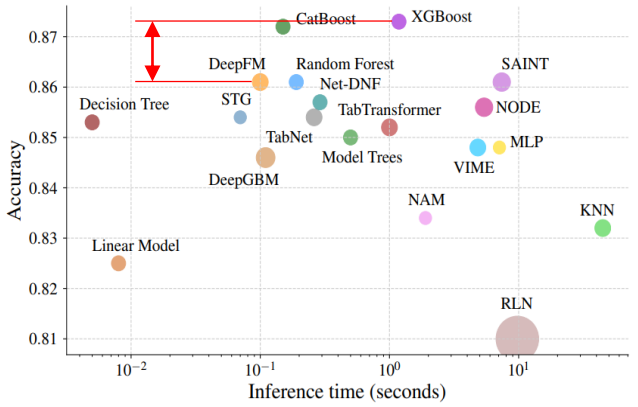
GCI2022WINTER
WEEK5 教師あり学習

作成者・講師：近藤　佑樹

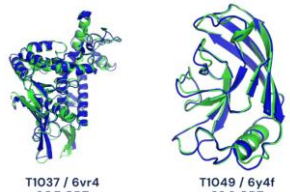
近年注目される深層学習とテーブルデータの関係



深層学習モデルとそれ以外の機械学習モデルの性能差．**未だにテーブルデータにおいては従来の機械学習モデルが優位.**

引用 : V. Borisov+ arXiv 2021

その他のテーブルデータに対する深層学習モデルの研究：
Y. Gorishniy+ NeurIPS 2021

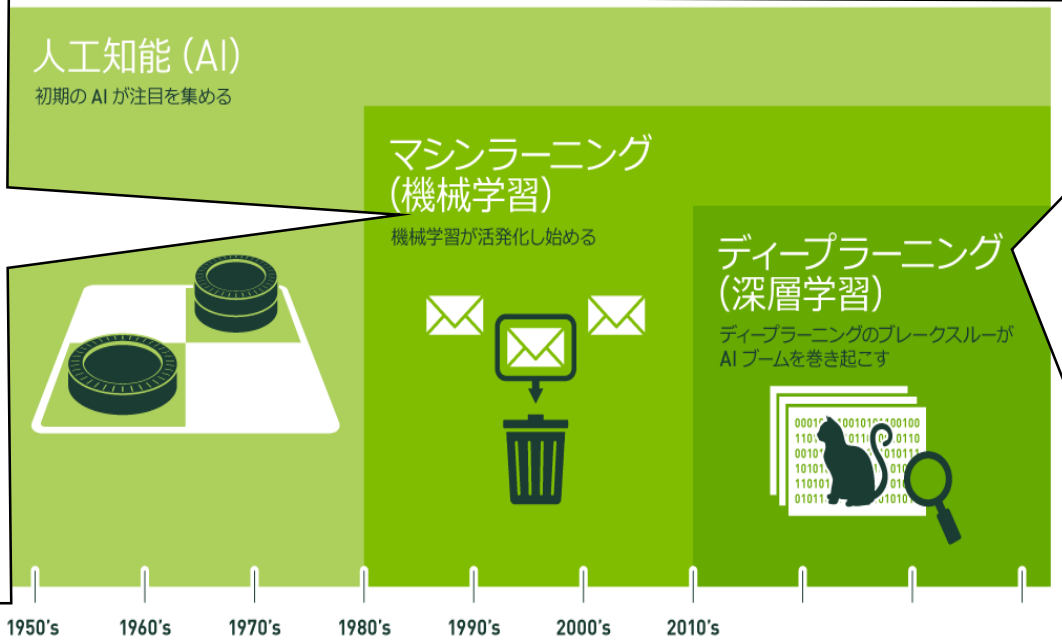


タンパク質構造計算
引用 :
https://github.com/deepmind/alphafold/blob/main/imgs/casp14_predictions.gif

Text2Image
引用 : https://github.com/CompVis/stable-diffusion

高解像度化(超解像)
引用 : C. Saharia+ TPAMI 2022

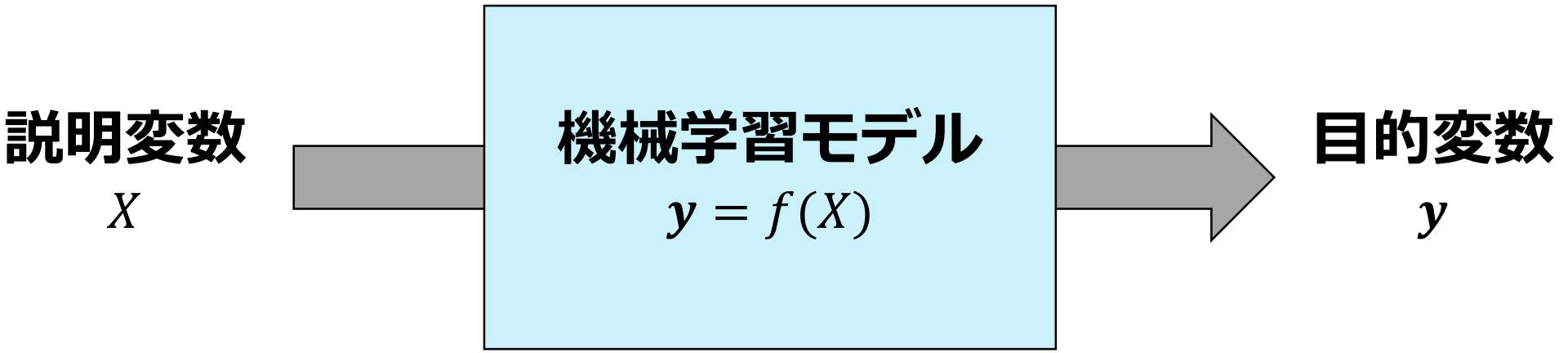
様々なモダリティの情報に対し，目覚ましい成果を挙げている



引用: https://blogs.nvidia.co.jp/2016/08/09/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

事前知識

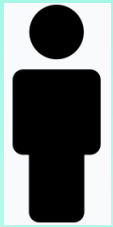
- 機械学習：
  データから**知識やパターンを理解させるアルゴリズムの総称.**
  アルゴリズムによる理解によって**予測，解釈などが可能.**
- 説明変数：
  機械学習モデルへの**入力**に用いられる変数．特徴量と呼ばれることもある.
- 目的変数：
  機械学習モデルの**出力**として定められる変数



機械学習の全体像

教師あり学習

- **正解データが定められた**学習法
- 入力データと正解データの関係性を**関数として近似**



- Age
- Sex
- Ticket

→ **Survived** or **Died**

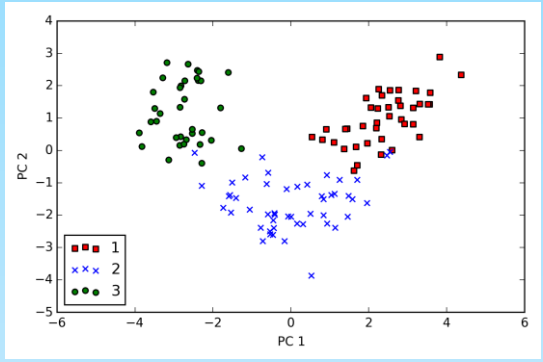
タイタニック号の生存者予測

- Kite
- Sea snake
- **Siberian husky**
- Drake
- ・
- ・
- ・

画像分類 (ImageNet)
引用 : J. Deng+ CVPR2009

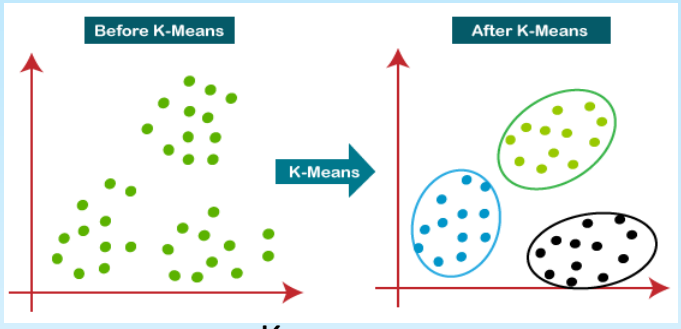
教師なし学習

- 正解データが定められていない学習法
- データの**潜在的なパターンを学習**



主成分分析

引用：https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch05/images/05_03.png

