

説明可能な機械学習

== 統計的学習によるハイパフォーマンスモデル ==

2024年10月

徐 良為

SLW代表、(株)NTTデータ数理システム 顧問

e-mail : liangweixu2205@gmail.com

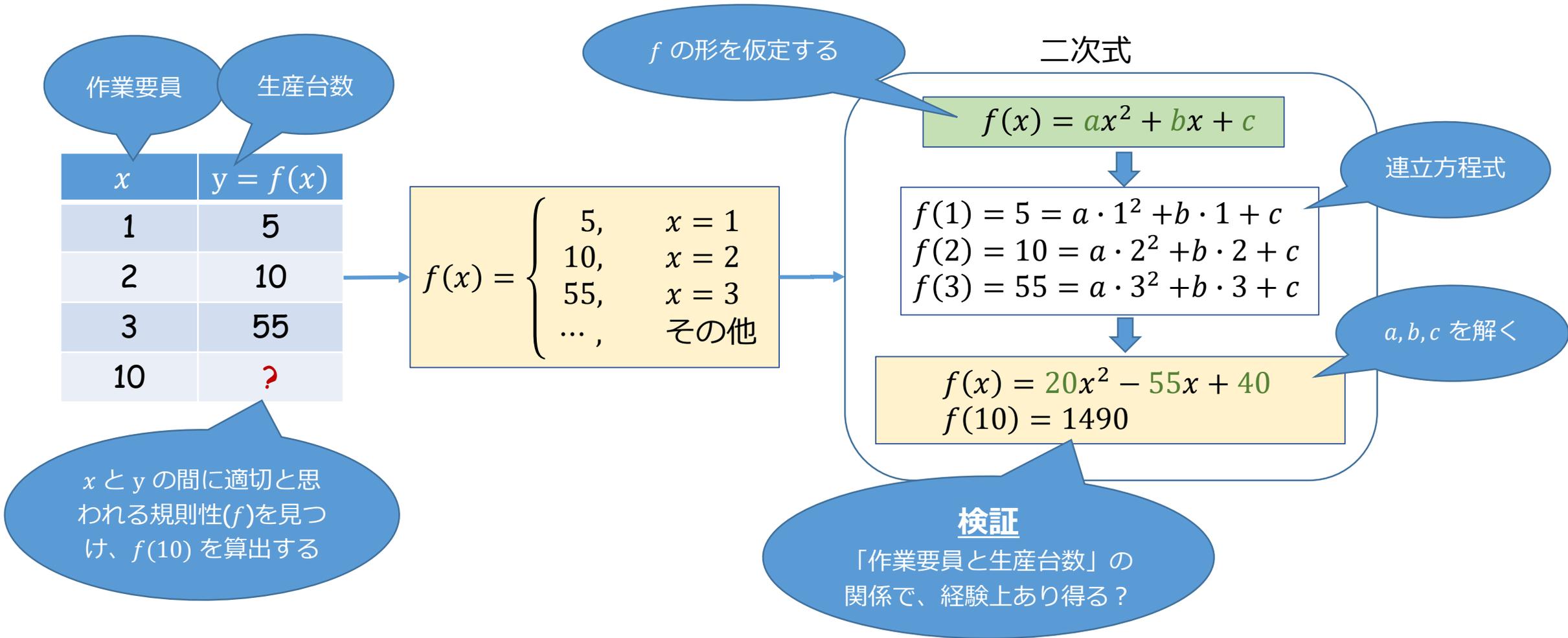
youtube : https://www.youtube.com/@ai_dentaku

目次

1. 機械学習の役割と課題
2. 説明可能な機械学習、「AI電卓」の技術概要
3. パフォーマンス検証事例
 - ① 事例1 Kaggle = House Price 不動産価格予測
 - ② 事例2 Kaggle = Store Sales POS 時系列予測
4. まとめ
5. 将来展望

1. 機械学習の役割と課題

パズル問題



機械学習

x	$y = f(x)$
1	5
2	10
3	55

【 Step 1 】 関数 f の形を仮定する

【 Step 2 】 $f(x)$ と正解値 y 間の誤差を定義する

【 Step 3 】 誤差最小化 (最適問題、Fitting)

方法 1 : 二次式

$$f_1(x) = ax^2 + bx + c$$

$$\text{誤差} = (f_1(1) - 5)^2 + (f_1(2) - 10)^2 + (f_1(3) - 55)^2$$

↓ 誤差が最小となる a, b, c を求める

$$a = 20, b = -55, c = 40$$

$$f_1(x) = 20x^2 - 55x + 40$$
$$f_1(10) = 1490$$



方法 2 : 一次式 (線形)

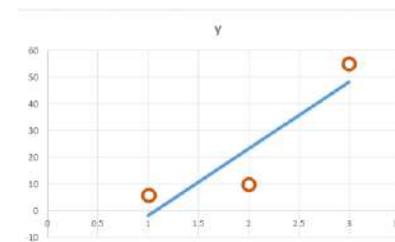
$$f_2(x) = ax + b$$

$$\text{誤差} = (f_2(1) - 5)^2 + (f_2(2) - 10)^2 + (f_2(3) - 55)^2$$

↓ 誤差が最小となる a, b を求める

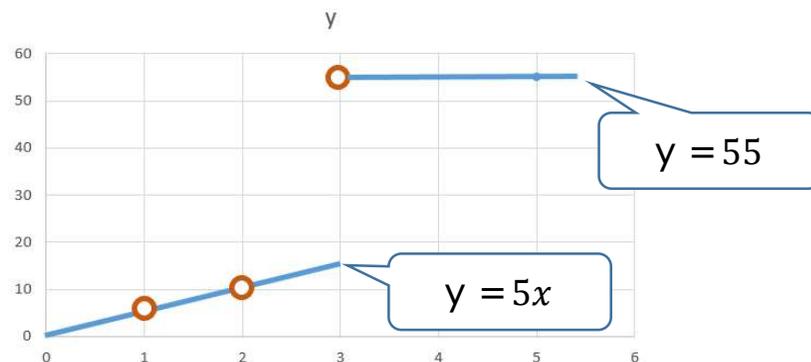
$$a = 25, b = -22.6$$

$$f_2(x) = 25x - 22.6$$
$$f_2(10) = 223.4$$



様々な線形

x	$y = f(x)$
1	5
2	10
3	55
10	?



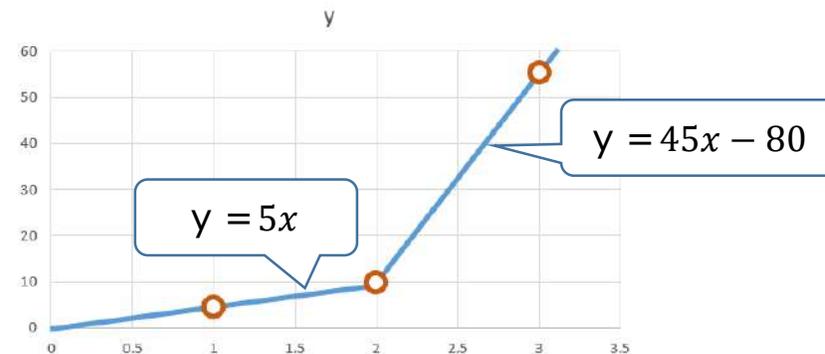
$$f_3(x) = g_1(x) + g_2(x)$$

where

$$g_1(x) = \begin{cases} 5x, & x \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

$$g_2(x) = \begin{cases} 55, & x > 3 \\ 0, & \text{otherwise} \end{cases}$$

$$f_3(10) = 55$$



$$f_4(x) = g_3(x) + g_4(x)$$

where

$$g_3(x) = \begin{cases} 5x, & x \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

$$g_4(x) = \begin{cases} 45x - 80, & x > 2 \\ 0, & \text{otherwise} \end{cases}$$

$$f_4(10) = 370$$

1. f への推定は、使われるデータだけでなく、仮定された形、誤差の定義にも依存する
2. 考えられる f の形、誤差定義は多く存在、事前では、最も適切なものは決められない
3. 現実世界での検証が必要

