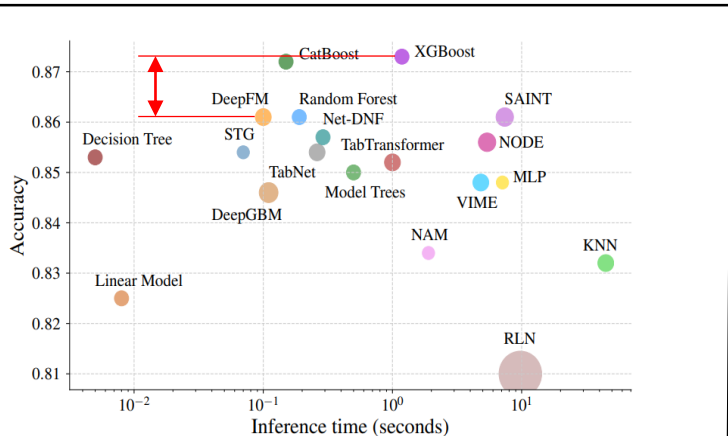


GCI2022WINTER WEEK5 教師あり学習

作成者・講師：近藤 佑樹

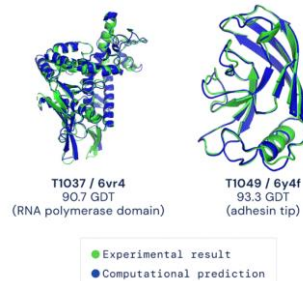
近年注目される深層学習とテーブルデータの関係² /27



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

人工知能 (AI)

初期の AI が注目を集める

マシンラーニング
(機械学習)

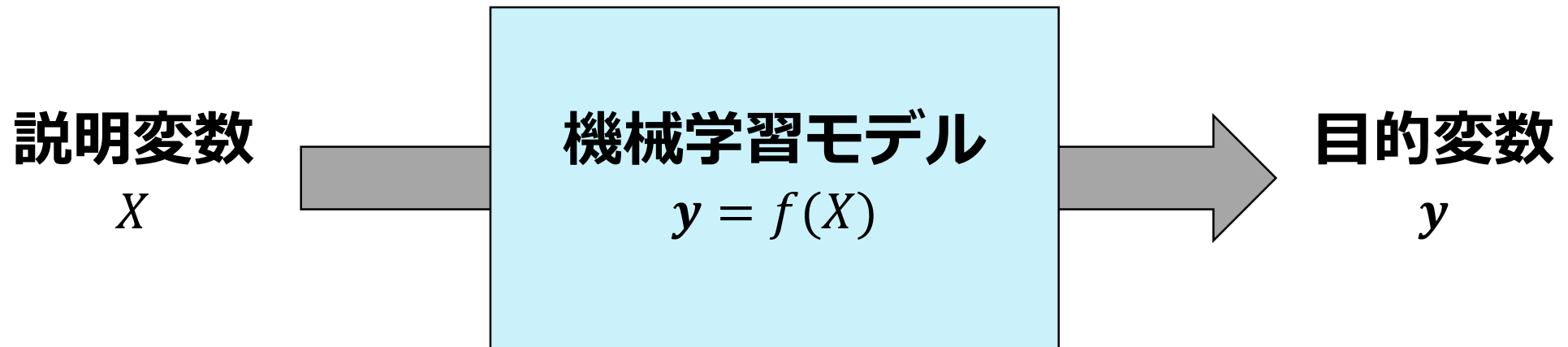
機械学習が活発化し始める

ディープラーニング
(深層学習)

ディープラーニングのブレイクスルーが
AI ブームを巻き起こす

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

- 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



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



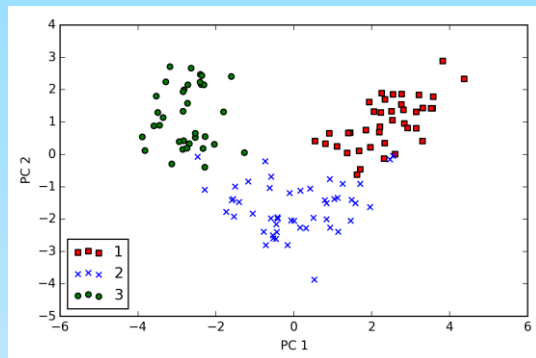
- Kite
- Sea snake
- Siberian husky
- Drake
- ...

画像分類 (ImageNet)

引用 : J. Deng+ CVPR2009

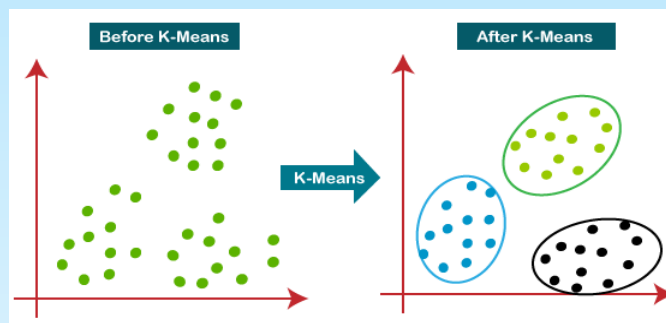
教師なし学習

- 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

After 240 minutes of training

This is where the magic happens:
it realizes that digging a tunnel through the wall is the most effective technique to beat the game.

