

# 機械学習



AVILEN

- ① 機械学習って何？
- ② モデル開発の流れ
- ③ アルゴリズム概論

1 機械学習って何？

2 モデル開発の流れ

3 アルゴリズム概論

## はじめに

Q

機械学習ってどういった技術なの？

## はじめに

Q

機械学習ってどういった技術なの？

A

コンピュータに知的処理を行わせるため、データからパターンを見つけ出す技術！

# 機械学習とは データからパターンを見つけ出す技術

## データ

	築年数 (年)	専有面積 [㎡]	駅からの距離 [m]	家賃 [円]
物件A	20	40	200	45,000
物件B	15	35	400	40,000
⋮	⋮	⋮	⋮	⋮
物件Y	8	45	500	60,000
物件Z	5	50	100	85,000



## パターン

築年数：●●年

専有面積：●●㎡

駅からの距離：●●●m

であれば、

家賃は●●,●●●円だと**推論**



# 機械学習で何ができる？

## ケース 1：仕入れのための売上予測

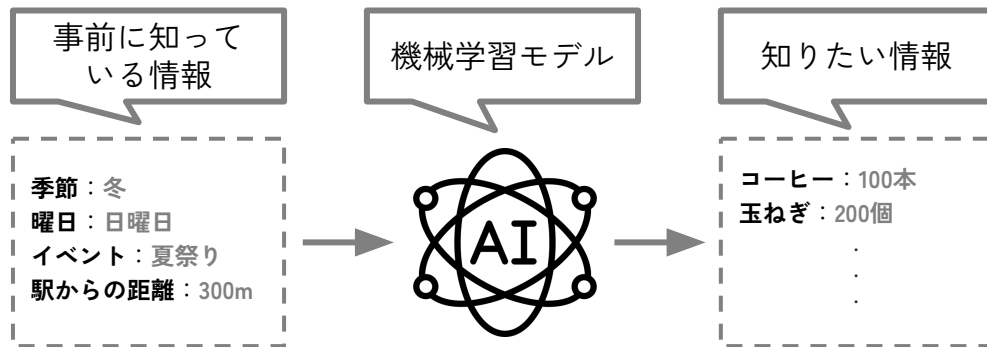
### 課題

あるスーパーでは、必要な仕入れ量を**ベテラン社員の勘と経験**で見積もっていた。

しかし属人性が高く、ベテラン社員がいないと正確に見積もりができない ...



### 機械学習の使いどころ



その日の情報や店舗の立地条件から**最適な仕入れ量を予測**  
→ **見積もり作業を機械化**することで属人性を排除！

# 機械学習で何ができる？

## ケース 2：工場製造における不良品検知

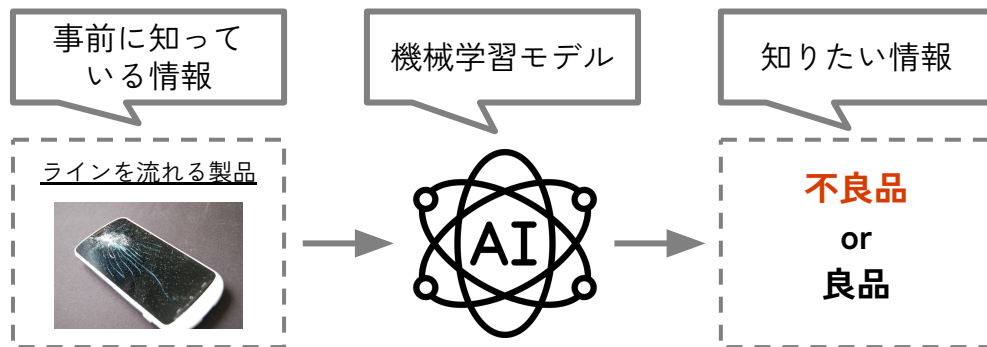
### 課題

ある工場では、製造ラインの不良品を**目視で検査**している。

しかし、単純作業のため従業員の負担が大きく、疲労によるミスも起きやすい ...



