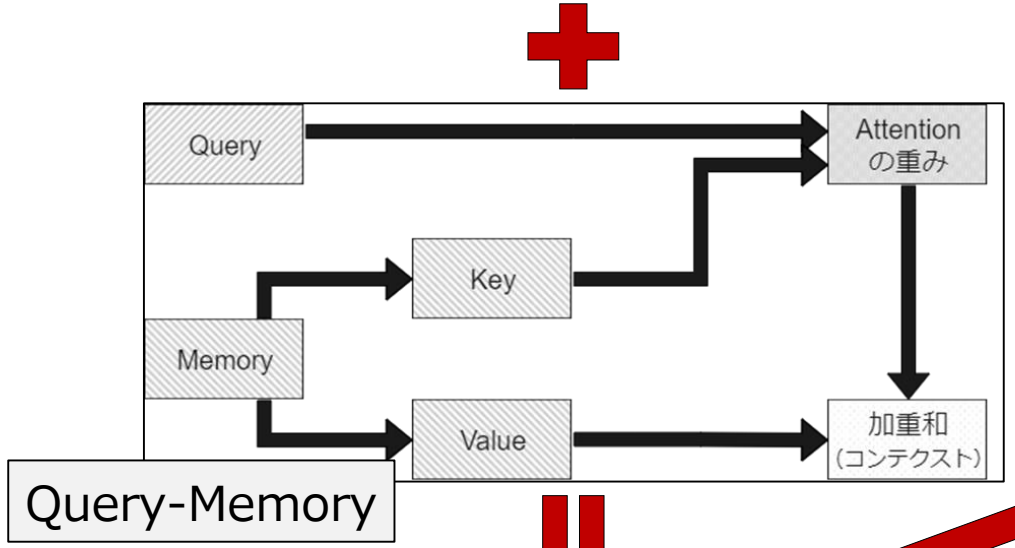
1. “AI”とデータ分析
    - 大規模言語モデルのこれまでとこれから
    - 時系列データ分析
2. データの前処理
    - データファイル整形
    - センサー間の時刻ズレ補正
    - ノイズ処理と欠損処理
    - スケーリングとデータの偏り
3. モデル設計
    - 業務課題→技術課題→問題設定
    - モデルのよさ vs 計算資源
    - モデル設計におけるドメイン知識利用
4. 実用化に向けて
    - 泥臭い処理と泥臭いプログラムとバグ
    - 既製品に任せよう
    - 可視化とドメイン知識利用