複雑な f : 深層学習、ニューラルネットワークの場合

- 入力データ(x)は入力層から、隠れ層経由、出力層まで、「伝達」される
- 各ノードの状態は、前の層のノードの状態の線形結合（ノード間の線上の重み W ）の結果に活性化関数（ h : 非線形関数）によって計算される
- f は、伝達関数の合成で計算される

$$x \rightarrow W_1 x \rightarrow h_1(W_1 x) \rightarrow W_2 h_1(W_1 x) \rightarrow h_2(W_2 h_1(W_1 x)) \dots$$

線形結合 非線形 線形結合 非線形

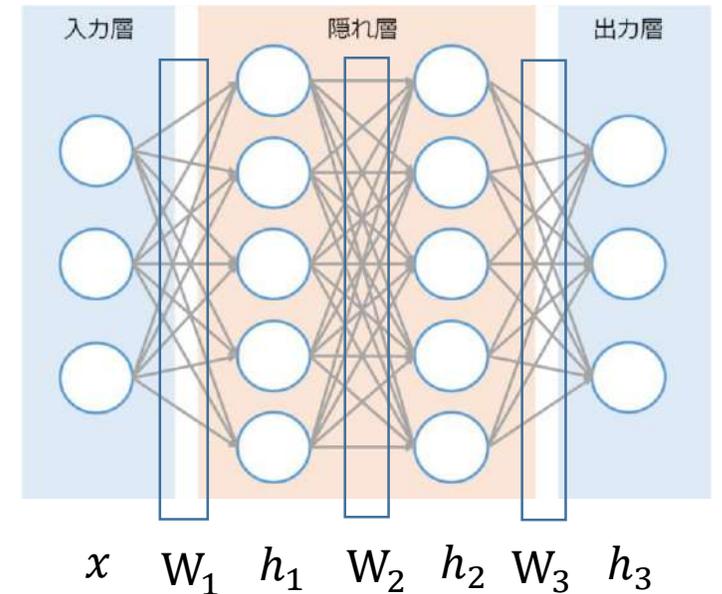
- 多種多様な活性化関数

- シグモイド関数 (Sigmoid)

$$h_i(x) = \frac{1}{1 + e^{-x}}$$

- ハイパブリックタンジェント (tanh: Hyperbolic Tangent)

$$h_i(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



現実世界での検証

- 結果既知のデータとの合致度合（正解値との比較）：予測精度への見積もり
- データの背後に隠された現実世界との適合検証
 - $f(a)$ への検証は、 a の近傍にある b の正解値 y_b と比較する
 - x が $f(x)$ への寄与度合（算出方法）、例えば、下記 f_1, f_2, f_3, f_4 のどちらが、経験上、現実世界の「作業要員」と「生産台数」の関係に近いかの検証
 - ニューラルネットワークのような複雑な f の計算式になれば、「検証」が難しい

x	$y = f(x)$
1	5
2	10
3	55
10	?

$f(x)$	
$f_1(x) = 20x^2 - 55x + 40$	$f_1(10) = 1490$
$f_2(x) = 25x - 22.6$	$f_2(10) = 223.4$
$f_3(x) = g_1(x) + g_2(x)$	$f_3(10) = 55$
$f_4(x) = g_3(x) + g_4(x)$	$f_4(10) = 370$

まとめ&用語

- 機械学習 (**教師あり学習**) とは、データ (x と y の組) から、 x と y 間の関数 (f) を構築する作業
- 関数(f)を **モデル** と呼ぶ
- 入力(x) を**説明変数**、出力結果(y)を**目的変数**、 y のとる値を**ラベル**とも呼ぶ
- データからモデルを構築する作業を**学習**と呼ぶ
- モデルに、入力値を代入して、対応する y の値を算出することを**予測**と呼ぶ
- モデルによる予測の正当性：**連続性**
 - a と b の距離が充分近ければ (近傍)、 $f(a)$ と $f(b)$ の差も小さい
- 理屈上、同じデータから、構築可能なモデルの数は無数にある
- データそのもの、及び、そこから得られたモデルの両方を **現実世界での検証** が必要不可欠
- 複雑なモデルほど、検証が難しくなる

x	$y = f(x)$
1	5
2	10
3	55
10	?

実例：健康診断データ → 血圧を予測するモデル

SL Viewer - medical_data

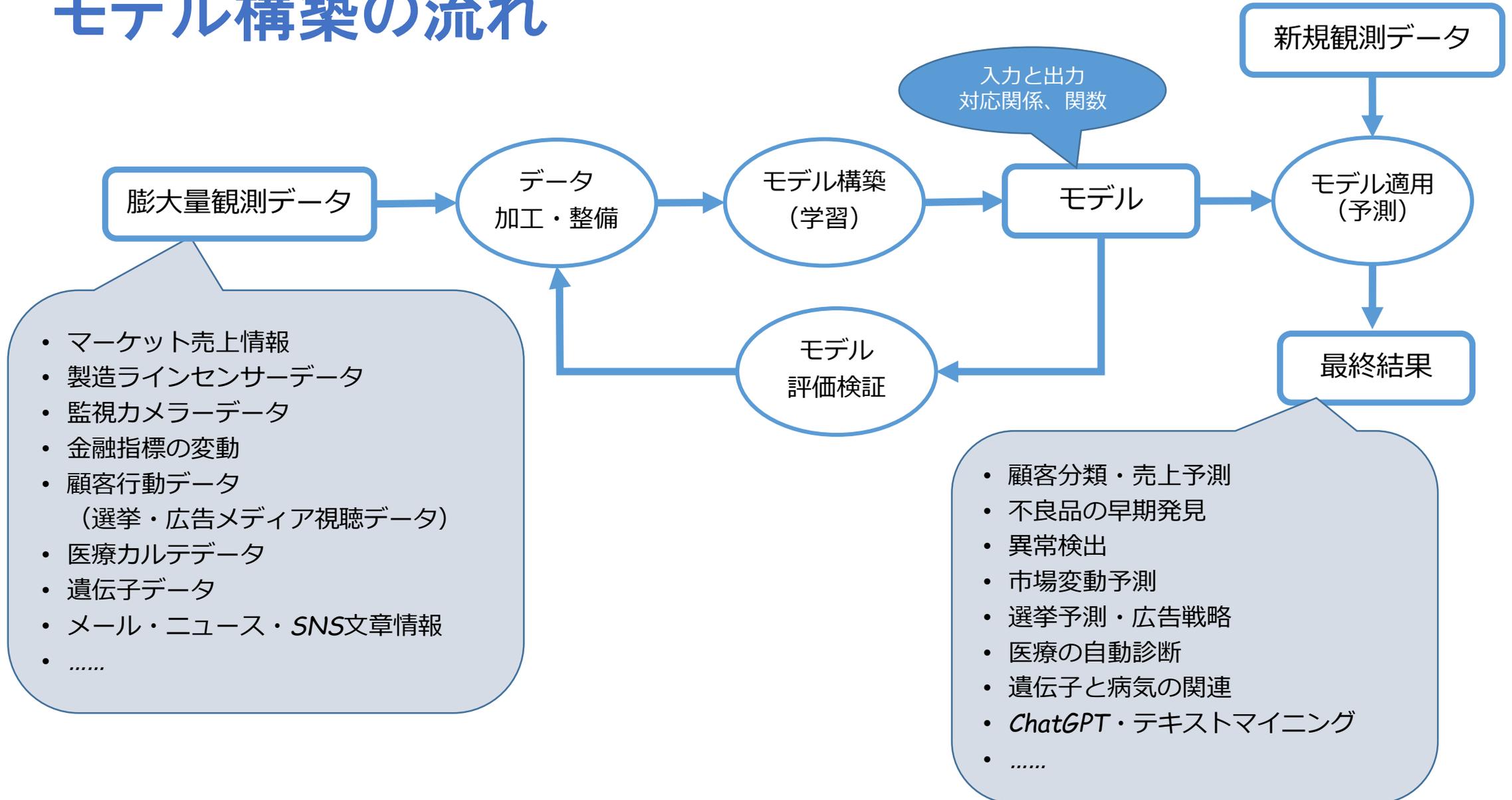
	性別	都道府県	地域	血圧薬投与	年齢	身長(cm)	体重(kg)	BMI_算出	食塩摂取量	拡張期(最低)血圧	収縮期(最高)血圧	エネルギー
1	女	茨城	関東2	yes	53	151	56	24.560326	9.901382	63.000000	116.000000	2160.3935
2	男	三重	東海	yes	68	164	81	30.116003	15.248550	91.000000	130.000000	2424.6037
3	女	鳥取	中国	no	30	162	52	19.814053	8.129100	66.000000	112.000000	375.5177
4	女	NA	NA	no	53	161	65	25.076193	7.451425	86.000000	111.000000	2506.0817
5	女	高知	四国	yes	63	149	59	26.575380	7.708094	82.000000	136.000000	487.3924
6	女	長崎	北九州	yes	35	161	49	18.581515	7.581515	63.000000	93.000000	714.5440
7	女	静岡									137.000000	1515.0446
8	女	群馬									140.000000	1414.2807
9	女	徳島									126.000000	1684.5352
10	男	埼玉	関東1	yes	38	157	68	27.387320	13.347844	80.000000	144.000000	2749.9951
11	女	富山	北陸	yes	30	164	47	17.474718	9.414392	61.000000	106.000000	974.0056
12	男	青森	東北	no	38	180	83	25.617285	20.869238	63.000000	141.000000	2246.6889
13	女	岐阜	東海	yes	47	154	54	22.769438	7.776840	63.000000	130.000000	
14	女	徳島	四国	yes	43	157	62	25.153151	8.475278	77.000000	130.000000	2276.7248
15	女	千葉	関東1	no	46	158	39	15.622497	7.493034	69.000000	117.000000	592.9820
16	女	栃木	関東2	NA	53	155	45	18.730490	10.330496	83.000000	126.000000	2609.8439
17	女	静岡	東海	no	63	157	40	16.227839	7.394319	69.000000	112.000000	739.2099