### 機械学習の使いどころ



製品の画像データを読み込ませ、**不良品を自動で検知**  
→ 従業員の**負担を軽減し、検出のムラを改善**！



# 説明変数から目的変数を入力する”関数”をつくる

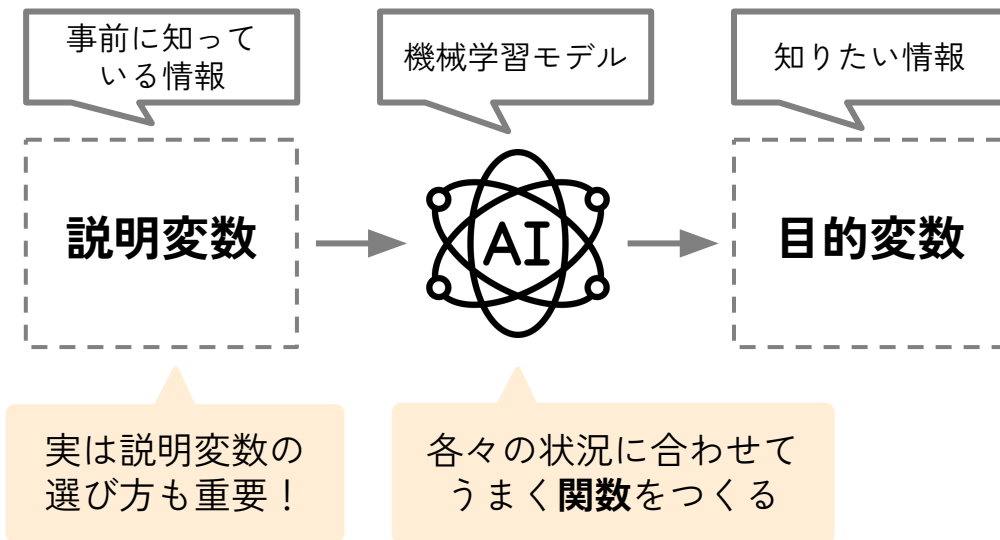
## 機械学習とは

持っている情報から知りたい情報を  
出力してくれるようにデータを使って  
モデルを学習させる方法

持っている情報＝**説明変数**

知りたい情報＝**目的変数**

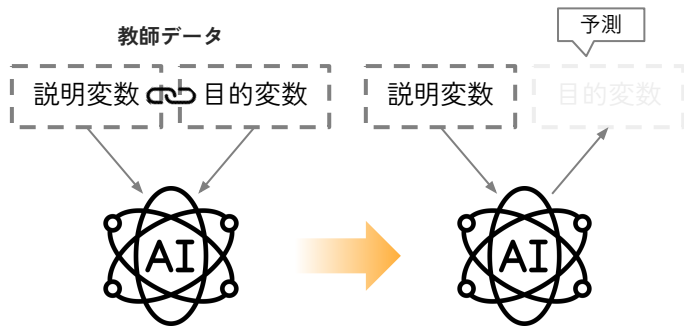
## 機械学習のしくみ



# 教師あり学習で解ける問題は大きく分けて2つ 「回帰」と「分類」

## 教師あり学習

説明変数と目的変数のペア  
(=**教師データ**) を大量に学習させて  
予測できるようにする方法



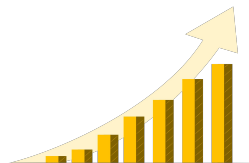
## 回帰と分類

教師あり学習で解ける問題

### 回帰

連続値を予測する問題

(例)売上金額の予測



### 分類

クラスを予測する問題

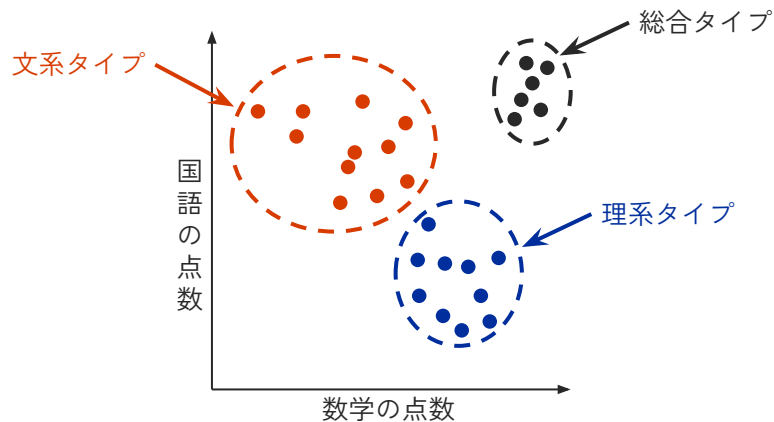
(例)不良品の分類