§01 “AI”とデータ分析

これまでの大規模言語モデル



Attention 機構



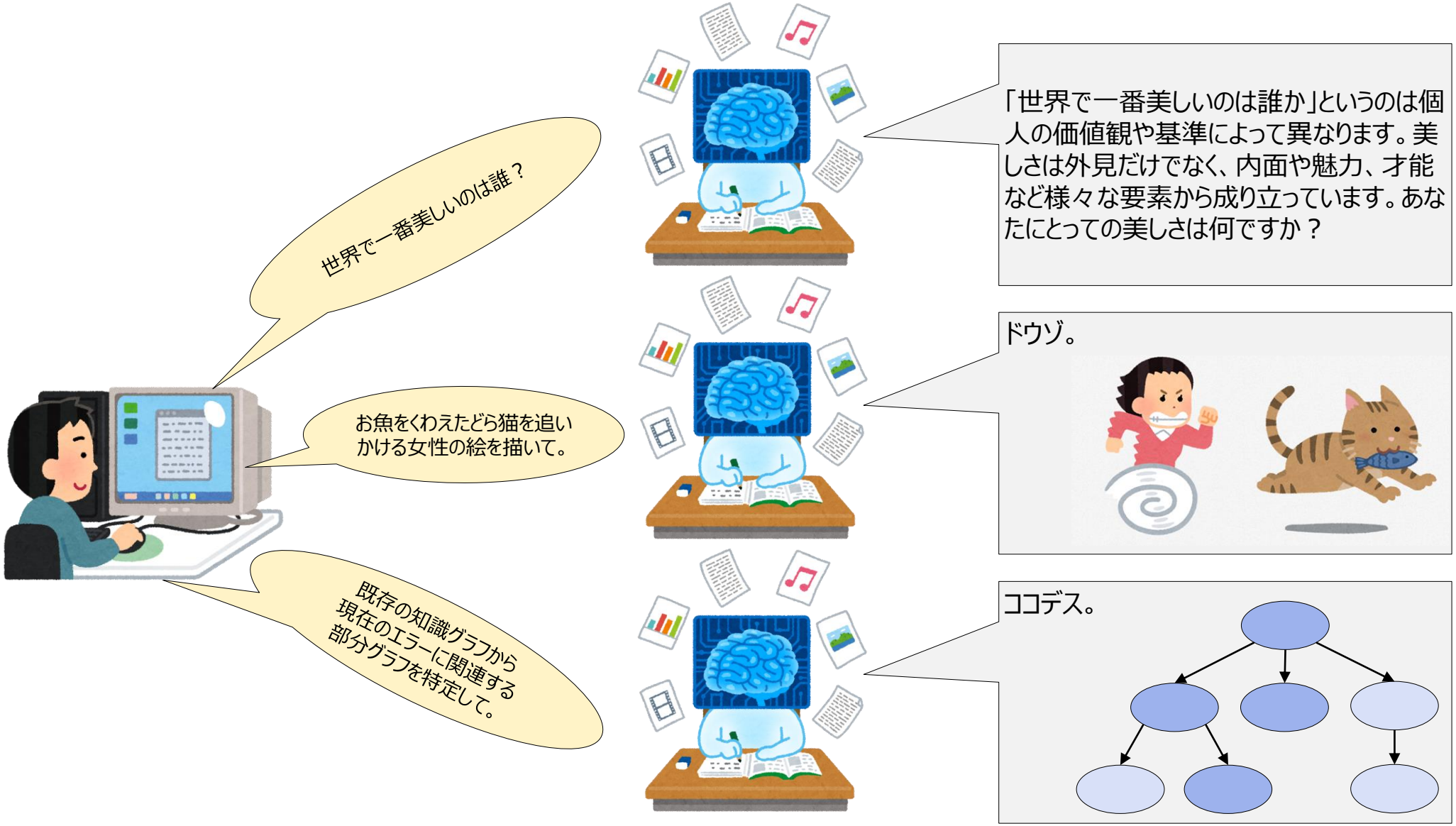
Query-Memory

Transformer

Attention 機構を全面に押し出した
Transformer により一気に高精度化

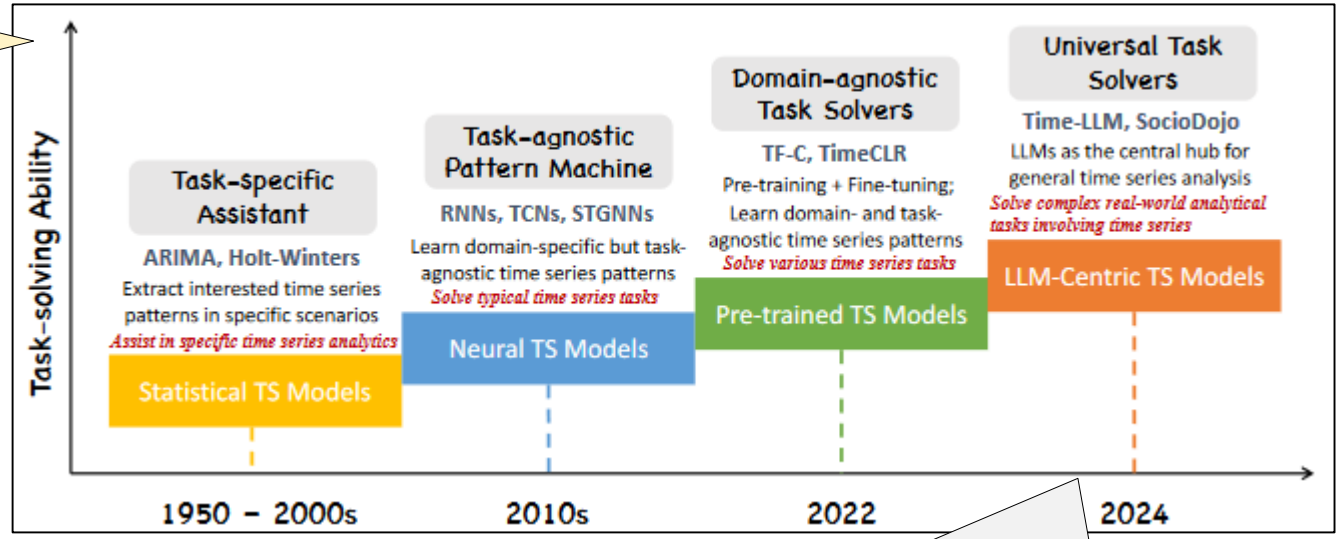
NTT DATA

次々と汎用性を獲得する大規模言語モデル