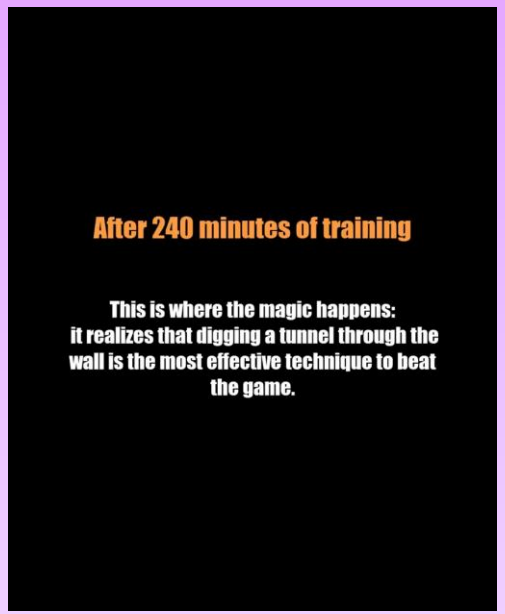


K-means

引用：https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

強化学習

- **報酬を最適化させる行動・知識を学習する**方法
- ロボット制御やゲーム等で応用されている



DQNを用いたAtariのプレイ動画
引用：https://www.youtube.com/watch?v=V1eYniJ0Rnk

＊GCIでは割愛

Week5の目標

1. **機械学習の大分類**が理解できる

2. **5つの機械学習モデルの概略**が理解できる

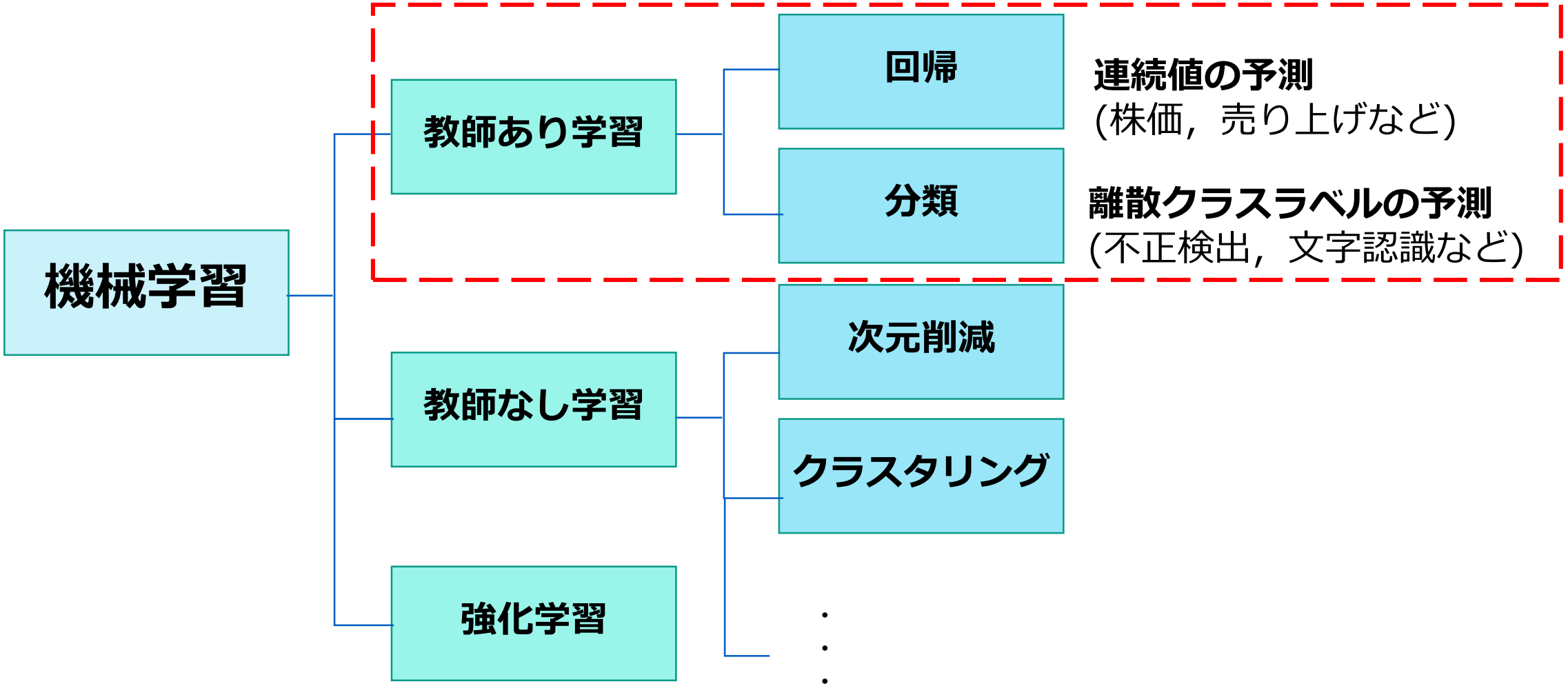
    1. **どのように動いているのか** (定性的な理解)

    2. **どういった特徴があるか？**

        ➡ **目的に応じた活用が可能**

3. 5つの機械学習モデルを**Sckit-learnで実装できる**

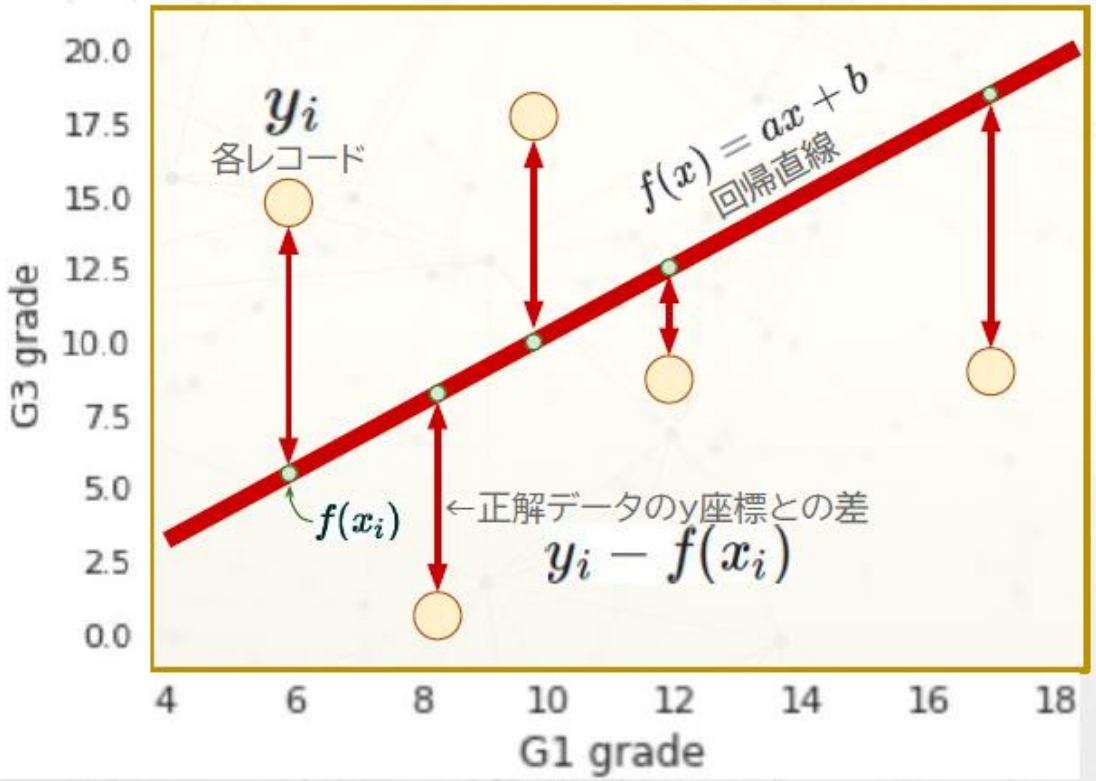
回帰と分類


機械学習
教師あり学習
回帰
分類
連続値の予測
(株価，売り上げなど)
離散クラスラベルの予測
(不正検出，文字認識など)
教師なし学習
次元削減
クラスタリング
強化学習