血圧 = f (性別, 都道府県, 地域, 薬投与, 年齢, 身長, 体重, BMI, 食塩, ...)

予測を行うために必要な情報：説明変数

予測対象：目的変数

モデル構築の流れ



機械学習の課題

- 予測精度の高いモデル構築には、熟練の機械学習技術者が必要
 - 複雑な計算環境構築
 - プログラミングコーディング技術
 - 手動での試行錯誤（職人技）→ モデルの属人化問題
- モデルの不可読性
 - モデル（ f ）は複雑すぎ、人間に理解不能（ブラックボックス状態）
 - 予測根拠・理由を示されない

妨げ

- 機械学習技術の普及
- 現実世界での検証→人間の意思決定の道具
- ドメイン専門家の知識・常識のモデルへの反映

モデルの予測精度 vs 説明能力

■ 機械学習に抱えるジレンマ

- 予測精度向上のために、モデルが複雑なものにする
- モデルは複雑になるほど、説明能力が失われる

■ 説明可能なAI (XAI = eXplainable AI or IML = Interpretable Machine Learning) に関する研究

- データから、高精度で複雑なモデル (M) を構築する
- 限られたデータ範囲の中で、M を説明可能なシンプルなモデル (S) で**近似する**
- Mの代わりに、Sで説明を行う (**局所的(local)** 的な説明)

データ範囲の決め方？
モデルの近似精度？

■ **大域 (global)** 的な説明

- データから、説明可能なモデル (S) を構築する、S そのもので説明を行う

予測精度？

■ 説明可能とは

予測対象の目的変数に対して、
説明変数の寄与条件と寄与度合
が明確である

二つ説明可能なモデル

■ 線形モデル（回帰、ロジステック回帰）

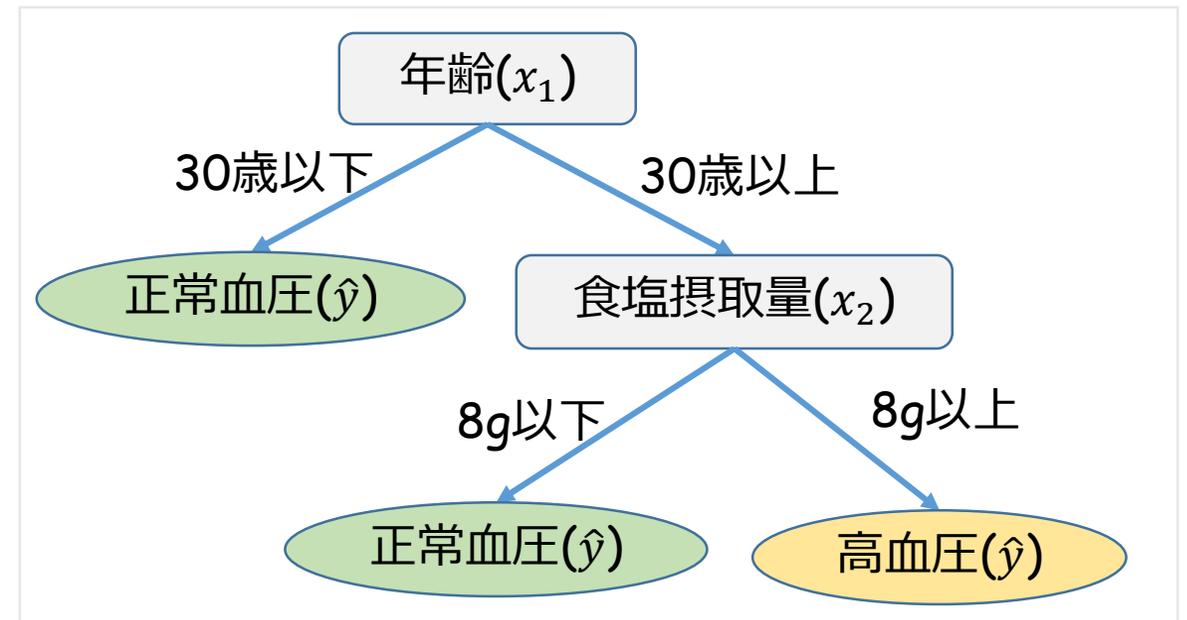
$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

【理由】 予測値 \hat{y} の計算式から説明変数 (x_1, x_2, \dots, x_n) 値の寄与度（係数 β ）は明確である

■ 決定木（Decision Tree）

【理由】 予測値 \hat{y} （血圧）の算出に、辿った説明変数値に関する前提条件が明示される。

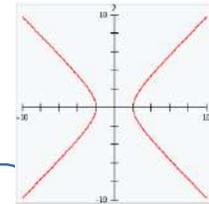
右図の例：「高血圧」と判断された理由は
[年齢 > 30 ∧ 食塩摂取量 > 8g]



複雑な問題に予測精度が劣る

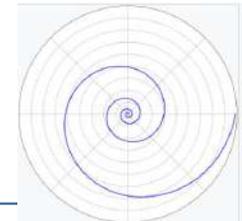
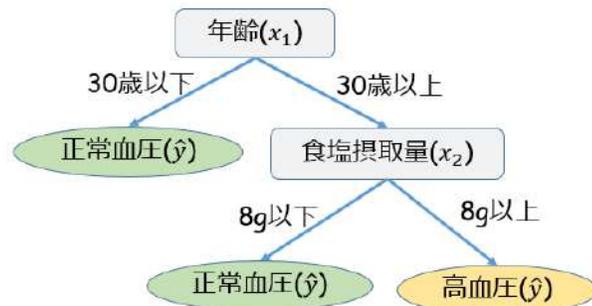
■ 線形モデル（回帰、ロジステック回帰など）