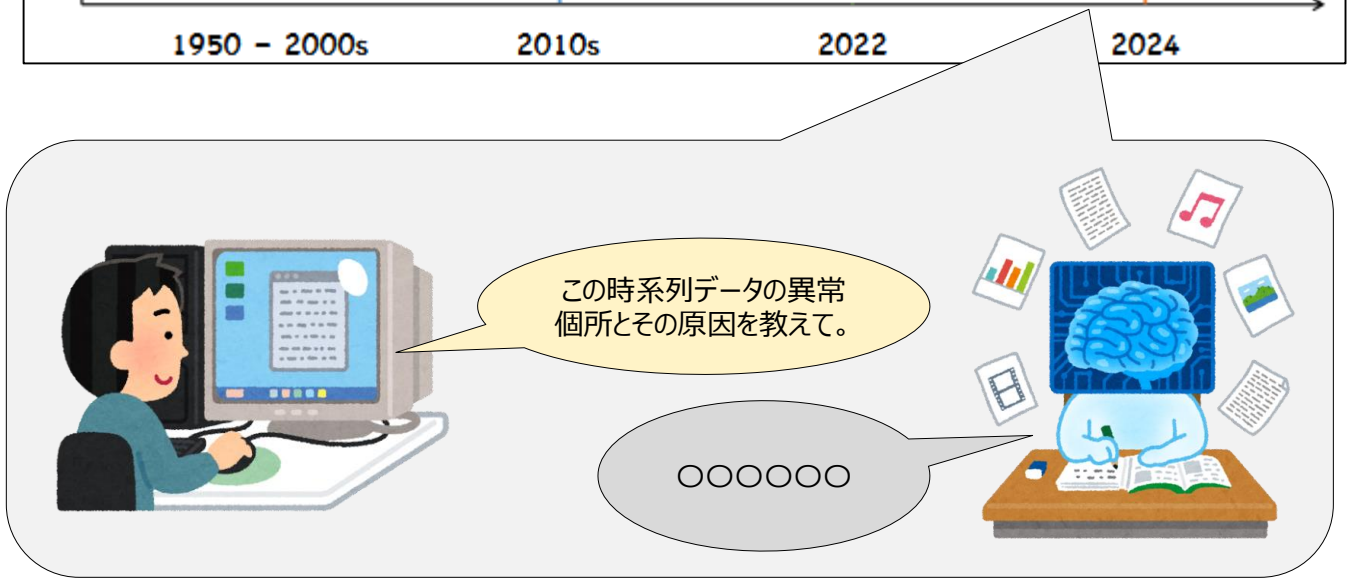
時系列データ分析と大規模言語モデル

時系列データ分析の歴史
文献 [1] Figure 1 を引用



時系列データを LLM で扱うモデル

- 2023年ごろからチラホラ見かける。
  → サーベイ論文は複数あり [1,2]。
- 時系列データをどうやってLLMに入力するかだけでも手法が様々で発展途上。
- ましてドメイン汎化された汎用ツールとなると相当先になるのでは。