Notebook へ

単回帰分析の復習

最小二乗法による単回帰分析



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

誤差が最小になる
係数aと切片bを
求める（計算は自動）

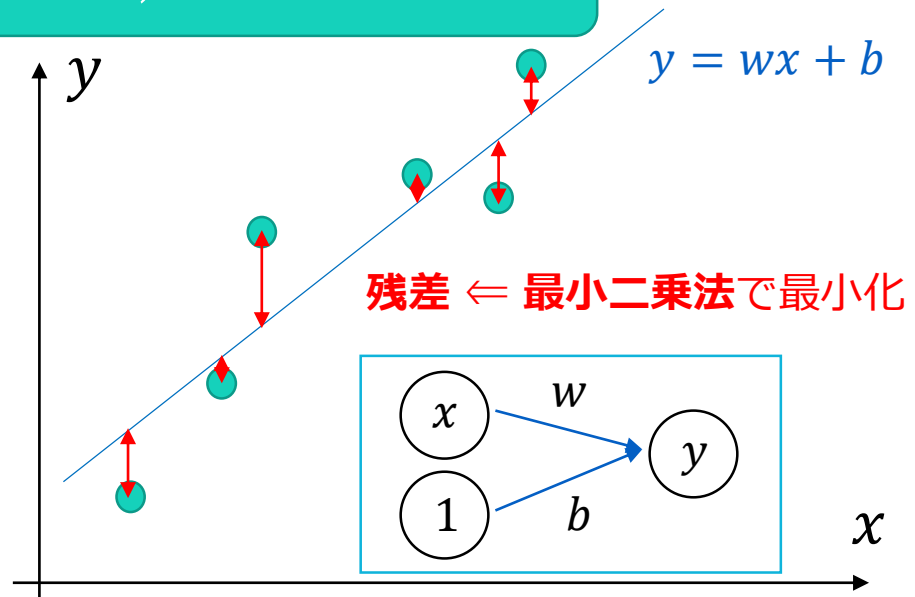
$$\sum_{i=1}^{n}(y_i - f(x_i))^2$$

二乗誤差

引用：GCI week4 スライドより

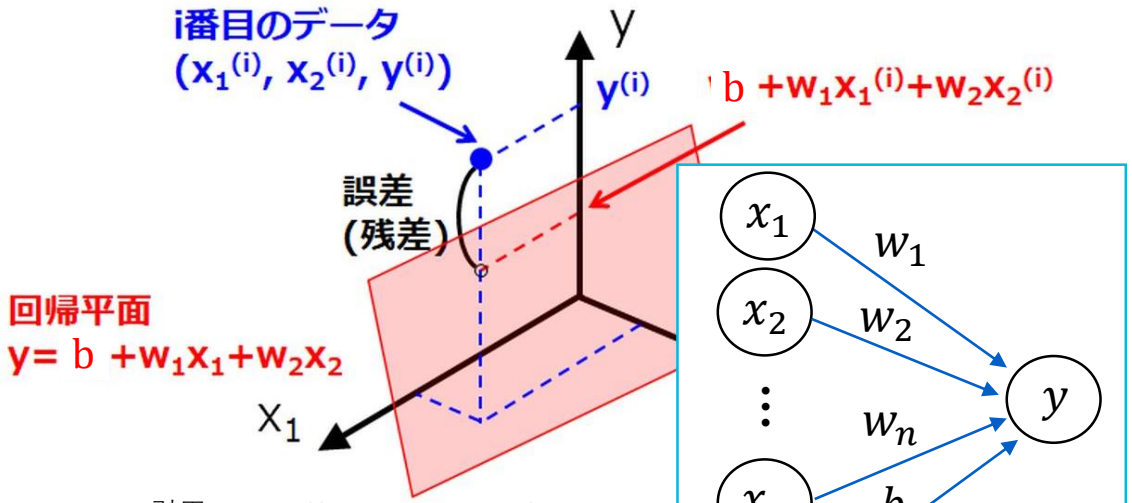
回帰分析の概要と説明性

(線形)単回帰分析



- 目的変数 1つ (y)
- 説明変数 1つ (x)
- 関数モデル $y = wx + b$
  ⇒ 推定パラメータ w, b

(線形)重回帰分析



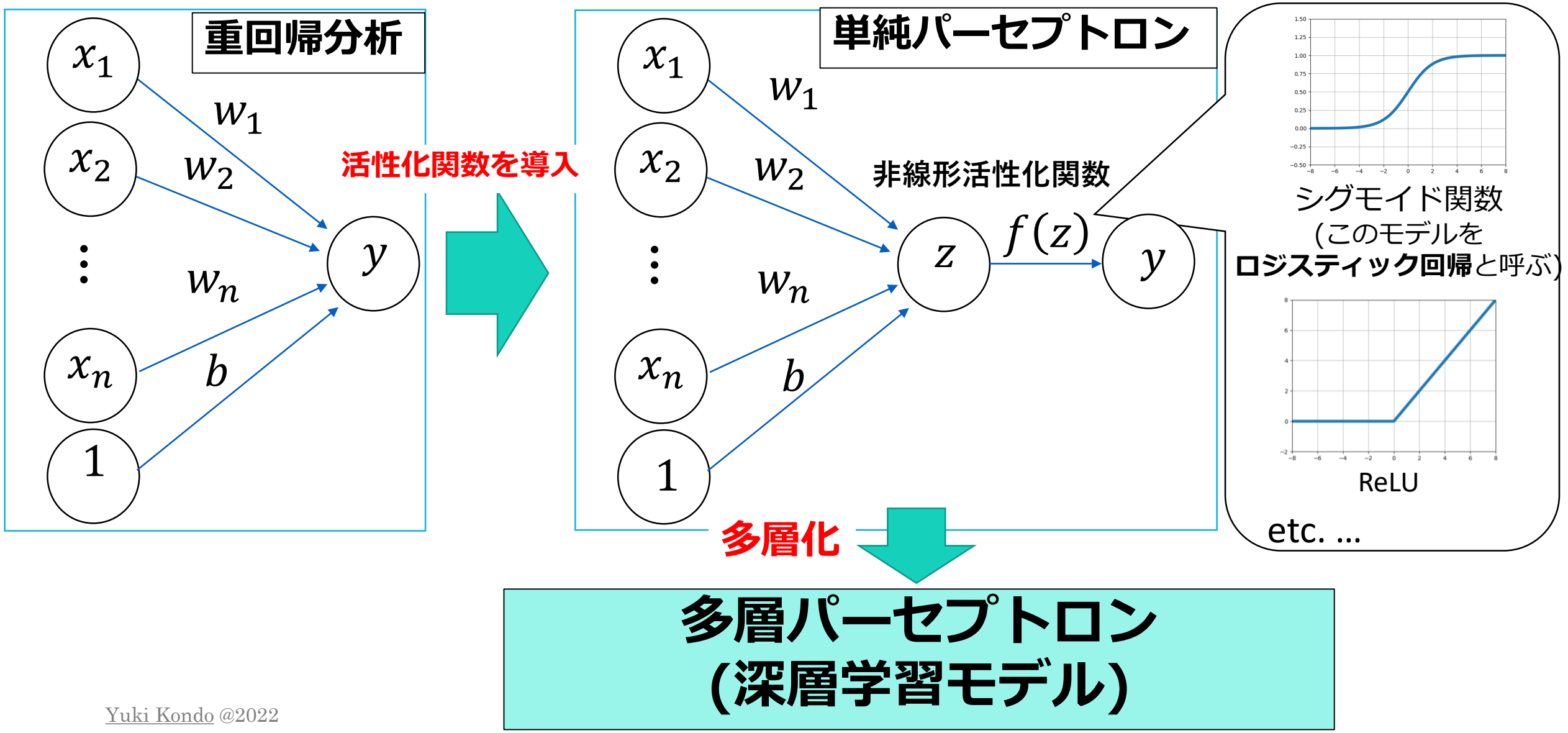
引用：https://speakerdeck.com/yuyumoyuyu

- 目的変数 1つ (y)
- 説明変数 **2つ以上** $(x_1, x_2, \ldots, x_n)$
- 関数モデル $y = \sum_{i=1}^{n} w_i x_i + b$
  ⇒ 推定パラメータ $w_1, w_2, \ldots, w_n, b$

- 重みを元に目的変数に対する説明変数の寄与を考えられる*