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

1. 機械学習の大分類が理解できる

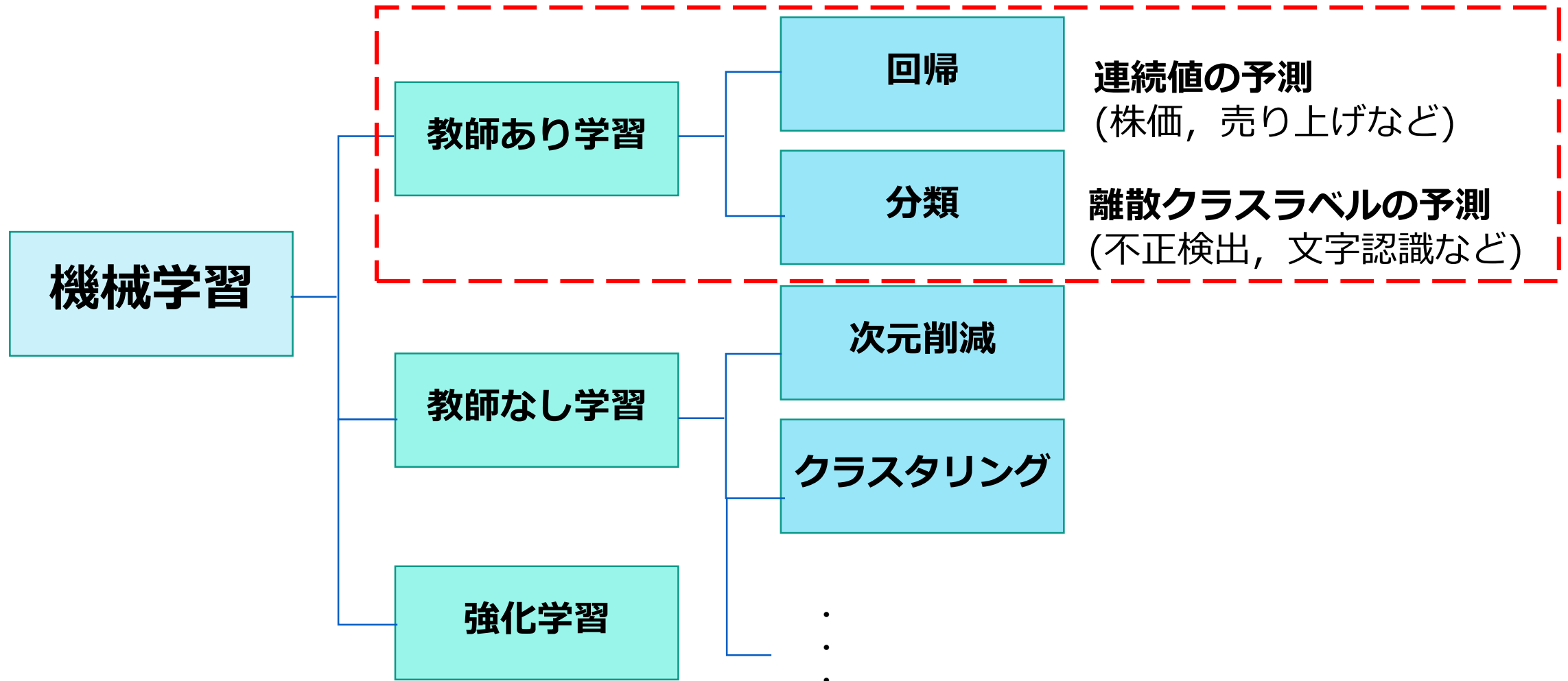
2. 5つの機械学習モデルの概略が理解できる

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

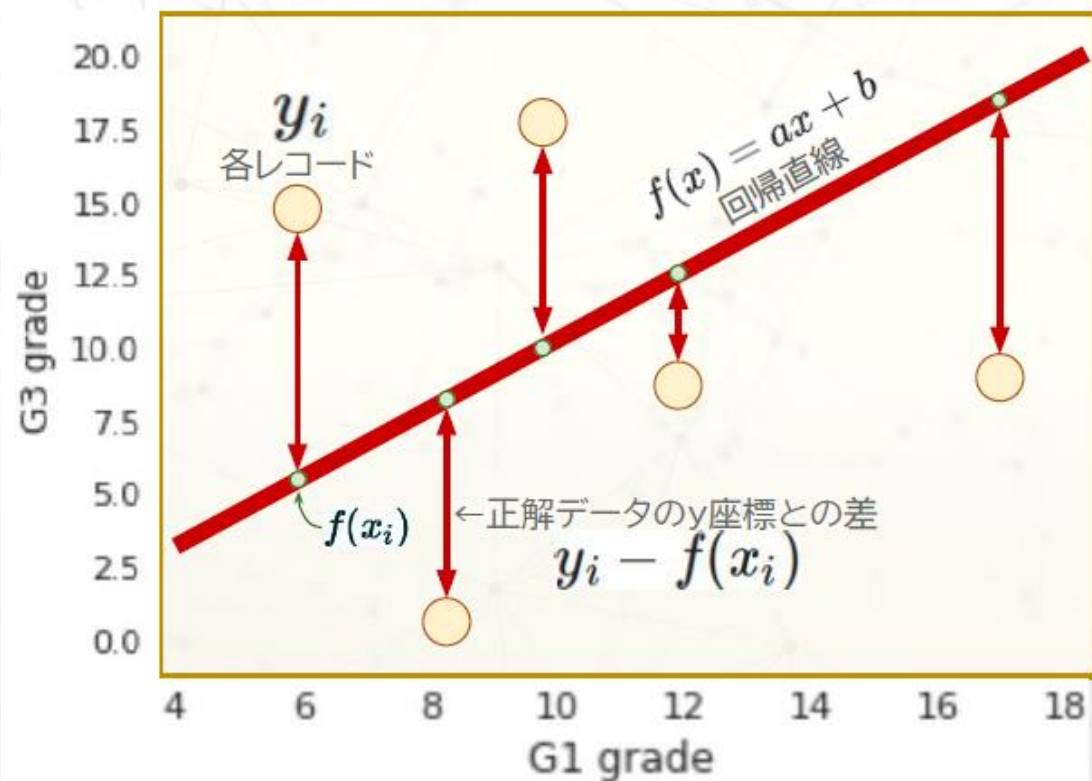
3. 5つの機械学習モデルをSckit-learnで実装できる



The background of the slide is a dark gray or black, featuring a pattern of numerous overlapping circles in various shades of gray. These circles vary in size and opacity, creating a bokeh-like effect. On the right side, there is a bright, glowing light source that creates a lens flare effect, with several bright white and light gray circles radiating from it.

Notebook ^

最小二乗法による単回帰分析



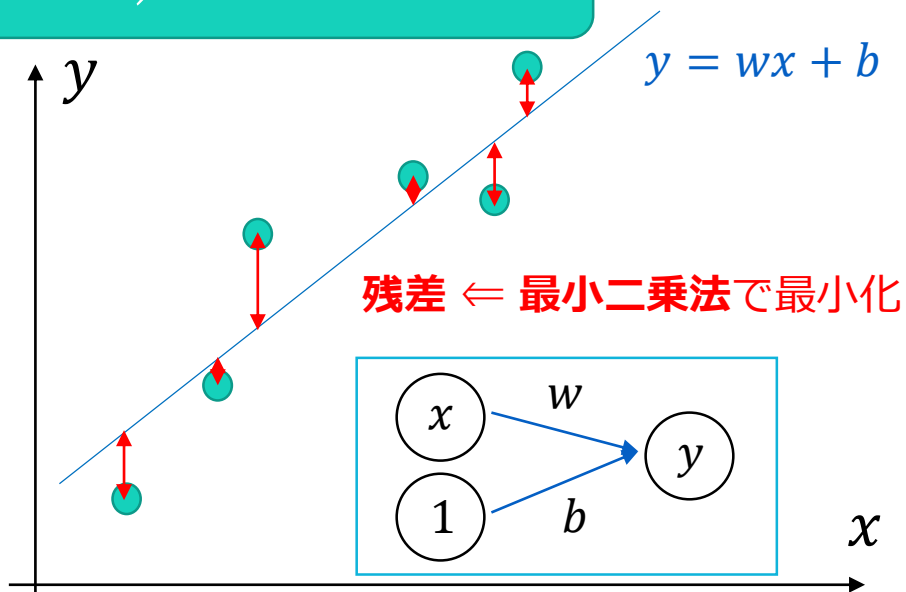
$$f(x) = ax + b$$

誤差が最小になる
係数aと切片bを
求める(計算は自動)