[1] Jin et al. “Position: What Can Large Language Models Tells Us about Time Series Analysis.” arXiv.2402.02713 (2024)
[2] Zhang et al. “Large Language Models for Time Series: A Survey.” arXiv.2402.01801 (2024)

§02 データの前処理

計測したての生データをどう扱うか？

IoT（モノのインターネット）

計測

入力

ChatGPT

Alkano

データ活用の確かなパートナー

scikit learn

出力

?

NTT DATA

データ整形

生データ
（CSVフォーマット）

timestamp	I	II	III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6

人手による追記
自動追記

扱いやすく整形する
必要がある。

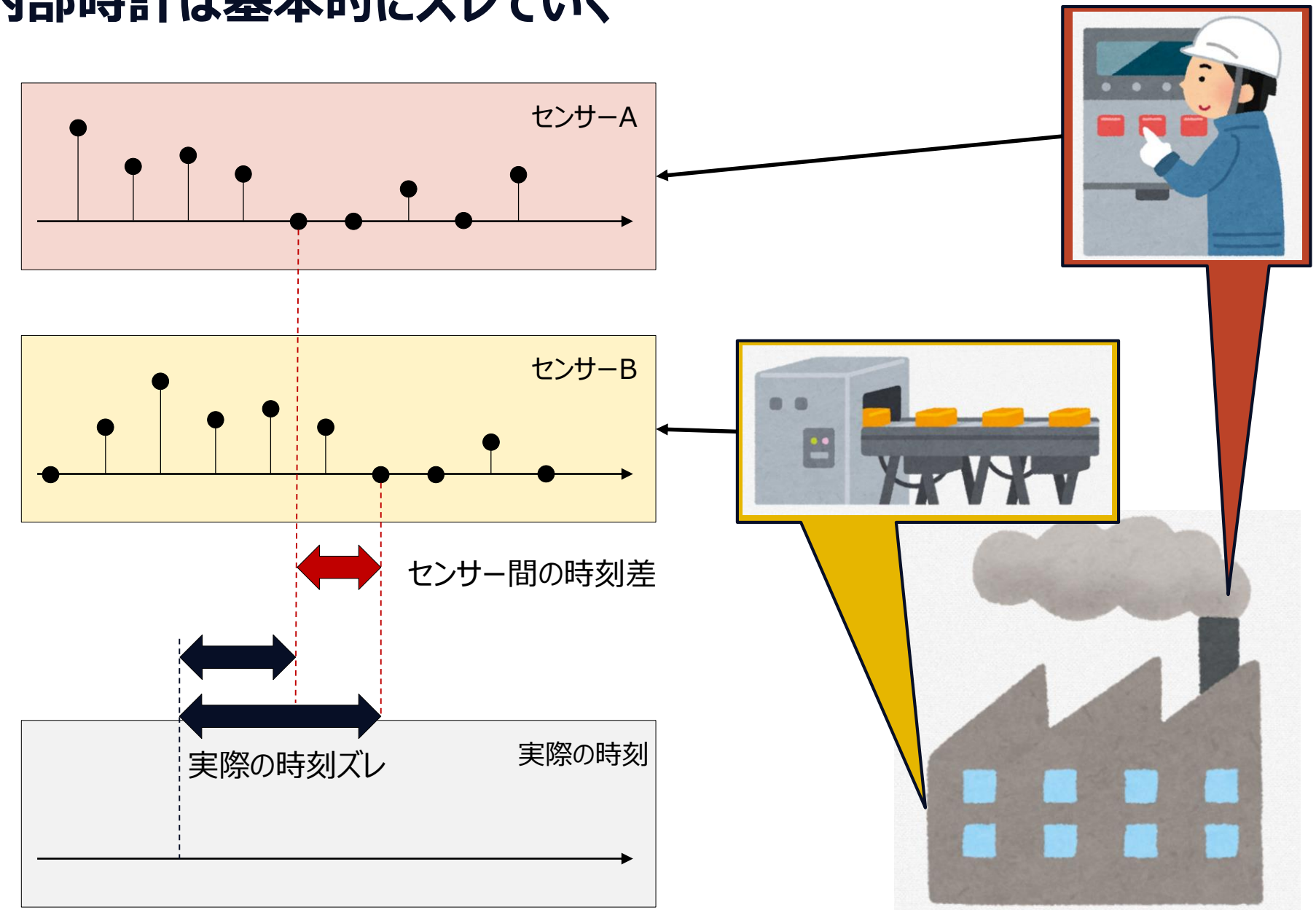
テーブル化など

患者の情報を追記

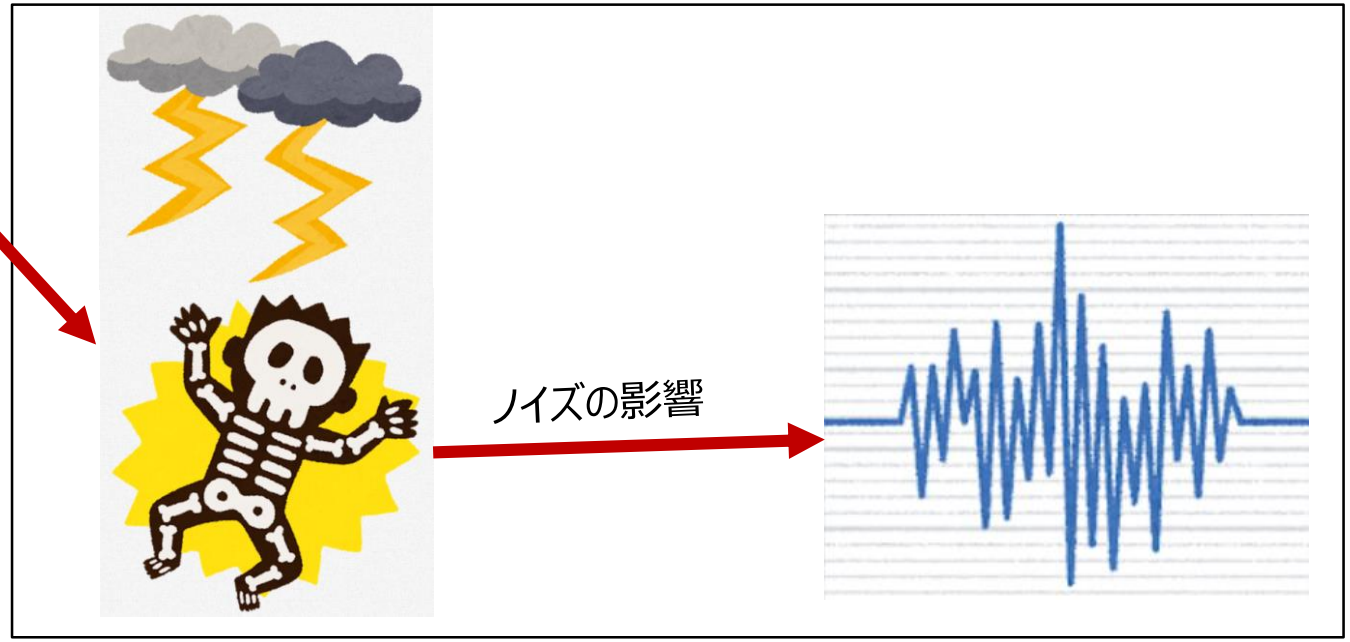
カルテID	11
患者名	数理花子
性別	女
満年齢	68
計測日	2024/10/18
身長	158.2
体重	52
体温	36.5
すごい健康診断値	1083
よくわからん異常値	98.00%