- 線形モデルのため，根拠が明確で扱いやすい

*特徴量スケーリング実施済みの場合．ただし多重共線性がある場合は，この考察は困難．

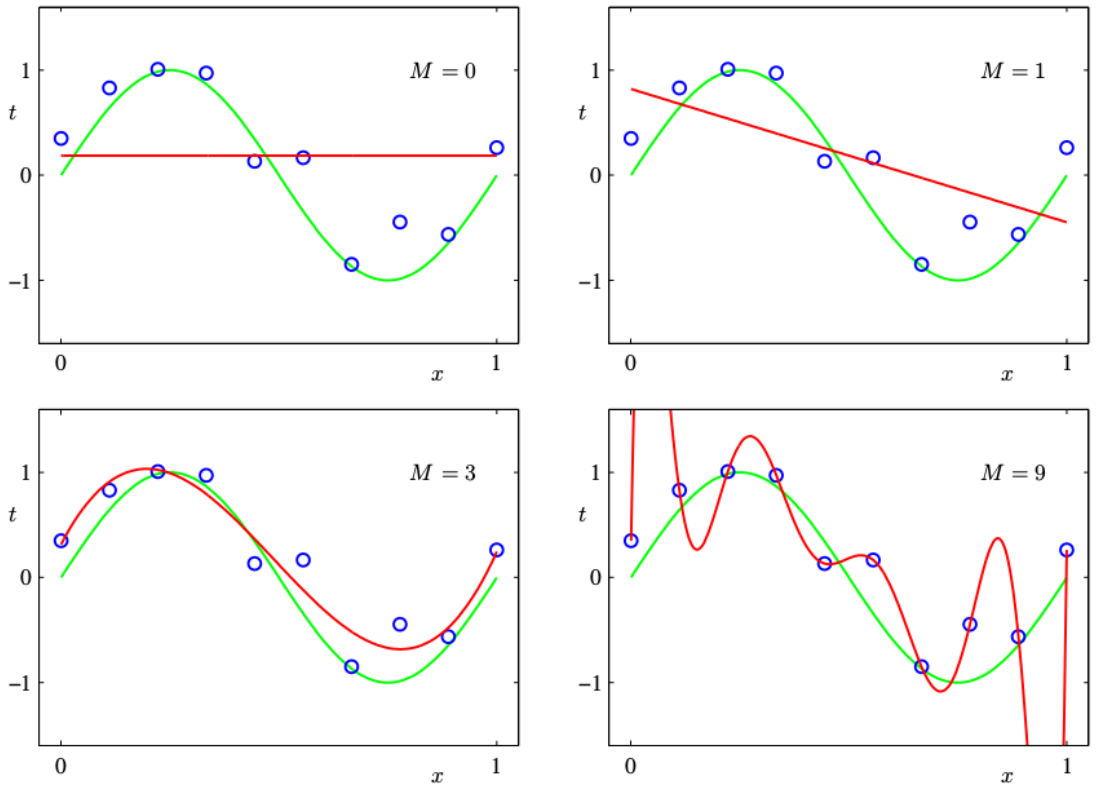
重回帰分析と深層学習


重回帰分析
x_1
x_2
x_n
1
w_1
w_2
w_n
b
y
活性化関数を導入
単純パーセプトロン
z
$f(z)$
非線形活性化関数
シグモイド関数
(このモデルを
ロジスティック回帰と呼ぶ)
ReLU
etc. …
多層化
多層パーセプトロン
(深層学習モデル)

Notebook へ

過学習

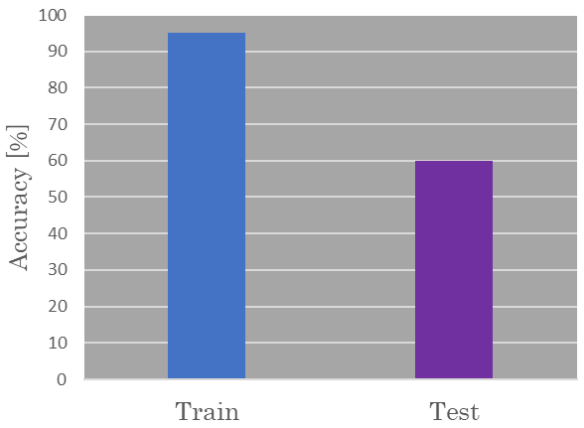
学習データに過剰にフィットすることで，**モデル化した関数**が**真のデータの関数**から離れてしまう状態



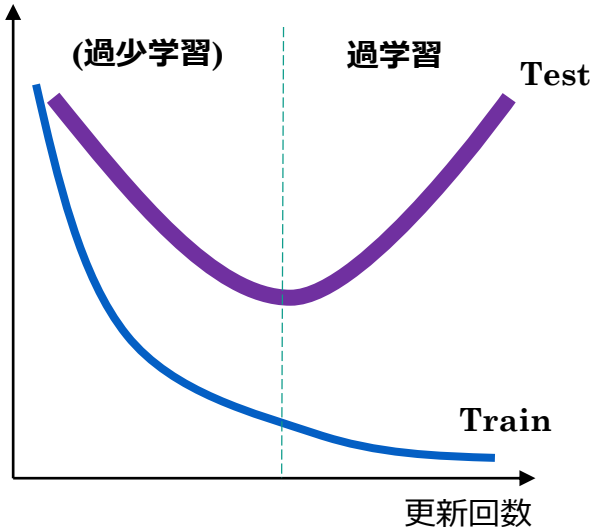
M=0, 1ではデータの関数の表現力が不足しており，M=3ではうまく表現できている．M=9では過学習が発生している．

引用 : C. Bishop “Pattern Recognition and Machine Learning”.

確認方法：
学習データのスコアに対し，**テストデータのスコア**が**大きく低下する**場合，**過学習しているとみなす**.



テストでスコアが35%も低下しているため，過学習していると言える.