# 教師データを使わない学習方法もある

## 教師なし学習

教師データを与えずに  
データの特徴や構造を抽出する

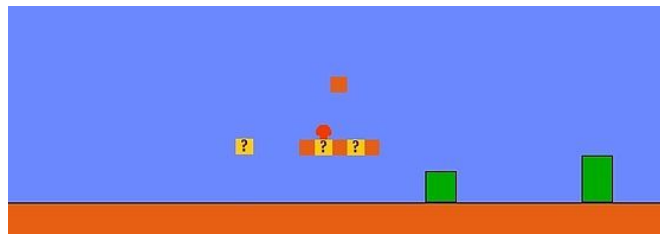


## 強化学習

明確な正解は与えないが、  
AIの判断結果に報酬を与えて学習させる

(報酬の例)

- ・ 敵を倒したら1点
- ・ 穴に落ちたら-5点



[https://commons.wikimedia.org/wiki/File:Super\\_Mario\\_Bros.\\_World\\_1-1.jpg](https://commons.wikimedia.org/wiki/File:Super_Mario_Bros._World_1-1.jpg)

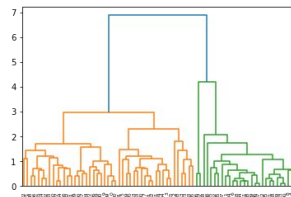
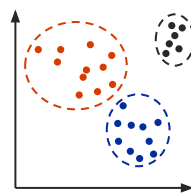
# データの種類や目的によって学習方法を選ぶ必要がある

## 教師あり学習の主な手法

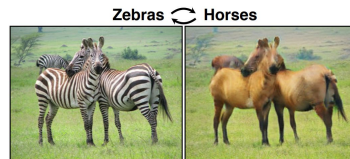


## 教師なし学習の主な手法

- k平均法(k-means)
- 階層型クラスタリング



- GAN(敵対的生成ネットワーク)



Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

## まとめ

01

**機械学習**とは、  
データからパターンを見つけ出す技術である

02

**教師あり学習**では**回帰・分類**を行えるが、  
問題の種類や状況に適した手法を選ぶ必要がある

① 機械学習って何？

② モデル開発の流れ

③ アルゴリズム概論

## はじめに

Q

モデルの開発はどのようにして行う？

## はじめに

Q

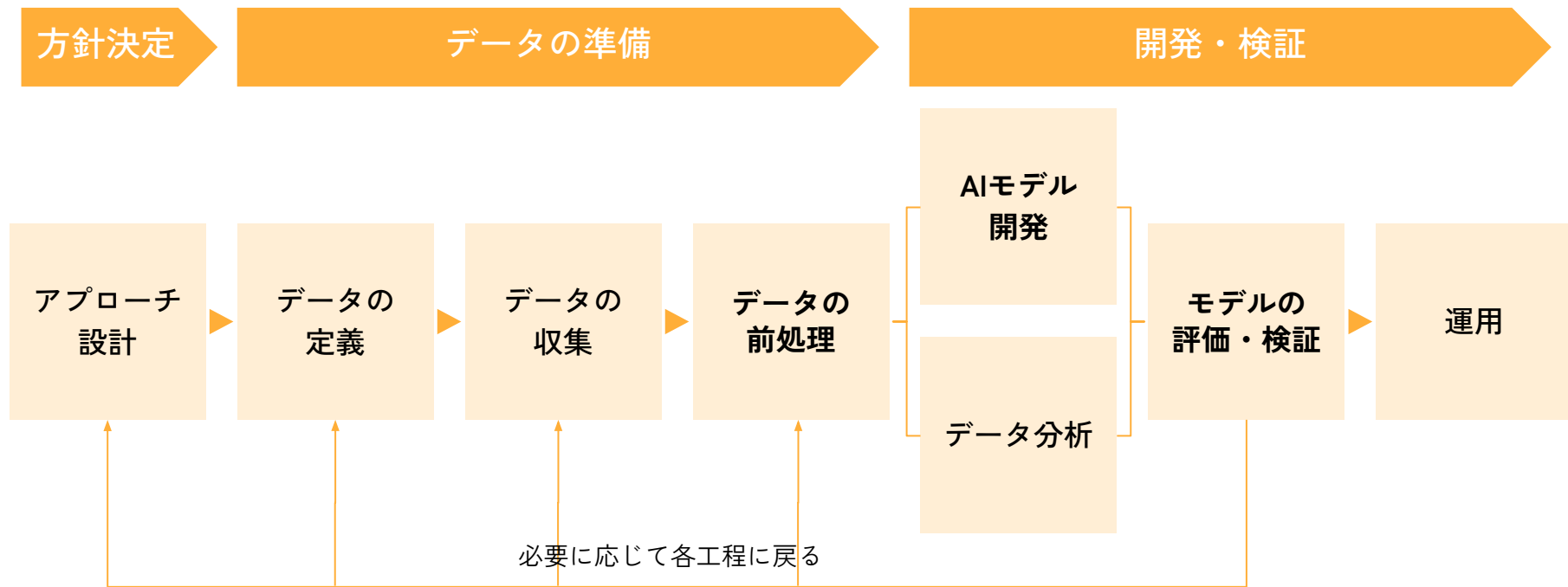
モデルの開発はどのようにして行う？

A

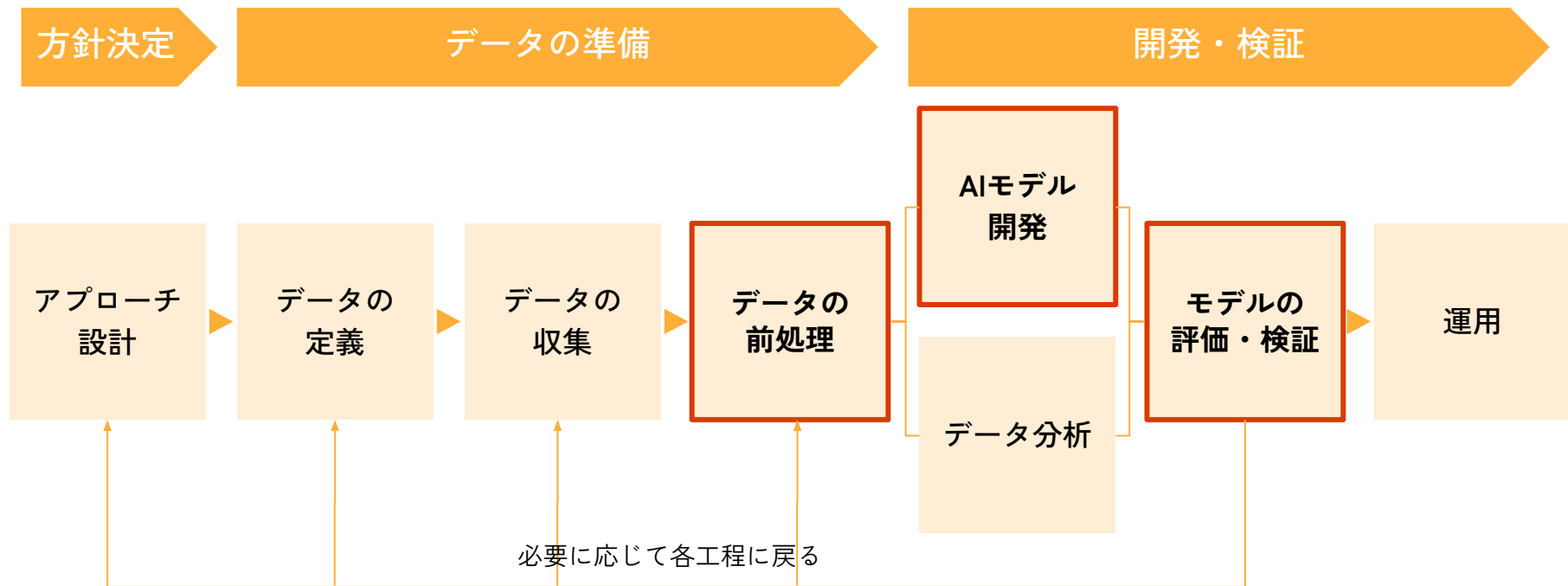
大まかな流れとして  
「前処理」「学習」「評価・検証」  
を行っていく！



# 機械学習プロジェクトの流れ



# 機械学習プロジェクトの流れ



# データの前処理

## 前処理の例

### 前処理とは...

データを加工してモデルに適した形に整えること