timestamp	I	II	III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6

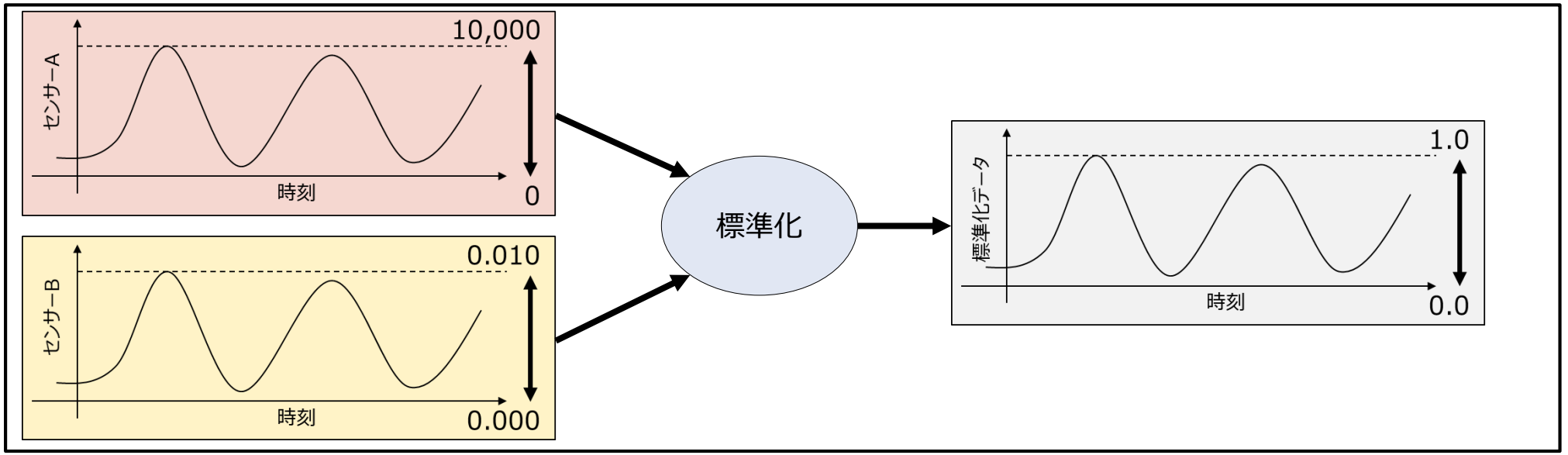
センサー内部時計は基本的にズレていく

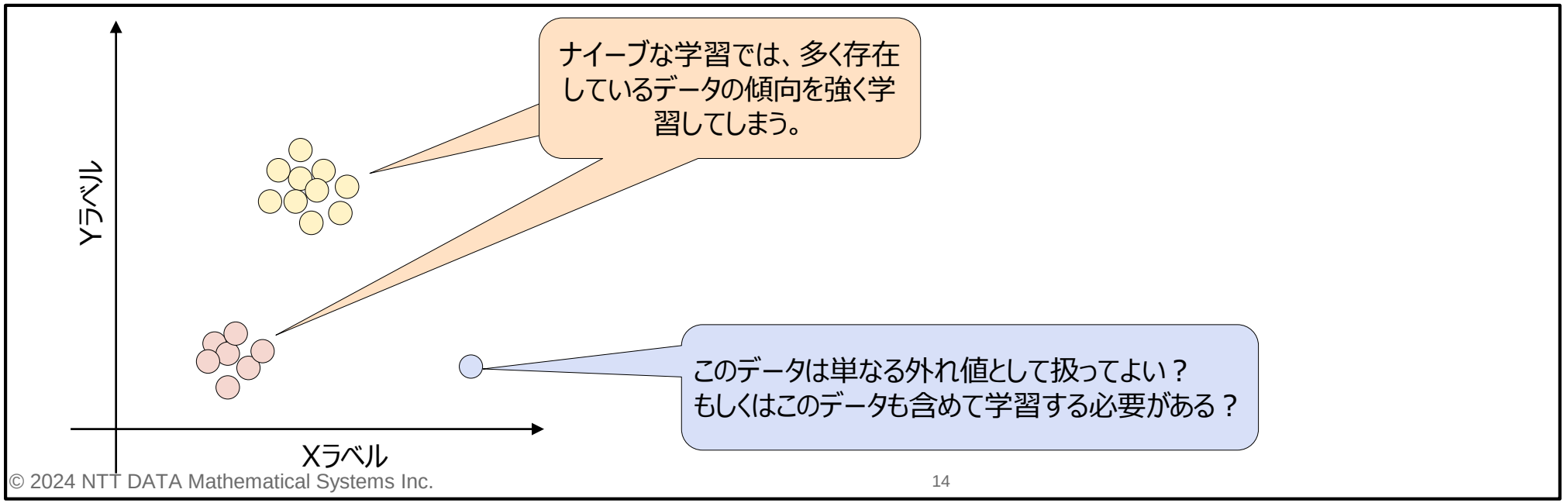


データ欠損とノイズの取り扱い



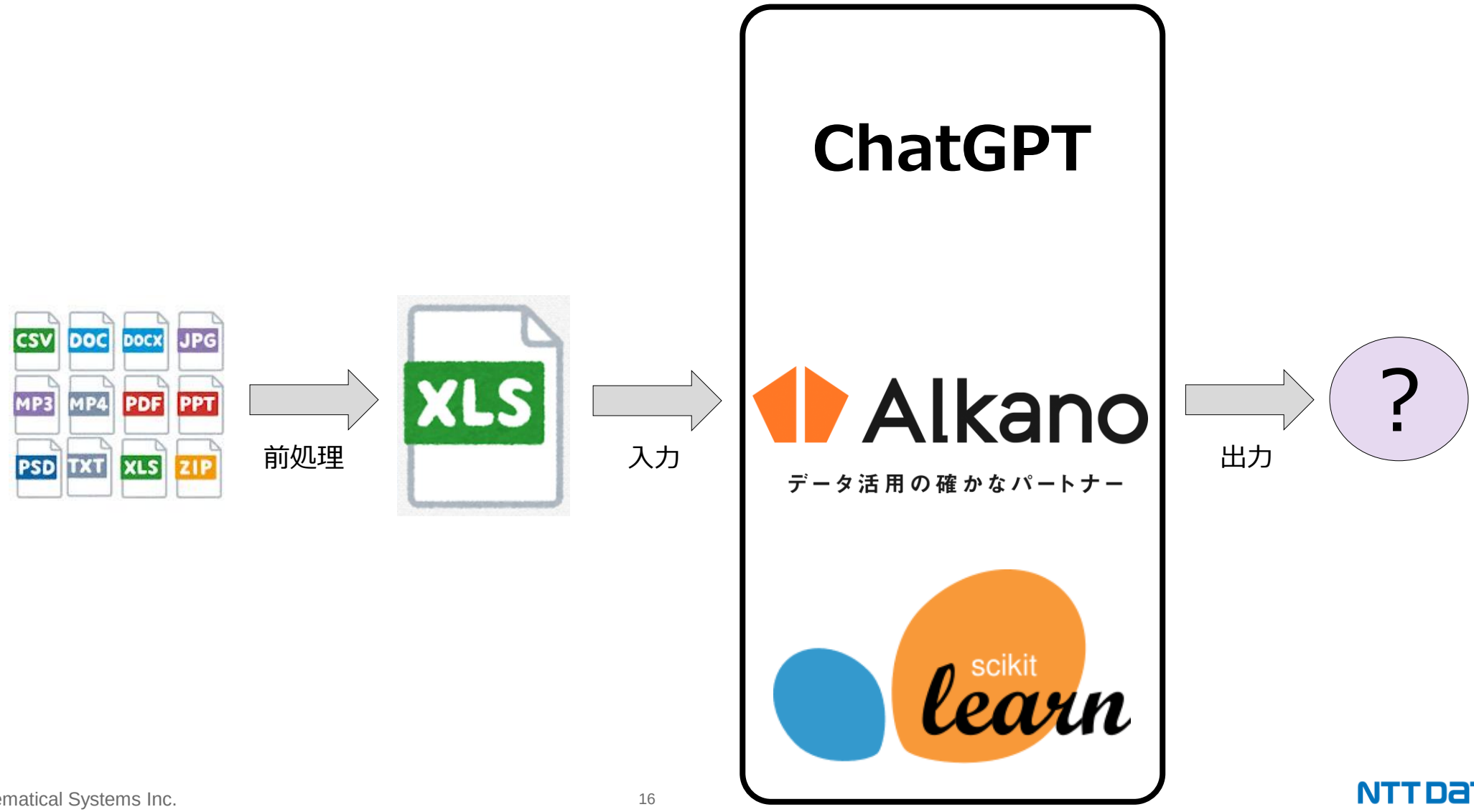
スケーリングとデータの偏り





§03 問題設定とモデル選択

業務課題を解決してくれる汎用モデルが存在するか



業務課題を技術課題に落とし込む

センサーデータを使って装置の故障を早めに知りたい。
今は作業員が故障を発見してから管理職経由で連絡が来るため遅い。

業務課題

Q. 実際に故障した細かい日時は分かっている？