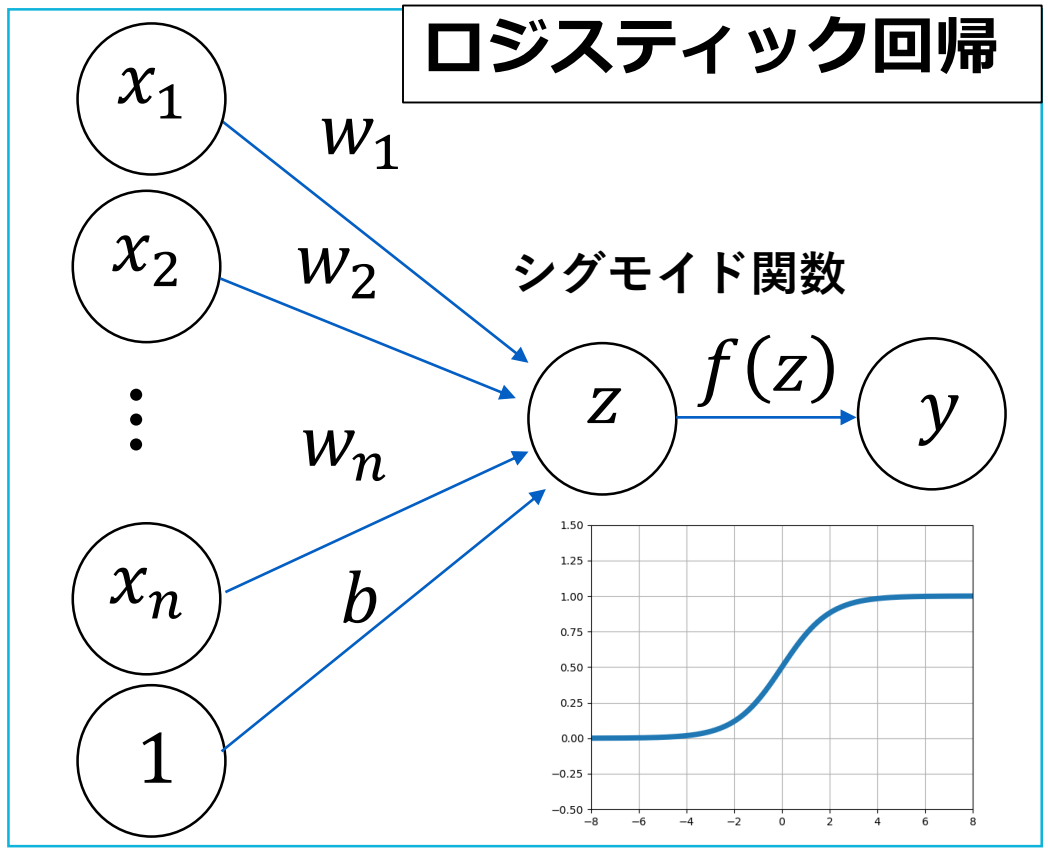


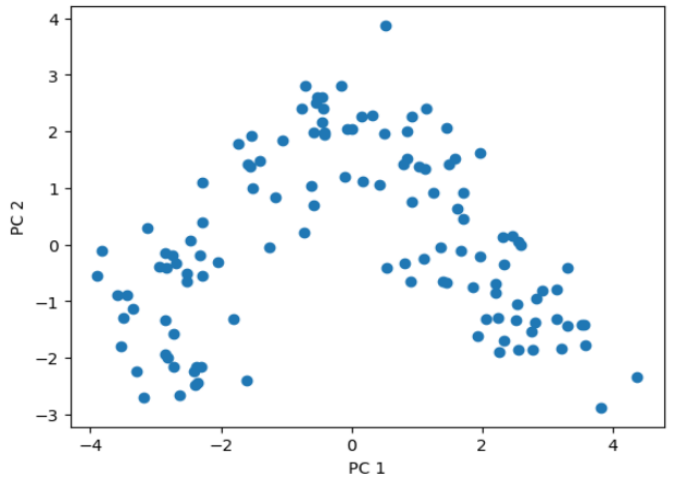
逐次パラメータを更新する場合，モデルが更新するほど，学習データにフィットするため，過度な学習は過学習につながる.

Notebook へ

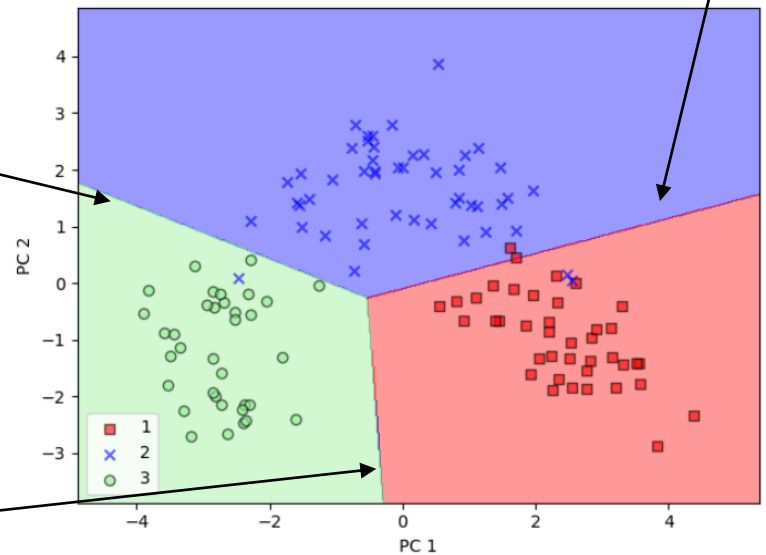
ロジスティック回帰 (⇐)

- 名前に回帰が付いているが，**分類問題に利用**する．
- **線形な決定境界**(分類する境界)が形成される
  * **超平面**とも言う





線形な決定境界



引用：https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch05/ch05.ipynb

Notebook へ

過学習と正則化

【考えうる過学習の要因】

モデルが過度に複雑(＝モデルのパラメータ数が多すぎる)

➡ **モデルのパラメータ数減少，パラメータの値の大きさ抑制**が有効

正則化：**モデルのパラメータの値の大きさ(複雑さ)**に
ペナルティを与え，**過学習を防ぐ**

【**正則化項**が加えられた損失関数 L 】

$$L = \sum_{i=1}^{n} \left(y_i - f(\boldsymbol{x_i})\right)^2 + R(\boldsymbol{w})$$

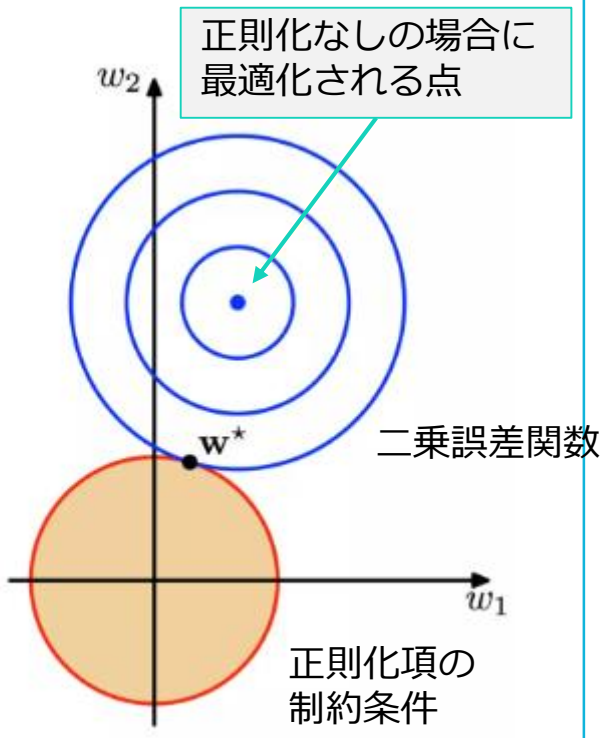
二乗誤差関数　　**正則化項**

リッジ回帰・ラッソ回帰

リッジ(Ridge)回帰

$$L = \sum_{i=1}^{n} \left(y_i - f(\boldsymbol{x_i})\right)^2 + \lambda \sum_{j=1}^{m} w_j^2$$