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

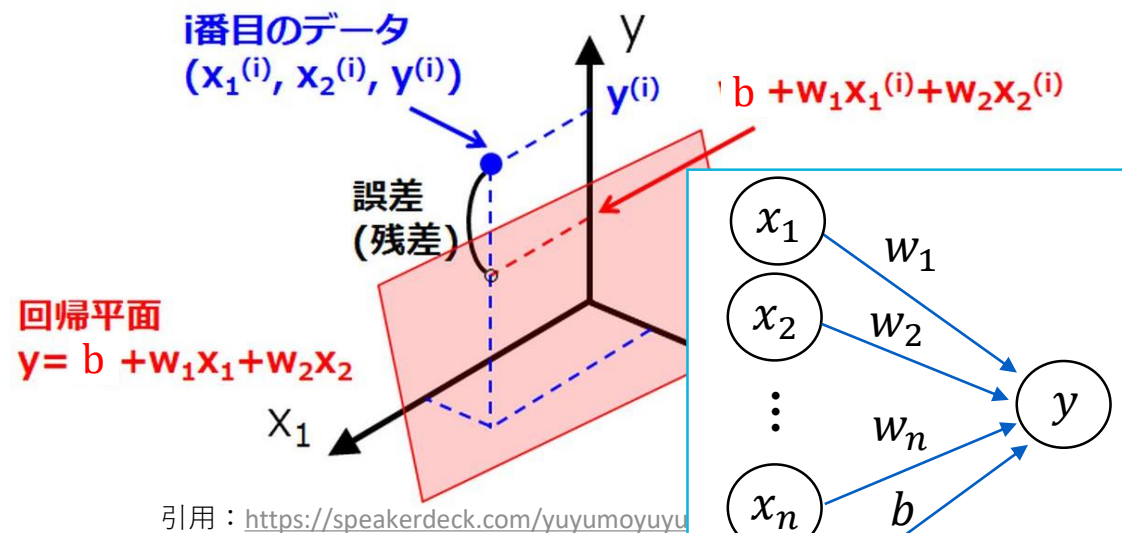
二乗誤差

(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

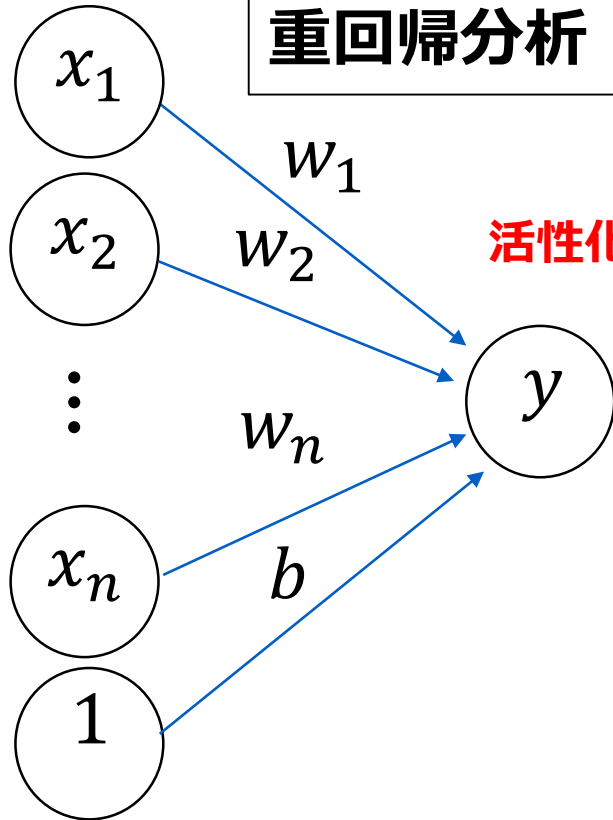


- 目的変数 1つ (y)
- 説明変数 **2つ以上** (x_1, x_2, \dots, x_n)
- 関数モデル $y = \sum_{i=1}^n w_i x_i + b$
⇒ 推定パラメータ w_1, w_2, \dots, w_n, b

- 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

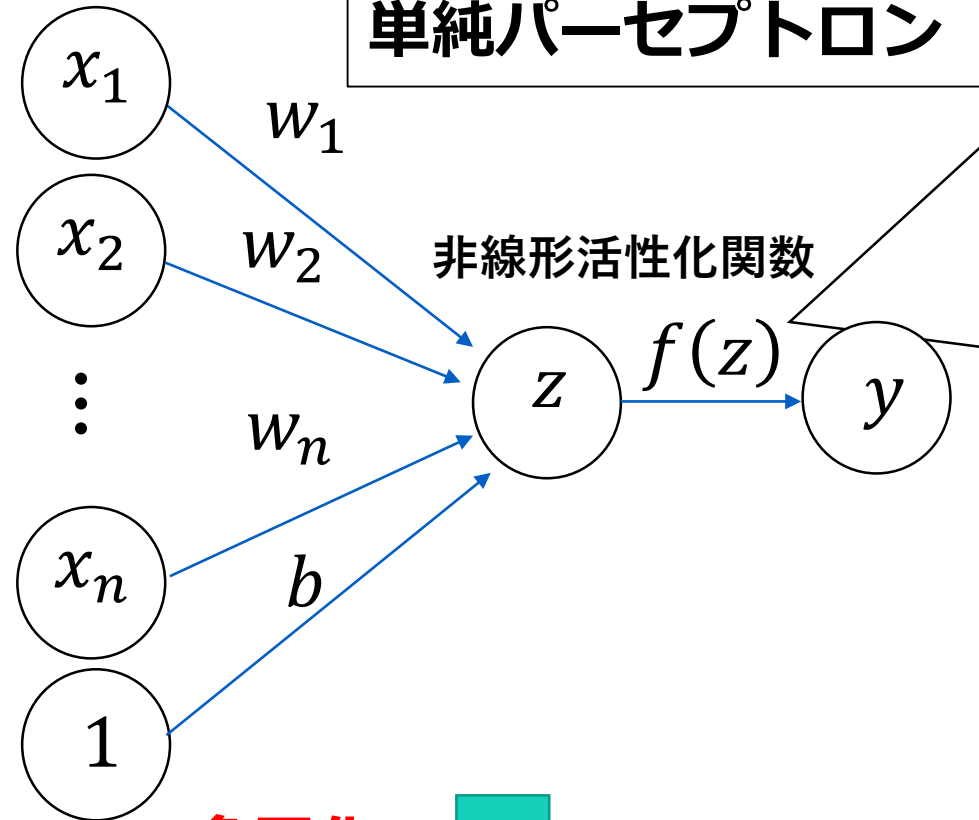
*特徴量スケーリング実施済みの場合。ただし多重共線性がある場合は、この考察は困難。

重回帰分析

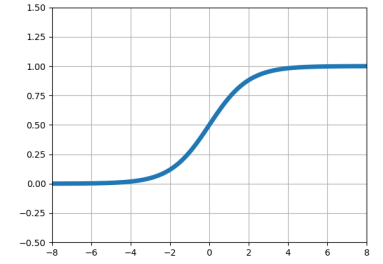


活性化関数を導入

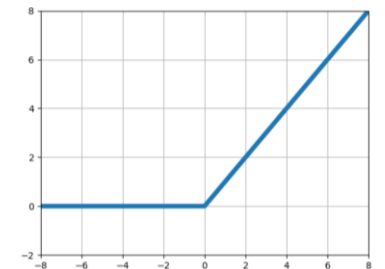
単純パーセプトロン



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. ...