A. おおよその日時ならわかる。

Q. いまはどのくらい遅れている？
A. 10時間から長いと丸2日間とか

Q. センサーデータをヒトが見れば故障を判定できる？
A. 故障しているときは普段とは異なる推移をしている印象はある。

技術的には二種類の方法がある。

- ◆ 平常時のデータの分布から大きく外れたデータを計測したらアラートを出す。
- ◆ 時系列予測モデルを立てて、予測値から大きく乖離した値を計測したらアラートを出す。

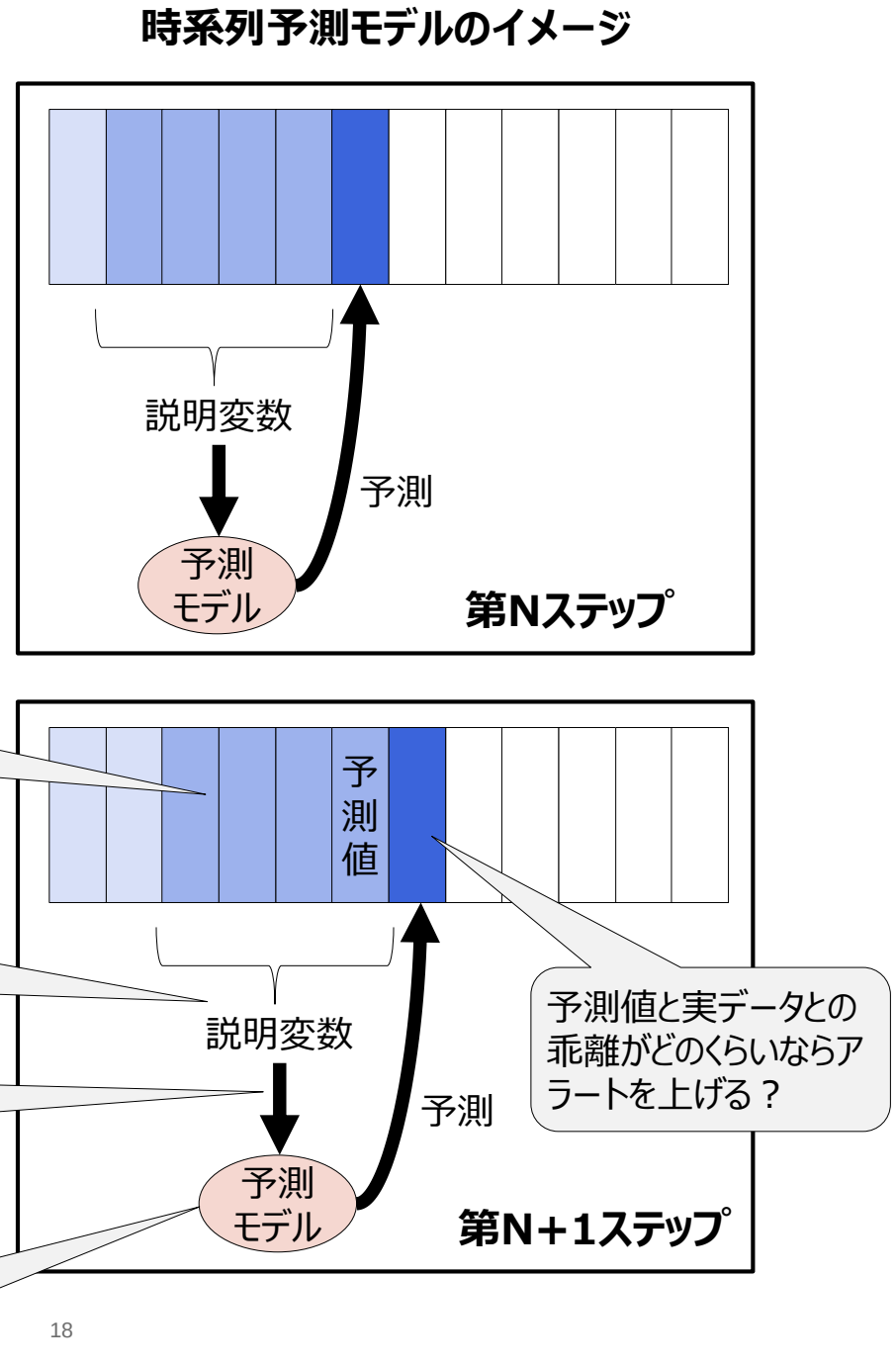
技術課題

NTT DATA

技術課題から問題設定へ

技術的には二種類の方法がある。
- ◆ 平常時のデータの分布から大きく外れたデータを計測したらアラートを出す。
- ◆ 時系列予測モデルを立てて、予測値から大きく乖離した値を計測したらアラートを出す。