- 正則化項によって，**w^*に最適化**.
- パラメータを**全体的に小さくする**作用



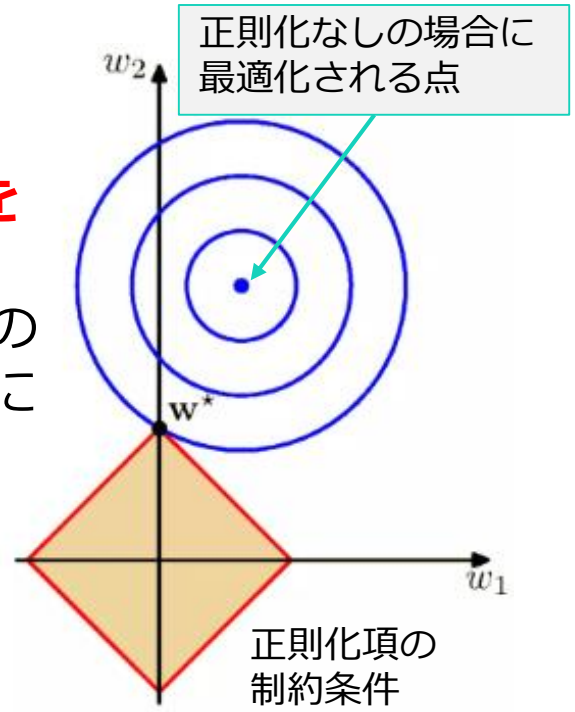
引用 : C. Bishop “Pattern Recognition and Machine Learning”.

- 余談 -
リッジ回帰とラッソ回帰の正則化項両方を加えたモデルを **Elastic Net** という.

ラッソ(LASSO)回帰

$$L = \sum_{i=1}^{n} \left(y_i - f(\boldsymbol{x_i})\right)^2 + \lambda \sum_{j=1}^{m} |w_j|$$

- 正則化項によって，**w^*に最適化**.
- **いくつかのパラメータを0にする**作用
  ➡ 実質的に元のデータの説明変数を除くことに相当
  (**スパースモデリング**という)

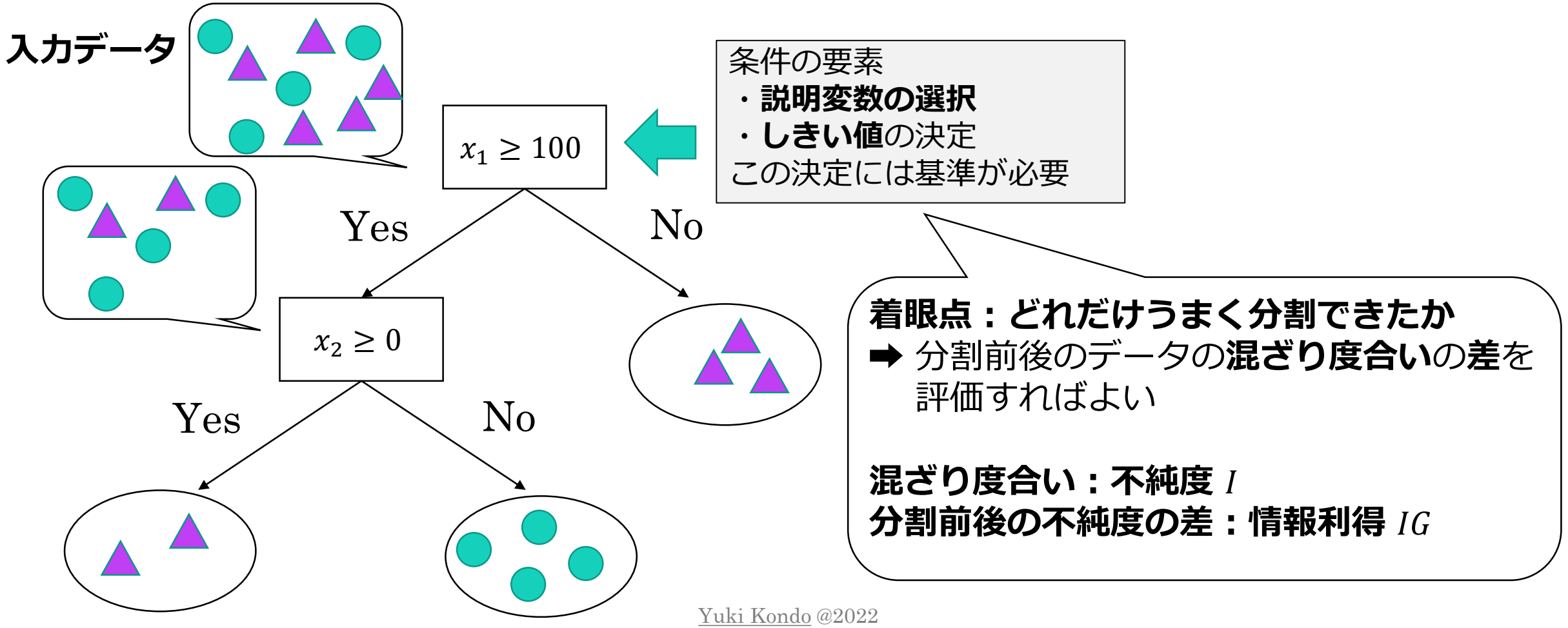


引用 : C. Bishop “Pattern Recognition and Machine Learning”.

Notebook へ

決定木

- **分類問題・回帰問題のいずれでも利用可能**
  - 分類の場合：分類木 ⇐ 以下の説明では**分類木で説明**
  - 回帰の場合：回帰木
- ノードごとに**一つの説明変数に注目し，データに質問して分ける**.

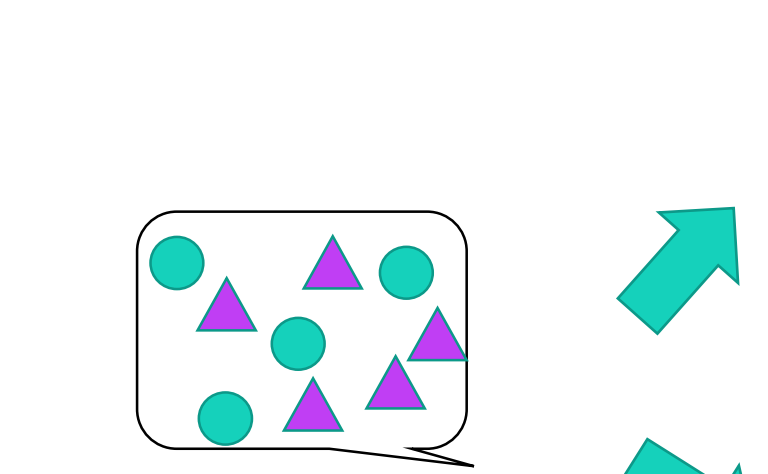


不純度の計算

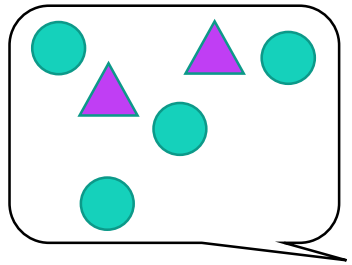
【不純度】

- 複数種類ある
  - (シャノン)エントロピー：$I_H = -\sum_{i=1}^{C} p_i \log_2 p_i$
  - ジニ不純度：$I_G = 1 - \sum_{i=1}^{C} p_i^2$
  - 分類誤差：$I_E = 1 - \max_{i=1,\ldots,C}(p_i)$

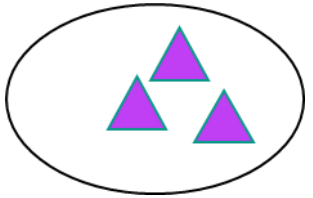
【計算例】



$I_H^{(1)} = -(\frac{5}{9}\log_2\frac{5}{9} + \frac{4}{9}\log_2\frac{4}{9})$
=0.9911