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$



目的変数と説明変数間、線形性に従わない問題に弱い

■ 決定木（Decision Tree）



1. 目的変数が連続値の場合は、予測誤差が大きい
2. 説明変数空間を「垂直」的に分割される以外の問題に弱い

2. 説明可能な機械学習、「AI電卓」の技術概要

AI電卓とは

大域説明可能、かつ、高水準な予測精度モデルを
電卓利用と同等な手軽さで実現

■ 予測精度の高いモデル構築には、熟練AI技術者が必要

- 複雑な計算環境構築
- プログラミングコーディング技術
- 手動での試行錯誤（職人技） → モデルの属人化問題

■ モデルの不可読性

- モデルはブラックボックス = 人間に理解不能
- 予測根拠・理由を示されない

煩雑な設定は不要
GUIによるマウスクリック

「人の労力・時間」の代わりに、
「計算機の計算力・計算時間」
を活用する

モデルは、人間に馴染み
がある論理式（ルール）で構成
大域説明可能だけでなく、
予測根拠も同時に示す

AI電卓モデル: ルールの線形結合

$$f(x_1, x_2, \dots, x_n) = \beta_0 + \sum_{i=1}^m \beta_i r_i(x_1, x_2, \dots, x_n)$$

where

$$r_i(x_1, x_2, \dots, x_n) = \begin{cases} 1 & \text{if } c(x_1, x_2, \dots, x_n) = \text{true} \\ 0 & \text{otherwise} \end{cases}$$

$c(x_1, x_2, \dots, x_n)$ は条件式

モデル f がルール r_i の線形結合と定義される

↓ r_i のもう一つ形

$$r_i(x_1, x_2, \dots, x_n) = \begin{cases} x_j & \text{if } c(x_1, x_2, \dots, x_n) = \text{true} \\ 0 & \text{otherwise} \end{cases}$$

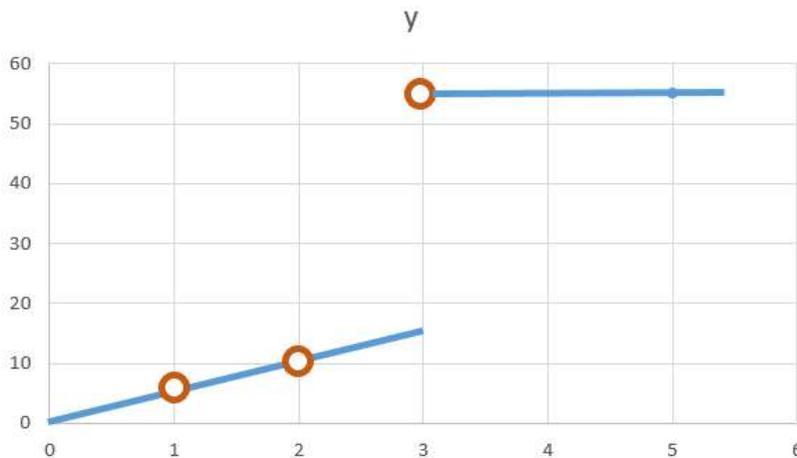
where

$$j \in \{1, 2, \dots, n\}$$

条件付き線形モデル対応

ルールの線形結合モデル

x	$y = f(x)$
1	5
2	10
3	55



$$f_3(x) = 5r_1(x) + 55r_2(x)$$

where

$$r_1(x) = \begin{cases} x, & x \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

$$r_2(x) = \begin{cases} 1, & x > 3 \\ 0, & \text{otherwise} \end{cases}$$

モデルの説明能力

1. $f(x)$ に対し、ルールの寄与度が決められる
2. ルールの「成立条件」は明確である
3. 同じルールにヒットするデータは、何らかの意味で共通する性質を持つと見なすことが可能

ルールの役割 I : 知識表現

【健康診断データのモデル例】

$$f(x) = \beta_0 + \beta_1 r_1(x) + \beta_2 r_2(x) + \dots + \beta_m r_m(x)$$

.....
 r_1 は、血圧を上げる要因

$$\text{血圧} = 78.36 + 3.50 * r_1 - 3.89 * r_2 - 2.07 * r_3 + \dots$$

r_1 : 食塩摂取量 > 2.21 \wedge 年齢 > 38.96

r_2 : 性別 = 男性 \wedge 年齢 \leq 38.25

r_3 : 肥満度BMI \leq 20.52 \wedge 年齢 \leq 47.9

r_2 は、血圧を下げる要因

ルールの役割 II データ間の近さ、距離

【共起テーブル】

データ	r1	r2	r3	...
s1	0	1	0	...
s2	1	0	0	...
s3	1	0	1	...
s4	0	0	1	...
s5	0	1	0	...
s6	1	1	0	...
...

同一ルールにヒットしたデータ同士が何らかの意味で「近い」と見ることが可能

データがルールの条件を満たす（ヒット）場合は、1を、満たさない場合は、0と表現

二つのデータは、共通にヒットするルールが多ければ、多いほど、距離が近いと見ることが可能

1. データ間の（共起）距離による、構造化
2. Unsupervised Learning / Clustering の基礎となる
3. 「距離」は共起を表すだけでなく、その背後に、ルール生成時に使われる目的変数まで辿ることが可能

ルールは知識になる、知識をルールに変えられる

SL Viewer - final_rule_table_transcript_1

RuleSeqNo	CondVName	Compare	Threshold	Pattern	Representation	importance	coeficient	intercept	ti
1									

Rule Entry (拡張期(最低) 血圧~食塩摂取量: medical_data1K)

Rule Weight: 1.0
Rp: 97.324 plot

Adjust to <mean>
Save the figure
Plot Options

Rule Entry (sales~family: aaaaa)

Rule Weight: 1.0
Rp: 2.072 plot

Adjust to <mean>
Save the figure
Plot Options

食塩摂取量 x 年齢

dragable SP1: 8.425 plot
dragable SP2: 22.082 plot

3σ 2σ 1σ
equal intervals
equal frequencies
cut uncut
Save the figure
Plot Options

family x

selection	reverse selection	frequency	nselection=11; freq_sel=37,835 (33.34%)
search pattern		include	
family			
1 BOOKS	✓	-0.060	0.390 34
2 HOME APPLIANCES	✓	-0.009	1.027 34
3 LINGERIE		0.036	1.007 34
4 BABY CARE		0.042	0.735 33
5 PREPARED FOODS		0.073	1.004 34
6 SEAFOOD		0.098	1.029 34
7 FROZEN FOODS	✓	0.210	0.772 34
8 HARDWARE	✓	0.214	1.068 34

View Cond Table Cancel Add to knowledge rule_base

Rule Entry (年齢~性別: 性別別年齢-平均 世界平均)

Rule Weight: 4.676 (2427)
adjustment to: none
sort in desc and
adjustment: none
y_axis: none
y_scale: 0.01
Save the figure
Plot Options

性別別年齢-平均 世界平均

dragable SP1: 65.077 plot
dragable SP2: 22.111 plot

3σ 2σ 1σ
equal intervals
equal frequencies
cut uncut
Save the figure
Plot Options

性別別年齢-平均 世界平均

Rule Weight: 0.152 (200483)
adjustment to: none
sort in desc and
adjustment: none
y_axis: none
y_scale: 0.01
Save the figure
Plot Options

性別別年齢-平均 世界平均

selection	reverse selection	frequency	aggregate direction
search pattern		include	
性別年齢			
1 1,500~2,000万円未満	✓	0.000 0.000 0 0.000	
2 1,200~1,500万円未満	✓	100.000 0.000 4 9.362	
3 1,000~1,200万円未満	✓	100.000 0.000 4 9.362	
4 不明		100.000 0.000 2 4.053	
5 500~600万円未満		90.000 10.000 10 23.256	
6 100~500万円未満		100.000 0.000 1 2.528	
7 200~300万円未満		100.000 0.000 2 4.053	
8 100~200万円未満		100.000 0.000 8 8.927	
9 0円~100万円未満		66.467 34.333 3 6.927	