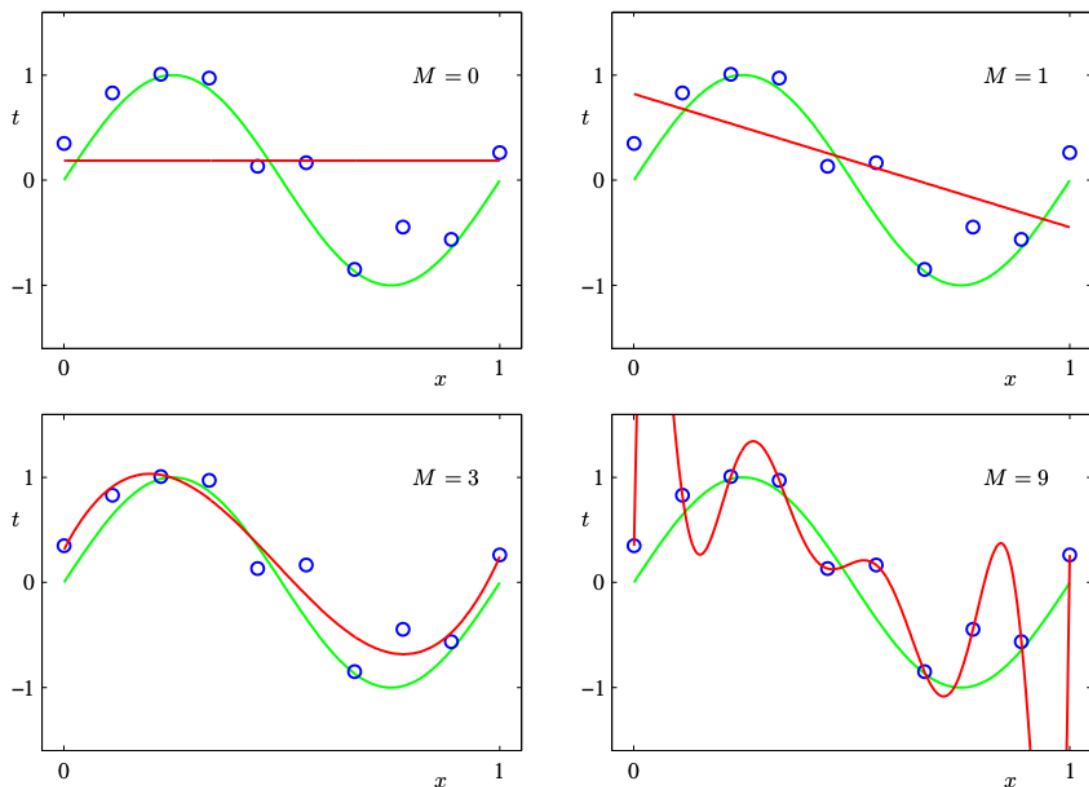
多層化

多層パーセプトロン (深層学習モデル)

The background of the slide is a dark gray or black, featuring a pattern of numerous overlapping circles in various shades of gray. These circles vary in size and opacity, creating a bokeh-like effect. Some circles are more prominent and brighter, while others are faint and blend into the background. The overall composition is abstract and modern.