$I_H^{(2left)} = -(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6})$
=0.9183



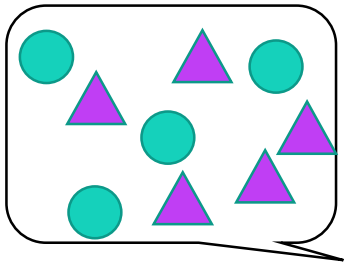
$I_H^{(2right)} = -(\frac{3}{3}\log_2\frac{3}{3}) =$ **0**

情報利得の計算

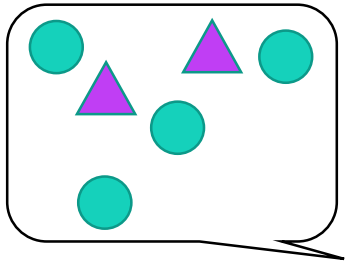
【情報利得】

- 2分決定木の場合
  - $IG = I^{(p)} - \frac{N_{left}}{N_p} I^{(left)} - \frac{N_{right}}{N_p} I^{(right)}$

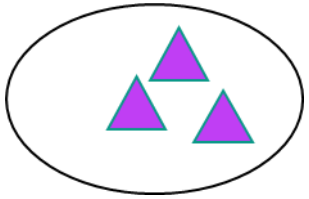
【計算例】



$I_H^{(1)} = -(\frac{5}{9}\log_2\frac{5}{9} + \frac{4}{9}\log_2\frac{4}{9})$
=0.9911



$I_H^{(2left)} = -(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6})$
=0.9183



$I_H^{(2right)} = -(\frac{3}{3}\log_2\frac{3}{3}) =$ **0**

左図より

$$IG = I_H^{(1)} - \frac{N_{left}}{N_p} I_H^{(2left)} - \frac{N_{right}}{N_p} I_H^{(2right)}$$

$$= 0.9911 - \frac{6}{9} \times 0.9183 - \frac{3}{9} \times 0$$

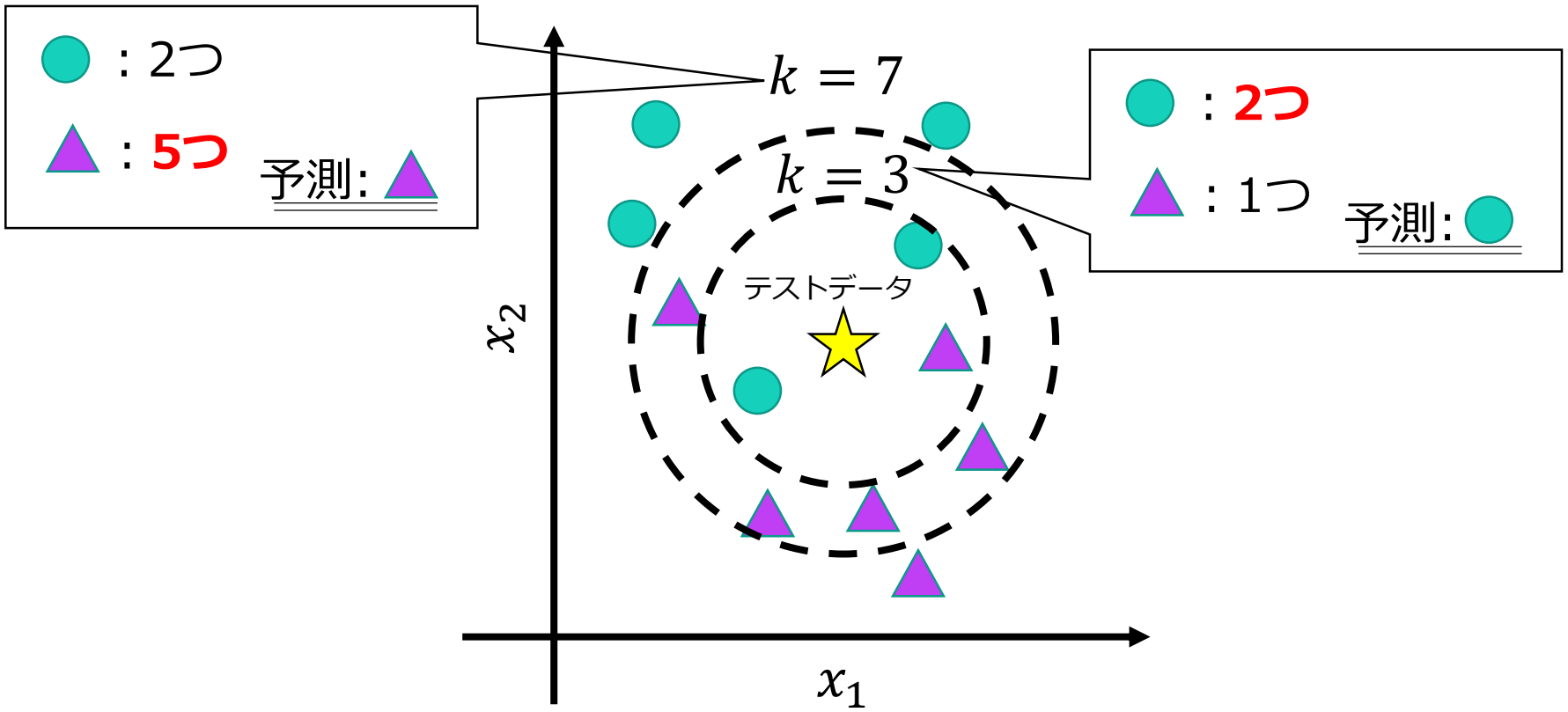
$$= \mathbf{0.3789}$$

この情報利得を最大化する
ルールを求める

Notebook へ

k-NN (k近傍法)

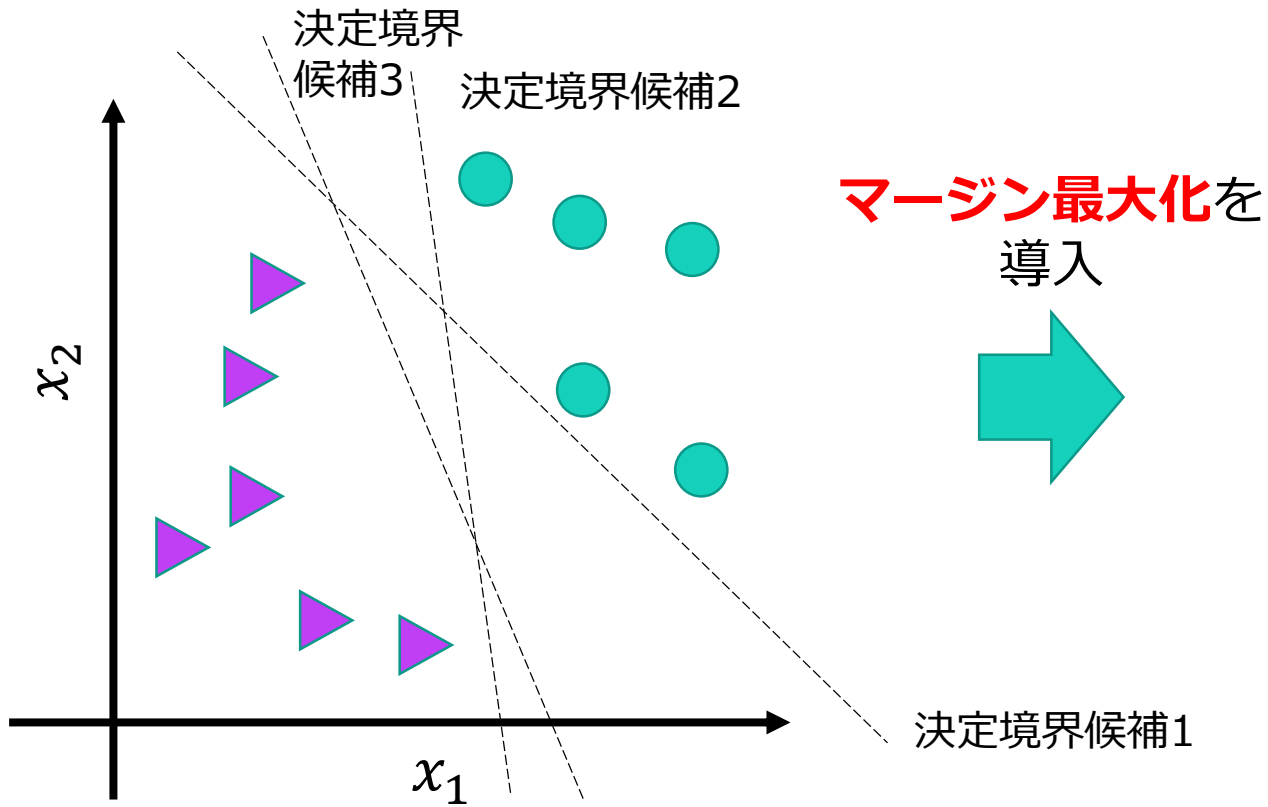
- **分類問題・回帰問題のいずれでも利用可能** (以下では分類で説明)
- 特徴量空間の**テストデータ近傍のk個の学習データの正解ラベルから推定**する
  - 分類の場合：多数決，重み付け多数決等を利用
  - 回帰の場合：平均値，中央値，重み付け平均値等を利用