View Cond Table Cancel Add to knowledge rule_base

AI電卓のルール編集画面、
ルールの特徴、分布を手軽に確認可能

AI電卓の機能一覧

外部データ
csv形式テキスト

導入

画面上の
右クリックメニュー

基本操作

- データのコピー
- データの外部保存
- 基礎集計

モデル

- 教師あり学習
- 自己教師あり学習
- 教師無し学習/クラスタリング

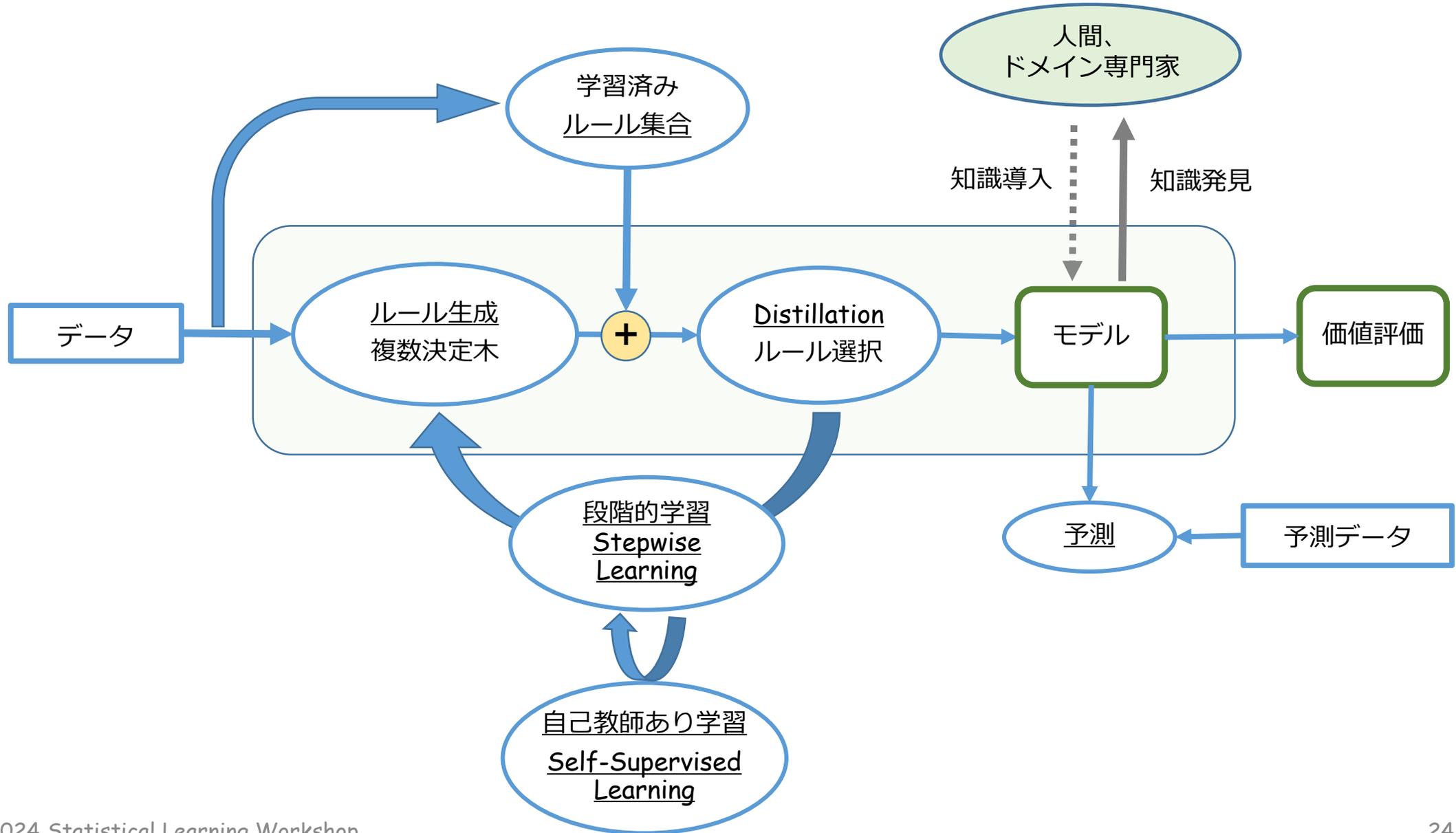
応用アプリ

- データ加工
- 知識ルール編集
- 時系列処理
- 異常検出
- 欠損補填
- 因果推論

SL Viewer - medical_data

	性別	都道府県	地域	任薬投与	年齢	身長(cm)	体重(kg)	BMI_算出	食塩摂取量
1	女	茨城	関東2	yes	53	151	56	24.5603	
2	男	三重	東海		68	164	81	30.1160	
3	女	鳥取			30	162	52	19.8140	
4	女	NA			50	61	65	25.0761	
5	女	高知			63	49	59	26.5753	
6	女	長崎			25	61	49	18.9035	
7	女	静岡						8.59103	
8	女	群馬						5.429117	7.261
9	女	徳島						1.926126	9.155
10	男	埼玉						7.5873	
11	女	富山						7.474	
12	男	青森						5.617	
13	女	岐阜			47	154	54	22.769	
14	女	徳島			43	157	62	25.153	
15	女	千葉			46	15	39	15.622	
16	女	栃木			53	15	45	16.227	
17	女	静岡			63	157	40	16.227	

AI電卓の教師あり学習の流れ



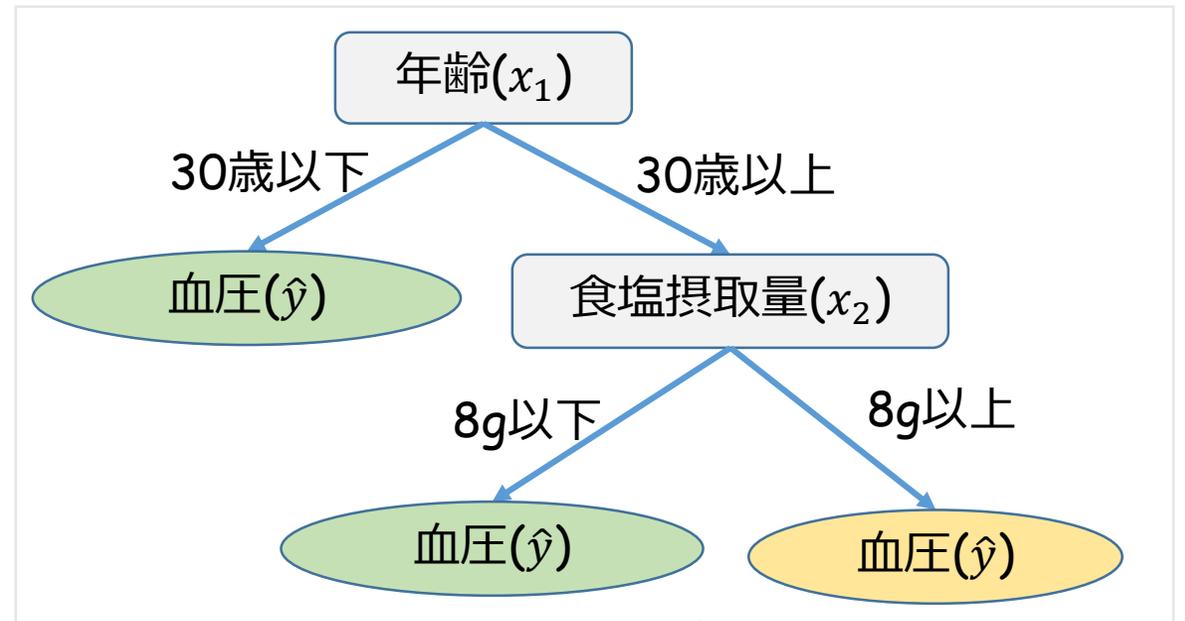
ルール生成(複数決定木)