Notebook ^

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

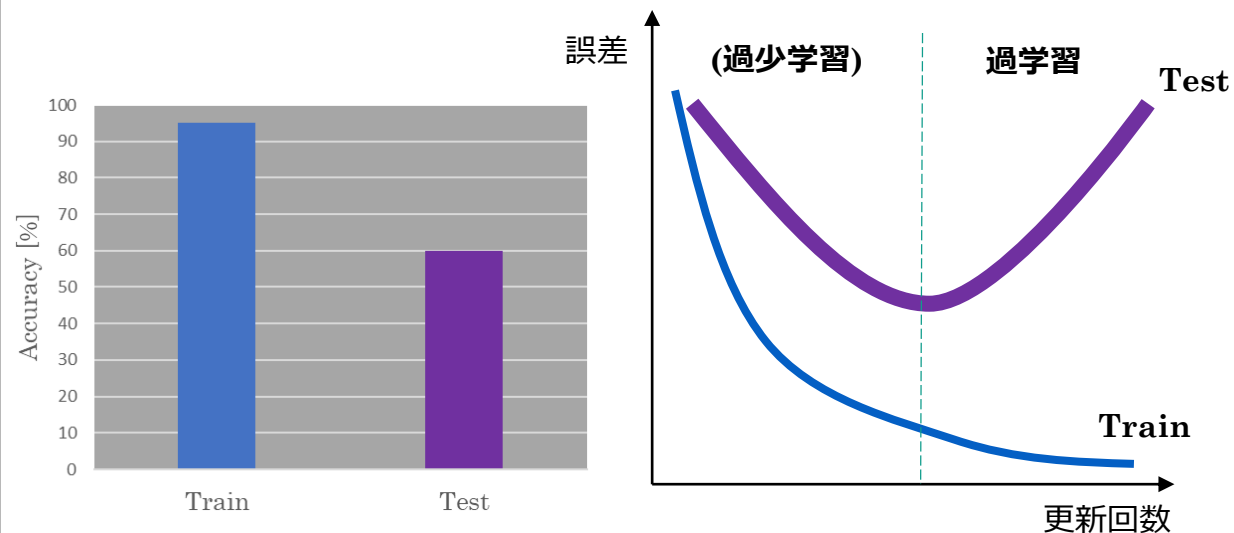


$M=0, 1$ ではデータの関数の表現力が不足しており、 $M=3$ ではうまく表現できている。 $M=9$ では過学習が発生している。

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。

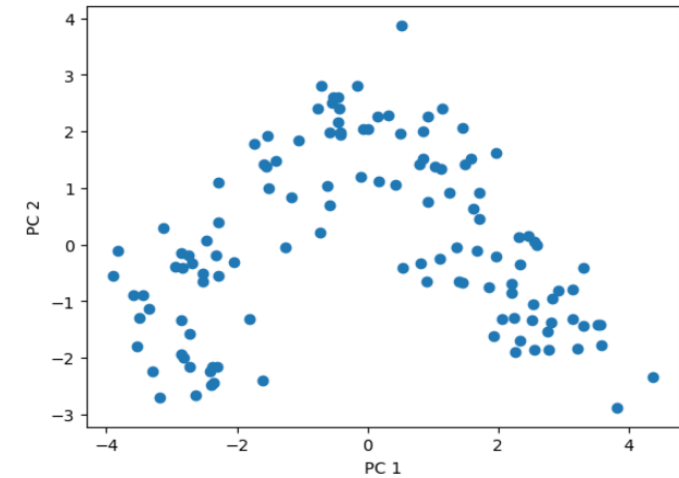
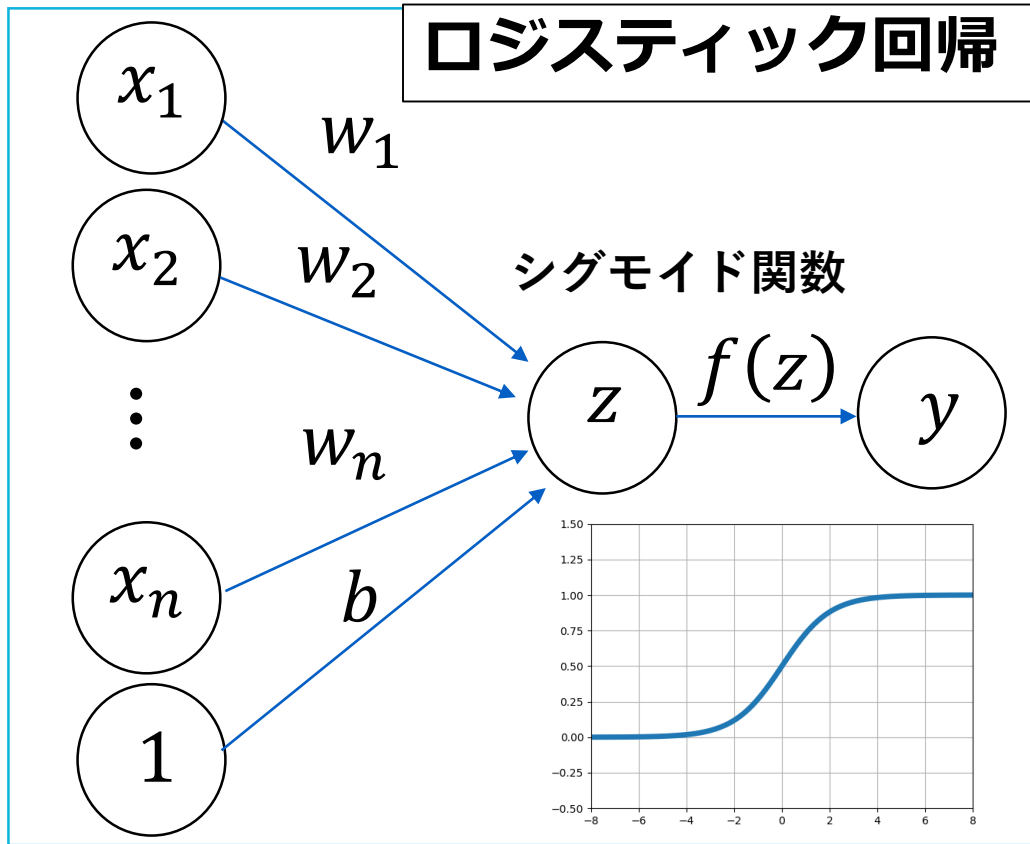
逐次パラメータを更新する場合、
モデルが更新するほど、学習
データにフィットするため、過
度な学習は過学習につながる。

The background of the slide is a dark gray or black, featuring a pattern of numerous overlapping circles in various shades of gray. These circles vary in size and opacity, creating a bokeh-like effect. Some circles are more prominent and brighter, while others are faint and blend into the background. The overall composition is abstract and modern.

Notebook ^

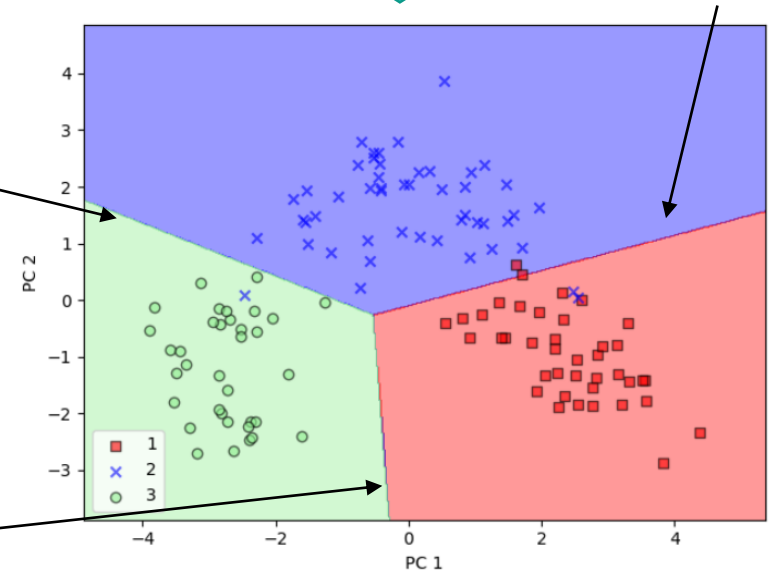
ロジスティック回帰 (⇐)

- 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

線形な決定境界



線形な決定境界

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

The background of the slide is a dark gray to black gradient, overlaid with a pattern of numerous overlapping circles. These circles vary in size and opacity, creating a bokeh or bubble effect. Some circles are solid dark gray, while others are lighter, appearing as if they are glowing or have a light source behind them. The overall composition is abstract and modern.

Notebook ^

【考える過学習の要因】

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**

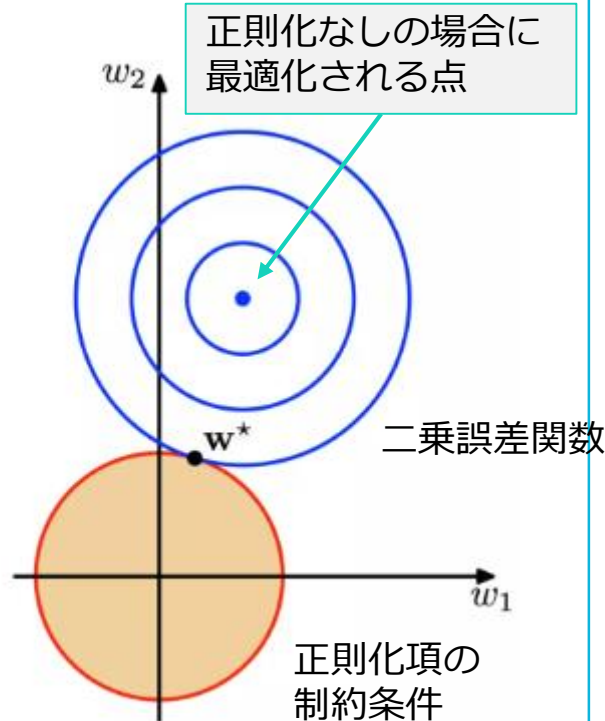
【**正則化項**が加えられた損失関数 L 】

$$L = \underbrace{\sum_{i=1}^n (y_i - f(x_i))^2}_{\text{二乗誤差関数}} + \underbrace{R(\mathbf{w})}_{\text{正則化項}}$$