技術課題

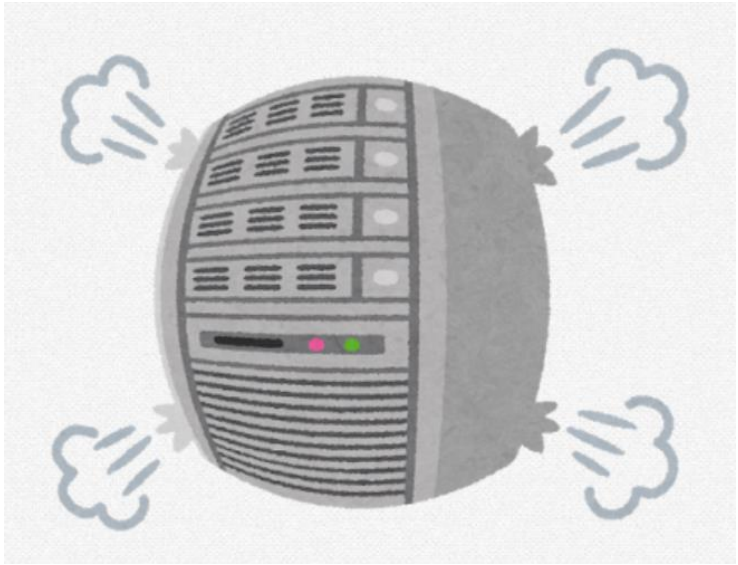


モデルの良さ vs 計算資源

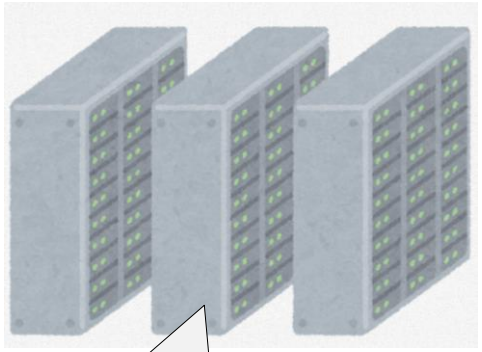
◆ 学習がうまくいって素晴らしい精度が得られたら万事解決か？
- 高いサンプリング周波数でのリアルタイム予測できないと・・・
- AWSなどのレンタルサーバに載せるのでお金が・・・
- 小さいデバイスに載せたいのでそんな計算性能は期待できない・・・

◆ 学習済みの深層学習モデルを軽量化することで対応できる範囲ならよい
- 深層学習ならではの強みとして転移学習やファインチューニングができる可能性も

◆ ただその程度の軽量化では対応できないくらいの省メモリ・省CPUが求められる場合も多い
→ モデルレベルの軽量化というより、ドメイン知識を利用したモデル設計レベルで軽量化を図る

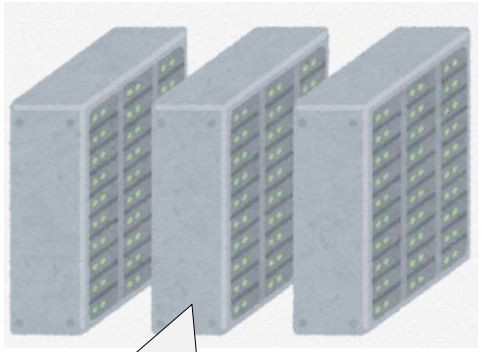


モデル設計におけるドメイン知識利用



判明している法則性を
含む大量のデータ

モデルに明示的な法則性を与えず、
法則性をデータから学習してもらう。



判明している法則性
以外の部分のデータ

判明している法則を明示的にモデル
に与えた上で、法則以外の部分を
学習させる。

§04 実用化に向けて

泥臭い処理のプログラムはやはり泥臭い

```
# シンプルな処理

df = pd.read_csv(ifp)
X, y = df[xcols],
df[ycol]
model = Model()
model.fit(X, y)
```

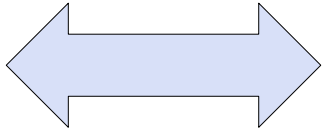
一見ほとんど同じ・・・？

```
# 泥臭い処理

df = merge_data(ifp_list)
df = preprocess_data(df)
X, y = create_dataset(df)
model =
MyModel(params_list)
model.fit(X, y)
```

各関数・各クラスがそれぞれ
1,000行以上あることもザラ

泥臭い処理はバグを生みやすい

```
#  シンプルな処理

df = pd.read_csv(ifp)
X, y = df[xcols],
df[ycol]
model = Model()
model.fit(X, y)
```



一見ほとんど同じ・・・？

```
#  泥臭い処理

df = merge_data(ifp_list)
df = preprocess_data(df)
X, y = create_dataset(df)
model =
MyModel(params_list)
model.fit(X, y)
```

各関数・各クラスがそれぞれ
1,000行以上あることもザラ

プログラム仕様は？
　→ 書きながら決まる
どうやってできあがるの？
　→ 試行錯誤の末に得られたコードの集合体
テスト駆動開発・・・
　→ そもそも仕様書すら事前には書けない

一区切りついたタイミングでコード群を整理することもテストを書くこともあるが、それはあくまで区切りのタイミング
→ **試行錯誤している最中にはバグに気づきづらい**