■ 決定木 (Decision Tree)

- 自然界の木を逆さに表示され、ルートノードは最も上に表示され、一番下に、葉ノードが表示される
- ルートノードから、葉ノードまで辿る経路には、条件式がおかれ、例えば、
[年齢 > 30 ∧ 食塩摂取量 > 8g]

■ 決定木の自動生成

- ルートノードから、機械的に、もっとも血圧の高・低を分けられるデータの分割方法を見つけっていく
- 例えば、右図に示したように、データを 年齢 ≤ 30 と 年齢 > 30 分けると、血圧の高・低に大きく分ける
- 更に、食塩摂取量 $\leq 8g$ と 食塩摂取量 $> 8g$ でデータを血圧の高・低に大きく分ける



- ルールの生成は、多数のこのような決定木から得られる
- ルールは、ルートノードから葉ノードまでのものだけでなく、中間ノードまでのものも含まれる

ルール選択(Distillation)

$$\text{血圧} \approx \beta_0 + \beta_1 r_1 + \beta_2 r_2 + \dots + \beta_m r_m$$

- フィッティング (Fitting)

上記、右側の計算は、血圧の「正解値」との間に最も誤差が小さくなるような

$$\beta_0, \beta_1, \beta_2, \dots, \beta_m$$

を求める

- ルール選択

フィッティングを行うときに、出来るだけ $\beta_i = 0$ のものを多く算出、

$\beta_i = 0$ となるものは、対応するルール r_i をモデルから除くことができるから

ルールの価値、モデルの価値

■ 単一ルールの価値

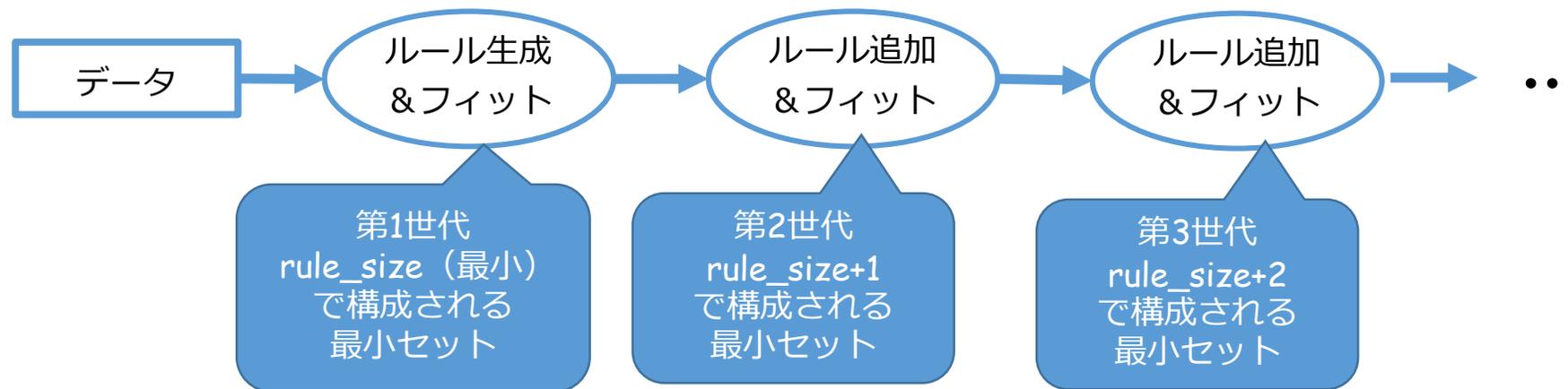
- **ヒット数** 条件を満たすデータの数 **【多いほど良い】**
- **大きさ (rule size)** 'and' で結ばれる条件の個数 **【小さいほど良い】**
 - rule_size=0: [年齢] ... 無条件ルール、任意のデータにヒットする、ルールの代表値は「年齢」
 - rule_size=1: [年齢>30]、[食塩摂取量>8g]、...
 - rule_size=2: [年齢>30 ∧ 食塩摂取量>8g]、...
- **予測対象の分散** 目的変数が存在した場合、ルールにヒットするデータの目的値の分散 **【小さいほど良い】**
 - ルールにヒットする「血圧」の分散

■ モデル（ルールの集合として）の価値

- 予測値 vs 正解値 間の誤差 **【小さいほど良い】**
- **ルールの数** **【少ないほど良い】**

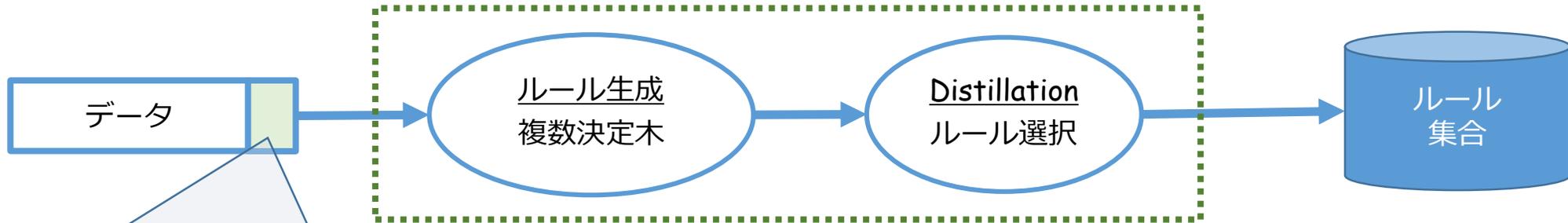
段階的学習 (Stepwise Learning)

正解値との誤差だけでなく、
モデルの**価値**も選択基準に、逐次的に探索を行う



自己教師あり学習 (SSL: Self-Supervised Learning)

教師あり学習アルゴリズム



教師値付与方法 (Labeling Function)

- **Reproduction** 既存ルールによるラベリング
- **Target** 目的変数値の変形
- **Improvement** ラベルの改善
- **PIV = Privileged Information Variable**

性別	都道府県	地域	hypertension 参加	年齢	身長(cm)	体重(kg)	BMI 算出	食塩摂取量	拡張期(最低)血圧	収縮期(最高)血圧	工
女	埼玉	関東1	yes	-inf	147	38	17.585266	5.356928	69	89	1
女	佐賀	北九州	no	48	-inf	52	-inf	3.647383	84	133	1
女	三重	東海	no	61	155	43	17.898024	9.546216	80	101	
女	高知	四国	no	51	161	52	20.060955	12.771319	82	129	
男	福岡	北九州	no	47	164	69	25.654373	12.510663	85	129	1
女	岡山	中国	no	62	147	56	25.915129	12.486358	78	139	1
女	東京	関東1	yes	45	161	57	21.989893	11.770765	73	111	1
女	沖縄	南九州	no	54	149	68	30.629251	7.829592	76	127	1
女	山口	中国	yes	58	161	58	22.375679	10.248017	67	101	1
女	島根	中国	NA	58	156	50	20.545694	13.719254	103	140	
男	福井	北陸	no	70	70	50	17.301039	4.818835	73	113	
女	宮城	東北							68	138	1
女	佐賀	北九州							70	125	
女	長崎	北九州							6	139	
女	岡山	中国							1	132	
男	滋賀	近畿2							2	123	
男	徳島	四国							0	113	3
男	熊本	北九州							1	131	
女	佐賀	北九州							2	123	
男	群馬	関東2							7	122	

Privileged Information

学習時利用可能な説明変数のこと、例えば、性別、都道府県、地域、年齢などは全てPIVと見なすことが可能

PIV と LLM (Large Language Model、言語モデル)

■ 文章の表形式表現

ChatGPTは、高度な言語理解力により、文章を理解し、適切な修正の提案が可能です。

文章

単語 1	単語 2	単語 3	単語 4	単語 5	単語 6	単語 7
ChatGPT	高度な	言語理解力	により	文章	を	理解
高度な	言語理解力	により	文章	を	理解	適切な
言語理解力	により	文章	を	理解	適切な	修正
により	文章	を	理解	適切な	修正	の
文章	を	理解	適切な	修正	の	提案
を	理解	適切な	修正	の	提案	が
理解	適切な	修正	の	提案	が	可能
適切な	修正	の	提案	が	可能	です