リッジ(Ridge)回帰

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

- 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



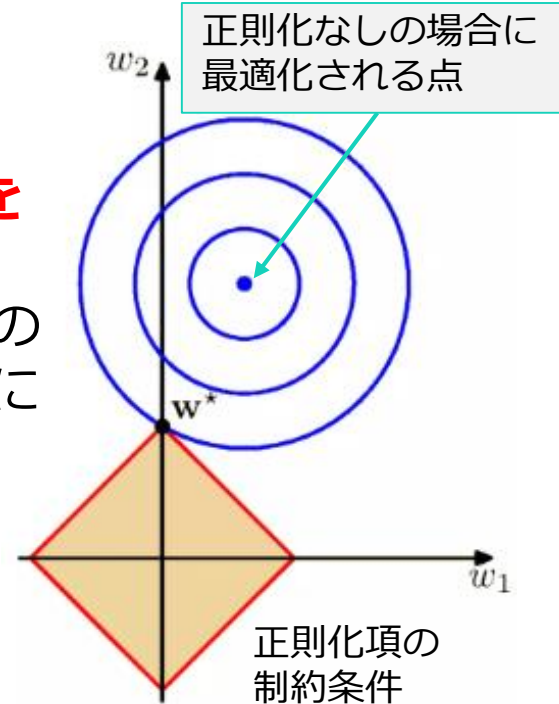
引用 : C. Bishop "Pattern Recognition and Machine Learning".

- 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

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

- 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)



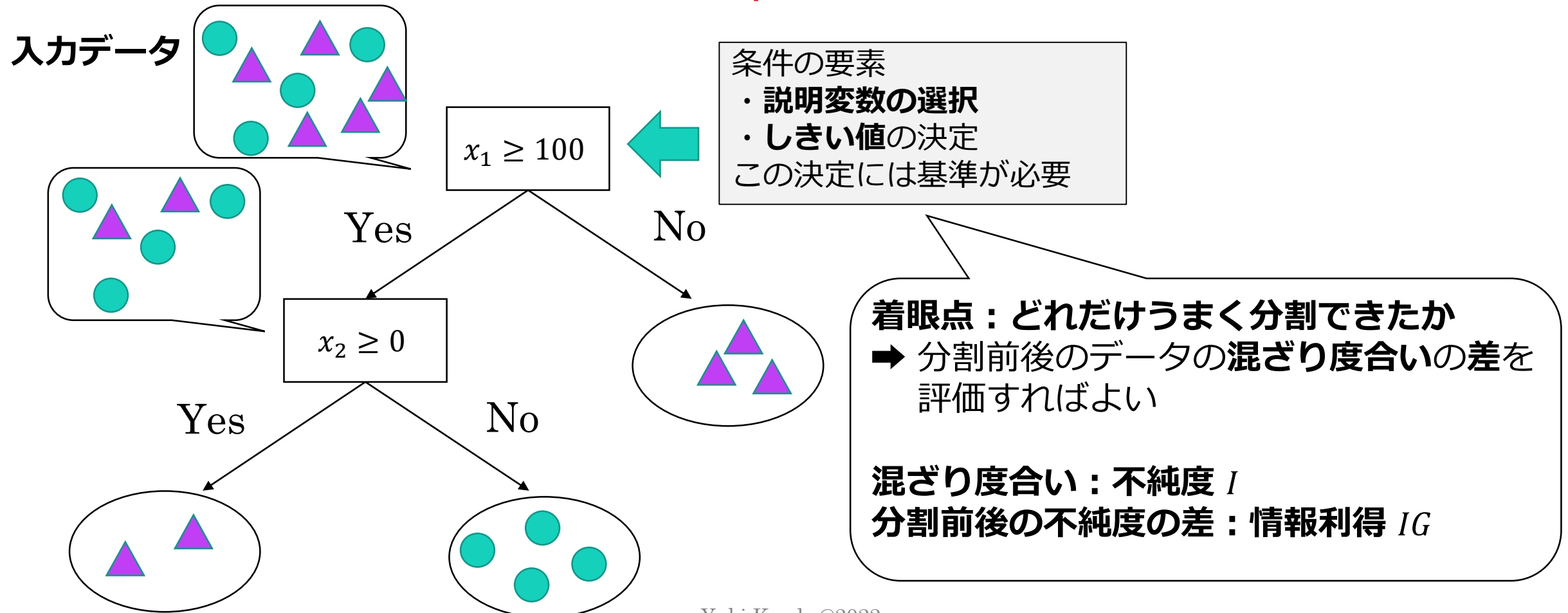
引用 : C. Bishop "Pattern Recognition and Machine Learning".

The background of the slide is a dark gray to black gradient, overlaid with a pattern of numerous overlapping circles. These circles vary in size and opacity, creating a bokeh or bubble effect. Some circles are solid dark gray, while others are lighter, appearing as if they are glowing or have a light source behind them. The overall effect is abstract and modern.

Notebook ^

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 \Leftarrow 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。**