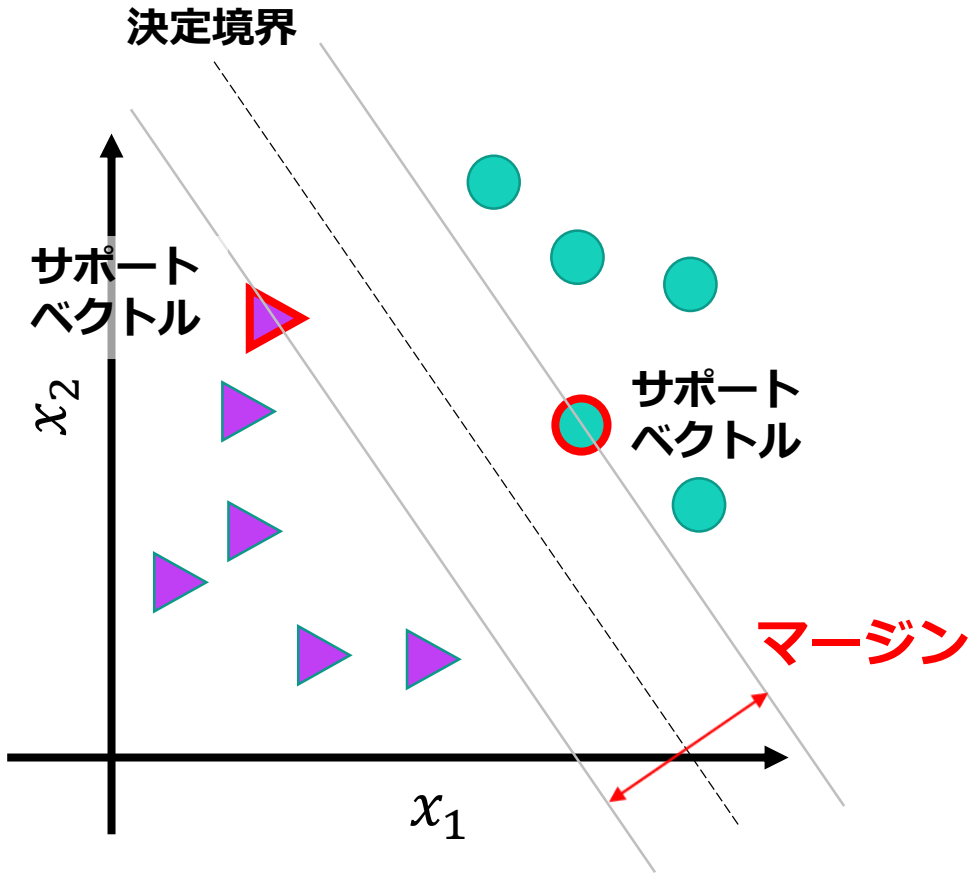
Notebook へ

サポートベクターマシン（SVM）

- **分類問題・回帰問題のいずれでも利用可能**（以下では分類で説明）
- **クラス間のマージンを最大化する条件下で決定境界を定める.**



マージン最大化を
導入



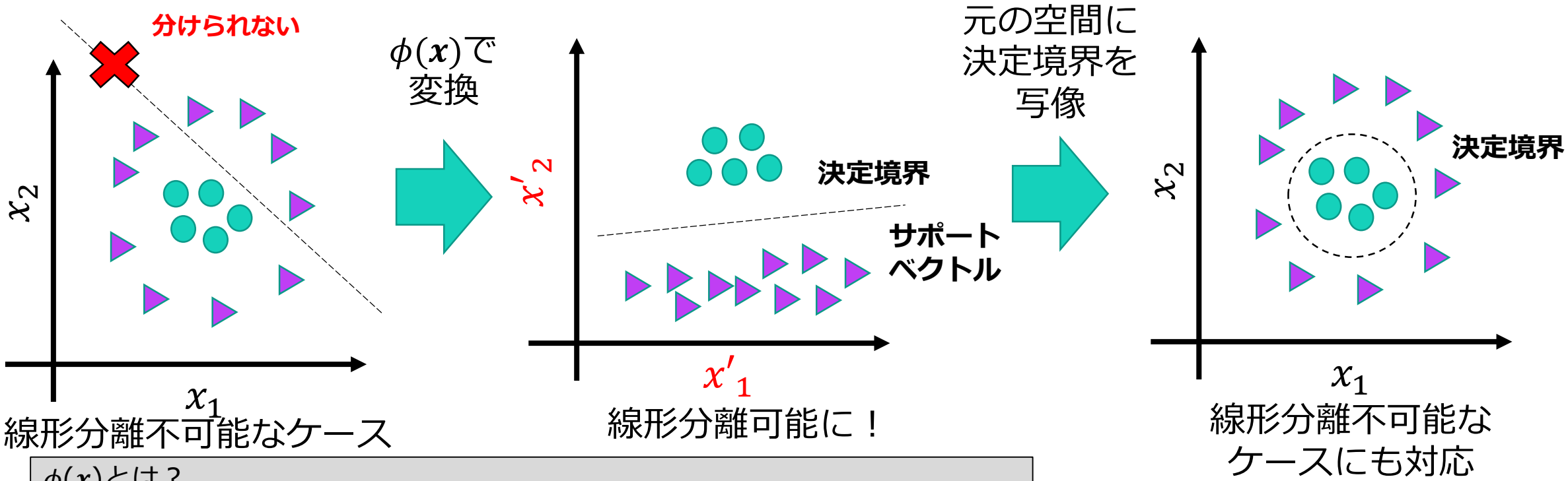
いずれの決定境界でも分類は可能だが，
汎化性は低そう…

非線形サポートベクターマシン（SVM）

SVMは線形な決定境界を形成 ➡ **線形分離不可な問題は解けない**

この問題に対応するのが**非線形SVM**

(scikit-learnでは **sklearn.svm.SVC**が対応．)



$\phi(x)$とは？
元の空間よりも高次元な空間に写像する関数．用いるカーネル関数によって，$\phi(x)$は異なる．カーネルトリックによって，計算量を減らす．

Notebook へ