表形式

PIを用いてのSelf-Supervised は、LLMの前・後に現れる単語から、単語を「予測」するモデルと同じ原理

どの列でも予測対象（目的変数）となりうる、その場合は、他の列を説明変数として利用する

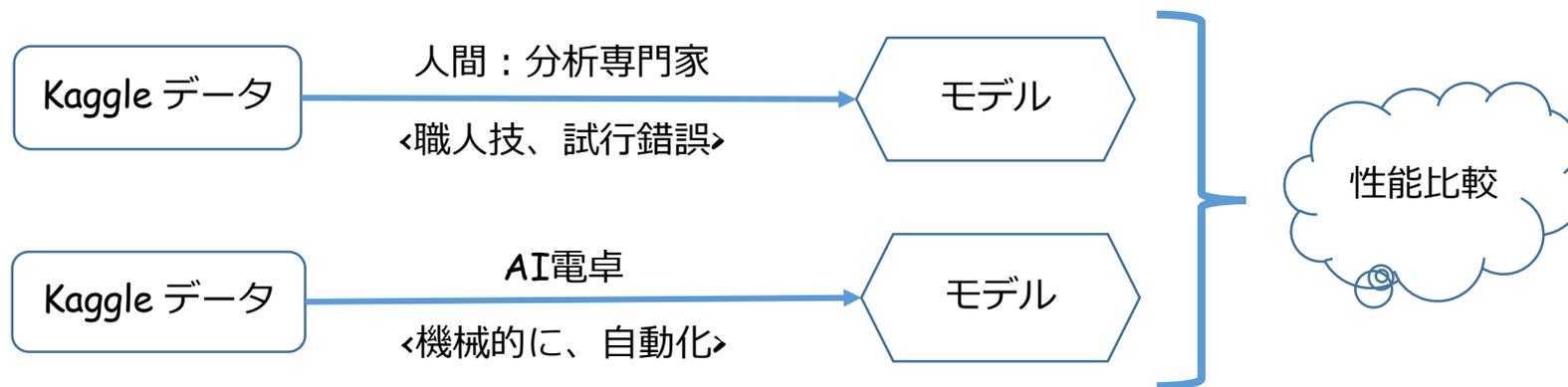
3. 検証事例

Kaggleデータ分析コンテスト

■ Kaggleとは

- Google社が運営、企業や、研究者がデータを投稿し、世界中の統計・機械学習・サイエンティストがデータに最適な予測モデルを競い合う
- 参加者
 - 世界最大規模(?)、約10万人参加
 - 情報科学者、統計学者、経済学者、数学者など
- 様々な業種から様々なテーマ
 - 金融、流通、生物・医療・バイオ、製造、メディア・広告、自然言語処理、有償テーマも多数
- 日本でも、Kaggleの攻略法、体験談などに関する書籍多数

■ 目的



事例1： Kaggle： House Price

- 不動産に関する属性から、不動産価格を予測するモデル

- 学習用データ（正解を含むデータ）

 - 属性数（列数）：81

 - 行数：1460件

- 検証用データ（正解未知）

 - 属性数（列数）：80

 - 行数：1459件

- モデル評価方法

学習データから構築したモデルを用いて、検証データに対して不動産価格を予測し、予測結果の精度

データ特徴

SL Viewer - train.csv

	bsnPorch	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
1	0	0	0	NA	NA	NA	0	2	2008	WD	Normal	208500
2	0	0	0	NA	NA	NA	0	5	2007	WD	Normal	181500
3	0	0	0	NA	NA	NA	0	9	2008	WD	Normal	223500
4	0	0	0	NA	NA	NA	0					140000
5	0	0	0	NA	NA	NA	0	1				250000
6	320	0	0	NA	MnPrv	Shed	700	1				143000
7	0	0	0	NA	NA	NA	0	8	2007	WD	Normal	307000
8	0	0	0	NA	NA	Shed	350	11	2009	WD	Normal	200000
9	0	0	0	NA	NA	NA	0	4	2008	WD	Abnorml	129900
10	0	0	0	NA	NA	NA	0	1	2008	WD	Normal	118000
11	0	0	0	NA	NA	NA	0	2	2008	WD	Normal	129500

nrow=1,460, ncol=81

予測対象：販売価格

- 学習用、検証用両方のデータに、欠損（NA）が多数存在
- 予測対象（不動産価格）の分布にかなりの格差がある
- 不動産価格に関連性が薄い属性が多く見受けられる

AI電卓でのモデル構築様子

■ 下記のような前処理を一切行わず

- 欠損の補填
- 予測対象（不動産価格）の分布への事前調整
- 特徴量抽出

■ 学習用 & 検証用データをそのまま導入、学習 & 予測

Unsupervised Learning Model (train)

Option
n_thread
rand_seed 1
 remove NA from target
-knn
search_steps 30
search_range 20
alpha 0.9
 early_stop_by_eval_block
gb
n_tree 0
max_height 100
max_leaf 64
n_cand_split 3
random_split 0.1
erf
n_tree 10
max_height 100
max_leaf 64
n_cand_split 3
random_split 0.1

Variable	<selv>	<selv>	ncate	biggest block	NA	mean
1 Id			0	0.010274	0.000000	730.500000
2 MSSubClass	✓	✓	15	0.367123	0.000000	-inf
3 MSZoning	✓	✓	5	0.788356	0.000000	-inf
4 LotFrontage	✓	✓	0	0.193151	0.177397	70.049957
5 LotArea	✓	✓	0	0.295205	0.000000	10516.828125
6 Street	✓	✓	2	0.995890	0.000000	-inf
7 Alley	✓	✓	3	0.937671	0.937671	-inf
8 LotShape	✓	✓	4	0.633562	0.000000	-inf
9 LandContour	✓	✓	4	0.897945	0.000000	-inf
10 Utilities	✓	✓	2	0.999315	0.000000	-inf
11 LotConfig	✓	✓	5	0.720548	0.000000	-inf
12 LandSlope	✓	✓	3	0.946575	0.000000	-inf
13 Neighborhood	✓	✓	25	0.154110	0.000000	-inf
14 Condition1	✓	✓	9	0.863014	0.000000	-inf
15 Condition2	✓	✓	8	0.989726	0.000000	-inf
16 BldgType	✓	✓	5	0.835616	0.000000	-inf
17 HouseStyle	✓	✓	8	0.497260	0.000000	-inf
18 OverallQual	✓	✓	0	0.271918	0.000000	6.099315
19 OverallCond	✓	✓	0	0.562329	0.000000	5.575343
20 YearBuilt	✓	✓	0	0.079452	0.000000	1971.267822
21 YearRemodAdd	✓	✓	0	0.121918	0.000000	1984.865723
22 RoofStyle	✓	✓	6	0.781507	0.000000	-inf

Exploratory Analysis - SalePrice (train)