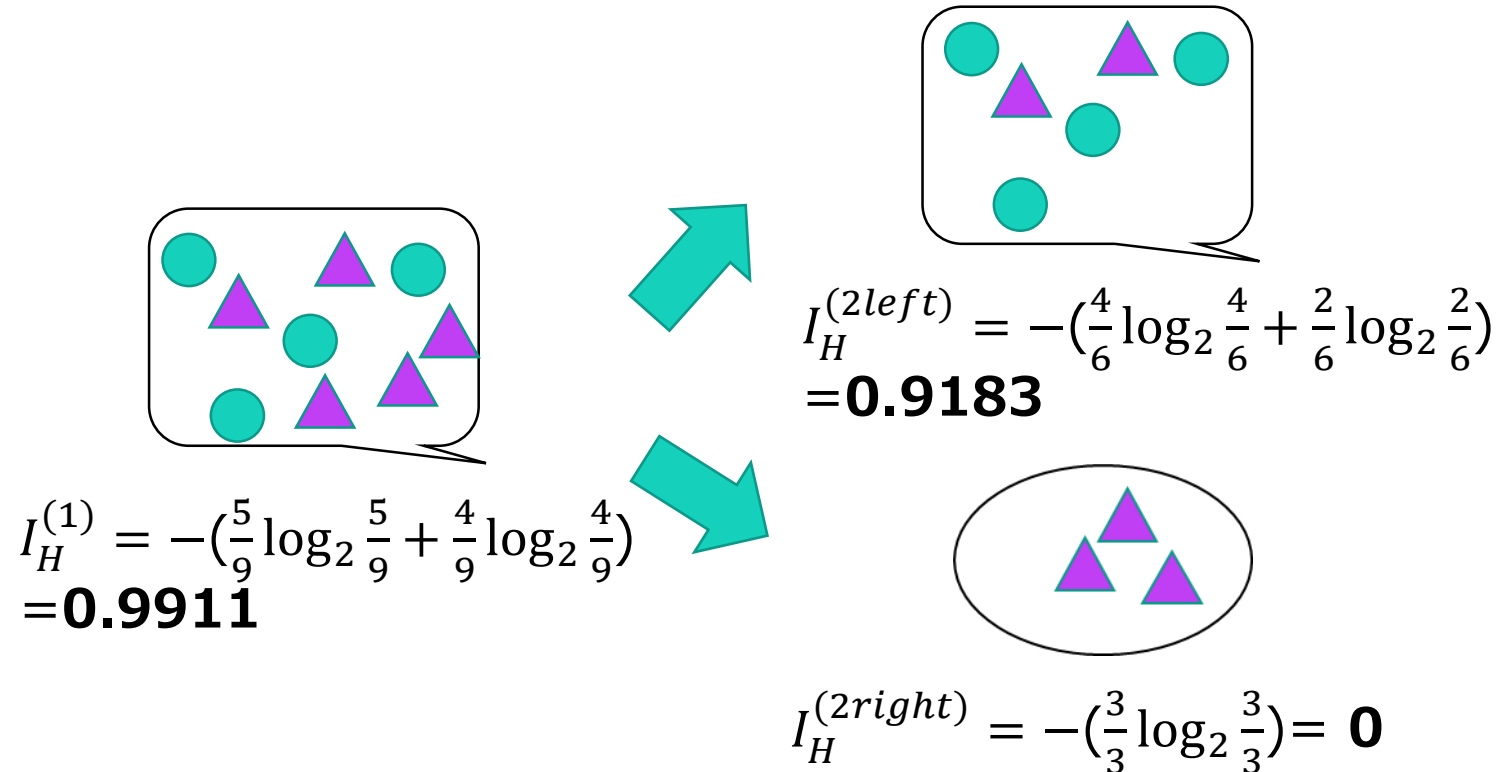


不純度の計算

【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 : $I_E = 1 - \max_{i=1, \dots, C} (p_i)$

【計算例】



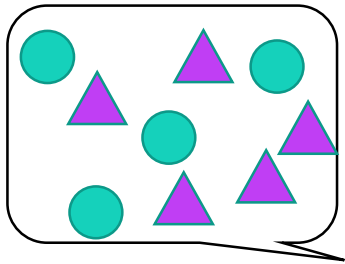
情報利得の計算

【情報利得】

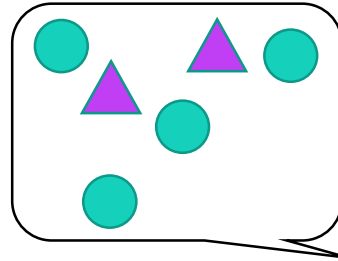
- 2分決定木の場合

- $$IG = I(p) - \frac{N_{left}}{N_p} I(left) - \frac{N_{right}}{N_p} I(right)$$

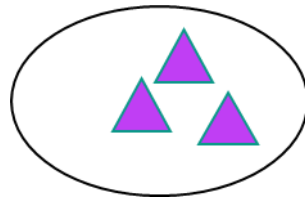
【計算例】



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



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



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

左図より

$$\begin{aligned} IG &= I_H^{(1)} - \frac{N_{left}}{N_p} I_H^{(2left)} - \frac{N_{right}}{N_p} I_H^{(2right)} \\ &= 0.9911 - \frac{6}{9} \times 0.9183 - \frac{3}{9} \times 0 \\ &= \mathbf{0.3789} \end{aligned}$$

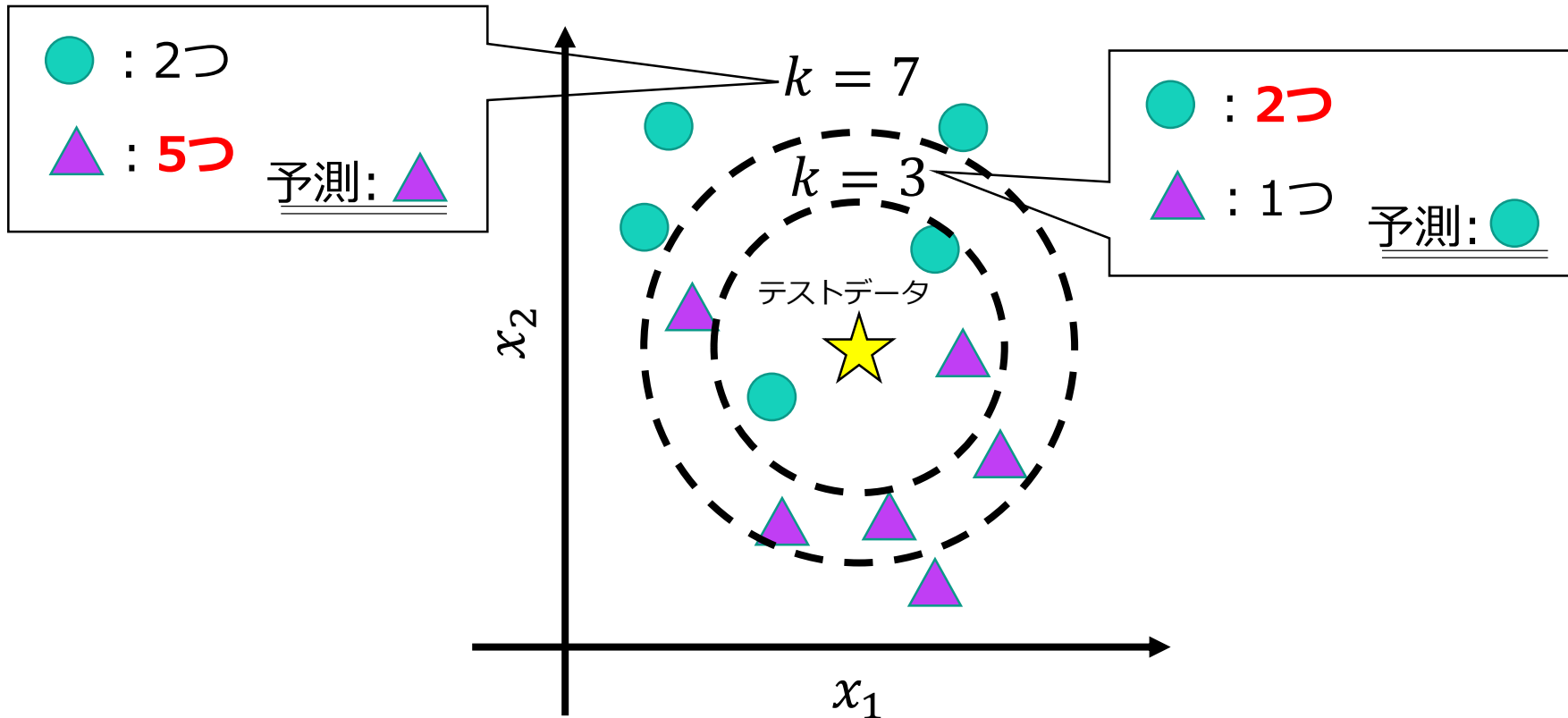
この情報利得を最大化する
ルールを求める

The background of the slide is a dark gray or black, featuring a pattern of numerous overlapping circles in various shades of gray. These circles vary in size and opacity, creating a bokeh-like effect. Some circles are more prominent and brighter, while others are faint and blend into the background. The overall composition is abstract and modern.