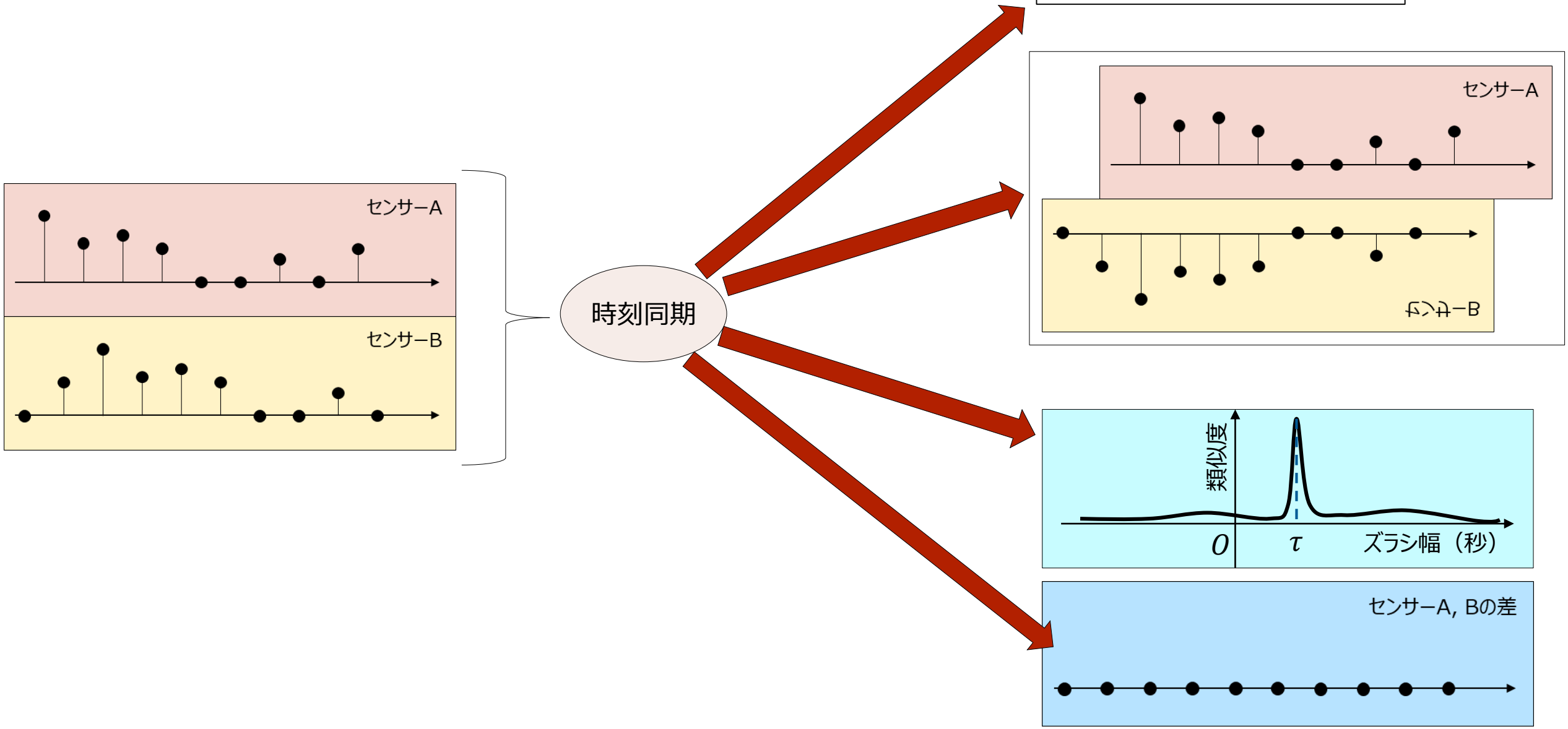
既製品の安心感

企業の製品・有名ライブラリは開発者やユーザがテスト済
→ 自前コードと比べて信頼度が段違い。
→ 使える処理には積極的に適用したい。

※製品の宣伝ではありません。~~おしろ分析のお仕事ください。~~

NTT DATA

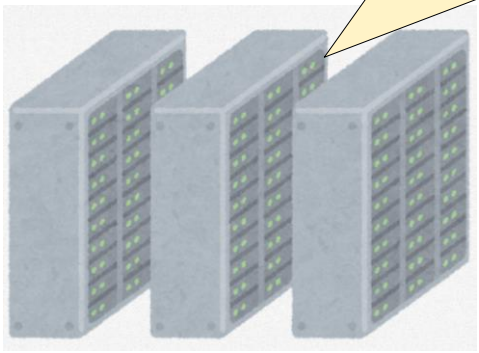
処理結果を“見える化”しよう～基礎編～



処理結果を“見える化”しよう～応用編～

前処理

データセット
生成

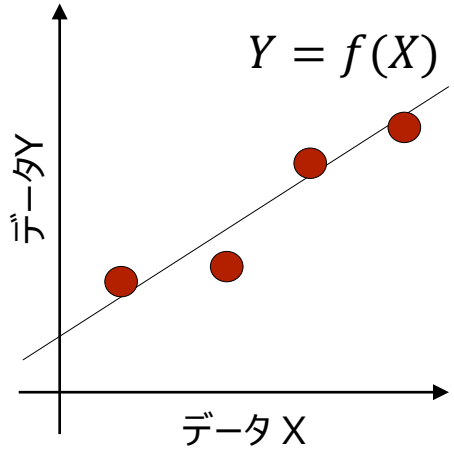


ドメイン知識から Y = f(X) という
法則があることは分かっているが
モデルに直接教えるのは難しい。

機械学習モデルにデータ
セットを学習させる。

Y = g(X) という法則
がありそう・・・

予測結果の可視化による
ドメイン知識に照らした
妥当性検証が必要



分析作業を滞りなく進めるために

◆ 結論がありふれた内容に過ぎるが

- お仕事ごとに、お客様の領域の知識を概要だけでも把握しておくことで“直感”を養う。
- 分析結果の可視化とお客様からのコメントによるフィードバックループが大事。
  - ✓ モデルに直接組み込めそうなドメイン知識がないかをお客様とともに考える。
  - ✓ モデルの予測がドメイン知識に照らして妥当かどうか確認する。

分析のお仕事ください！


NTT DATA