Selection
nselection=79(nan%)

search pattern

Variable	ncate	biggest block	NA	mean
1 OverallQual	0	0.271918	0.000000	6.099315
2 Neighborhood	25	0.154110	0.000000	-inf
3 GrLivArea	0	0.193151	0.177397	70.049957
4 ExterQual	0	0.295205	0.000000	10516.828125
5 BsmtQual	2	0.995890	0.000000	-inf
6 KitchenQual	3	0.937671	0.937671	-inf
7 GarageCar	4	0.633562	0.000000	-inf
8 Garage1stFlr	4	0.897945	0.000000	-inf
9 TotalBsmtSF	2	0.999315	0.000000	-inf
10 1stFlrSF	5	0.720548	0.000000	-inf
11 FullBath	3	0.946575	0.000000	-inf
12 Garage1	9	0.863014	0.000000	-inf
13 Fireplaces	8	0.989726	0.000000	-inf
14 TotRms	5	0.835616	0.000000	-inf
15 YearBuilt	0	0.079452	0.000000	1971.267822
16 YearRemodAdd	0	0.121918	0.000000	1984.865723
17 Foundation	8	0.497260	0.000000	-inf
18 GarageType	0	0.271918	0.000000	6.099315
19 MSSubClass	0	0.562329	0.000000	5.575343
20 Fireplaces	0	0.079452	0.000000	1971.267822
21 BsmtFinType1	7	0.459140	0.000000	0.954486

```
train for SalePrice - train
fit_regression: rpw_preparing ...
fit_regression: rpw_preparing finished
=== fit_regression options (23-5-25 7:27:59) ===
nloop=30 nrule=236758 early_stop(use_eval=0, threshold=8000,000000)
===
fit_regression: itr=1/30 lambda=547843.250 active=1 mae=57434.691
fit_regression: itr=2/30 lambda=373241.250 active=20788 mae=57235.841
fit_regression: itr=3/30 lambda=254286.297 active=54643 mae=56471.293
fit_regression: itr=4/30 lambda=173240.234 active=86412 mae=54982.016
fit_regression: itr=5/30 lambda=118029.242 active=114167 mae=52804.856
fit_regression: itr=6/30 lambda=80412.383 active=139192 mae=49425.133
fit_regression: itr=7/30 lambda=54784.318 active=158855 mae=45384.051
fit_regression: itr=8/30 lambda=37324.121 active=173352 mae=40658.195
fit_regression: itr=9/30 lambda=25428.627 active=185388 mae=39550.176
fit_regression: itr=10/30 lambda=17324.322 active=194093 mae=39042.953
fit_regression: itr=11/30 lambda=11802.923 active=201366 mae=26099.230
```

Variable	ncate	biggest block	NA	mean
1 OverallQual	0	0.271918	0.000000	6.099315
2 Neighborhood	25	0.154110	0.000000	-inf
3 GrLivArea	0	0.193151	0.177397	70.049957
4 ExterQual	0	0.295205	0.000000	10516.828125
5 BsmtQual	2	0.995890	0.000000	-inf
6 KitchenQual	3	0.937671	0.937671	-inf
7 GarageCar	4	0.633562	0.000000	-inf
8 Garage1stFlr	4	0.897945	0.000000	-inf
9 TotalBsmtSF	2	0.999315	0.000000	-inf
10 1stFlrSF	5	0.720548	0.000000	-inf
11 FullBath	3	0.946575	0.000000	-inf
12 Garage1	9	0.863014	0.000000	-inf
13 Fireplaces	8	0.989726	0.000000	-inf
14 TotRms	5	0.835616	0.000000	-inf
15 YearBuilt	0	0.079452	0.000000	1971.267822
16 YearRemodAdd	0	0.121918	0.000000	1984.865723
17 Foundation	8	0.497260	0.000000	-inf
18 GarageType	0	0.271918	0.000000	6.099315
19 MSSubClass	0	0.562329	0.000000	5.575343
20 Fireplaces	0	0.079452	0.000000	1971.267822
21 BsmtFinType1	7	0.459140	0.000000	0.954486

結果
モデル

Kaggleからの成績評価(2023-5-25)

House Prices - Advanced Regression Techniques
Predict sales prices and practice feature engineering, RFs, and gradient boosting

Kaggle · 4,728 teams · Ongoing

Overview Data Code Discussion **Leaderboard** Rules Team Submissions **Submit Predictions** ...

Leaderboard Raw Data Refresh

YOUR RECENT SUBMISSION

✓ **submission2.csv** Score: 0.11728
Submitted by Xu Liangwei · Submitted 15 minutes ago

90	M.R.0024		0.11261	2	2mo
91	ningnujel		0.11292	3	11d
92	Xu Liangwei		0.11385	271	16m

😊 Your Best Entry!
Your submission scored 0.11728, which is not an improvement of your previous score. Keep trying!

参加チーム数 : 4728
順位 : 92
上位 : 1.94%

事例2: Kaggle: Store Sales = 時系列データ分析

- 食料品チェーン店 (grocery store) の日々の売上実績から、将来の売上を予測する
- 店舗数 (store_nbr) = 54、商品分類 (family) 数=33
- 店舗毎、商品分類毎の売上に関する時系列数 = $54 * 33 = 1782$
- 学習用データ (正解を含むデータ)
 - 売上履歴 (日数) : 1680日
 - 行数 : 294万件、属性数 (列数) : 17
- 検証用データ (正解未知)
 - 予測対象日数 : 16日
 - 行数 : 28512件 (= 店舗数 * 商品分類数 * 予測対象日数)
- モデル評価方法
学習データから構築したモデルを用いて、検証データに対して売上を予測し、予測精度

データ特徴

date	store_nbr	Store_type	Store_cluster	Store_city	Store_state	family	sales	onpromotion	dcoilwtico	Holiday	SalaryDay	transact
2014-07-30	17	C	12	Quito	Pichincha	MAGAZINES	0.000000	0	104.290001		no	1098
2014-07-30	17	C	12	Quito	Pichincha	MEATS	226.317001	0	104.290001			
2014-07-30	17	C	12	Quito	Pichincha	PERSONAL CARE	195.000000	1	104.290001			
2014-07-30	17	C	12	Quito	Pichincha	PET SUPPLIES	0.000000	0	104.290001			
2014-07-30	17	C	12	Quito	Pichincha	PLAYERS AND ELECTRO	5.000000	0	104.290001		no	1098
2014-07-30	17	C	12	Quito	Pichincha	POULTRY		0	104.290001		no	1098
2014-07-30	17	C	12	Quito	Pichincha	PREPARED FOODS		0	104.290001		no	1098
2014-07-30	17	C	12	Quito	Pichincha	PRODUCE		0	104.290001		no	1098

商品分類

予測対象：売上

プロモーションFG

店舗情報

日ごとの石油価格

振替休日・給料日

- 時系列データ（店舗、商品ごと、日々売上）以外の「外部要因」も含まれる
- 予測対象（売上金額）の分布にかなりの格差（桁違い）がある
- データの行数は比較的が多い（300万件前後）

Kaggleからの成績評価(2023-9-26)

Store Sales - Time Series Forecasting
Use machine learning to predict grocery sales

Kaggle · 695 teams · Ongoing

Overview Data Code Discussion **Leaderboard** Rules Team Submissions **Submit Predictions** ...

39	Will Gilchrist		0.39261	2	12d
40	Mario Refoyo López		0.39495	4	1mo
41	Xu Liangwei		0.39778	110	5m
 Your Best Entry! Your most recent submission scored 0.39778, which is an improvement of your previous score of 0.40014. Great job!					
42	David Gilbertson		0.39842	20	1mo

参加チーム数 : 695
順位 : 41
上位 : 5.89%

Tweet this

5. まとめ

- AI技術者をモデル構築プロセスから排除
- 説明可能な、ハイパフォーマンスモデル
- ルールとデータの共起から、データに構造を持たせ、教師無し学習の基礎になる
- 予測だけでなく、異常検出、因果推論、クラスタリング、欠損補填にも有効

ルールは知識になる、知識をルールに変えられる！

6. Future Work

- より高品質、よりハイパフォーマンスモデル
- 学習プロセスの高速化
 - **ハード面** 超並列計算の GPU化 (条件分岐 (if_else) 不得意?)
 - **ソフト面** 高品質ルール生成の高速アルゴリズム開発、研究
 - **目標** 大規模データ (数百万件) からトップ予測精度モデルまでの学習時間 : 1週間 ⇒ 1日
- 共起テーブルの活用
- 後工程 : 自然言語によるモデルの予測への説明文章
 - モデルの可読性の有効利用
- 他の適用分野への開拓、検証 (Kaggle テーマ)
 - 医療診断、生物・遺伝子解析
- 自然言語処理
 - 説明可能な言語モデル