Notebook ^

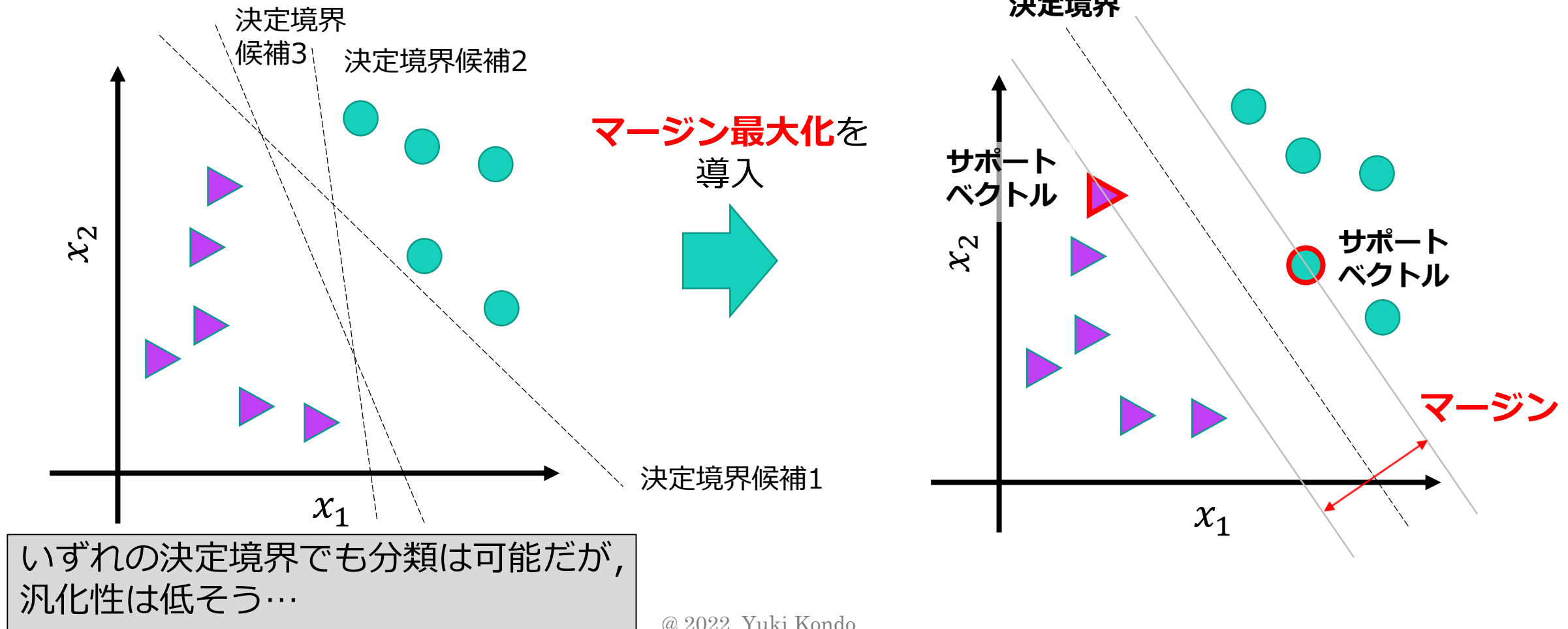
k-NN (k近傍法)

- 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 ^

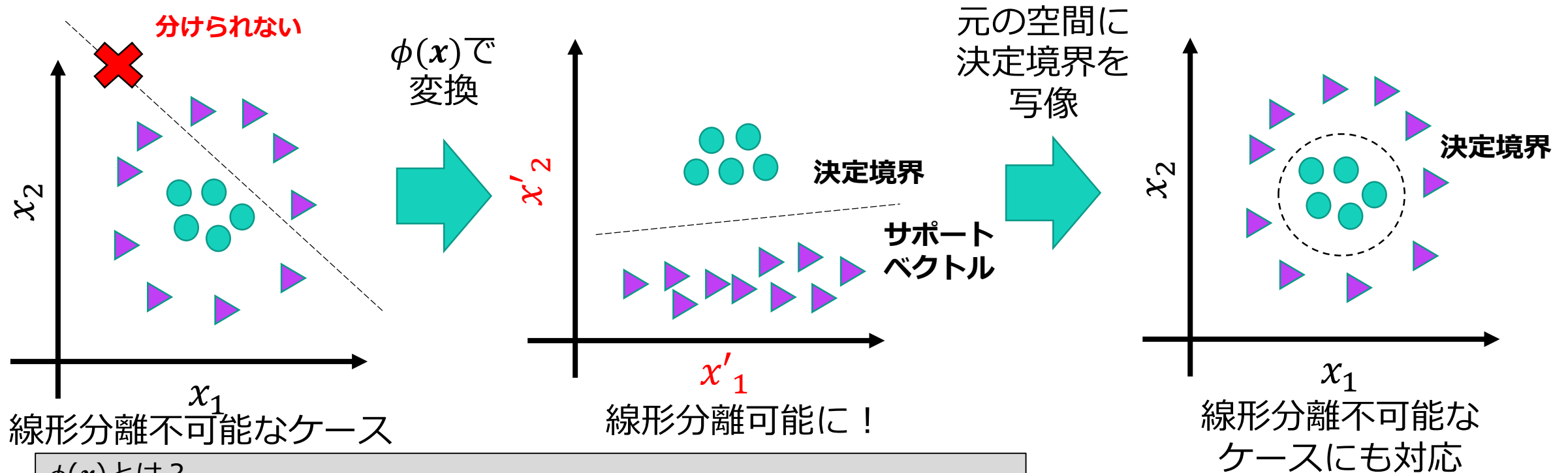
- 分類問題・回帰問題のいずれでも利用可能 (以下では分類で説明)
- クラス間のマージンを最大化する条件下で決定境界を定める。



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。

The background of the slide is a dark gray to black gradient, overlaid with a pattern of numerous overlapping circles. These circles vary in size and opacity, creating a bokeh or bubble effect. Some circles are solid dark gray, while others are lighter, semi-transparent, and overlap each other, creating a sense of depth and movement. The circles are distributed across the entire frame, with a slight concentration of larger, more prominent circles on the right side.

Notebook ^