### 探索的データ分析(EDA, Exploratory Data Analysis)

を行って、データを深く理解することが重要

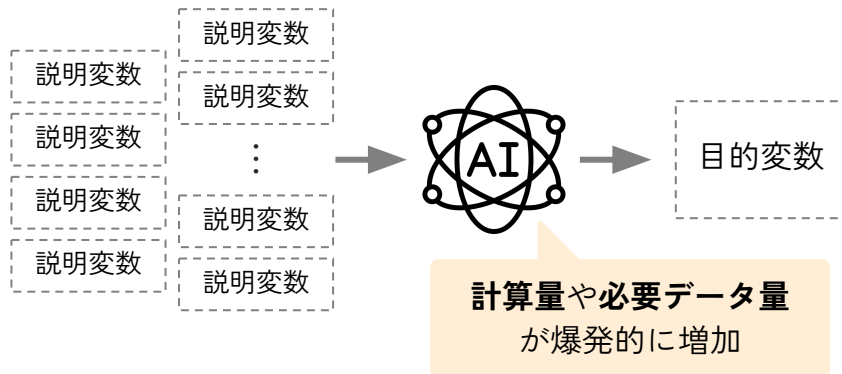
- ❑ **欠損値**がないか？
- ❑ 数値の**スケール**はどうか？
- ❑ **カテゴリ変数**は数値に変換されているか

etc...

## 説明変数の選択

一般的には、説明変数を多くすれば精度は上がる

しかし、**次元の呪い**という問題がある

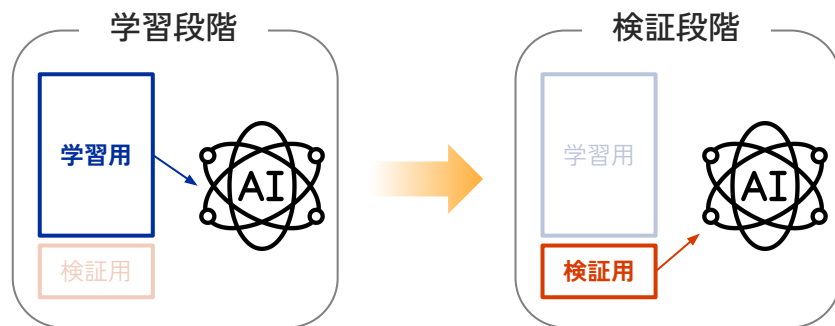


# 学習モデルの評価と検証

## 評価・検証の重要性

運用開始して「AIが全然使い物にならない！」と  
なってしまっては手遅れ

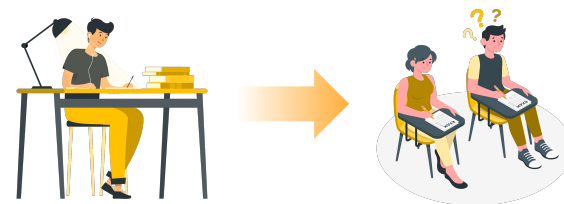
→ データの一部を検証用にとっておく



## 過学習

過去問に過剰適合すると、過去問では満点  
をとれるが、傾向が変わると解けなくなる

→これが機械学習でも起こる(過学習)



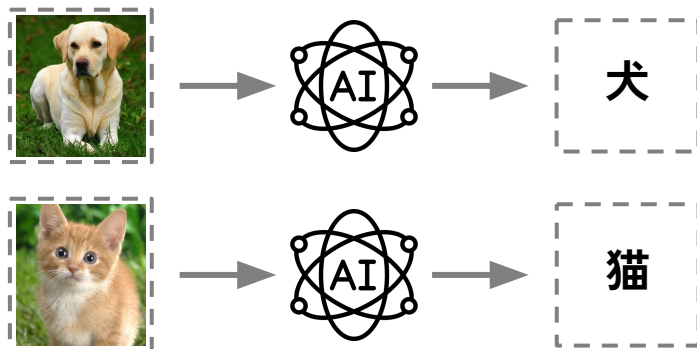
類題も解けるかどうかを確かめる

→ 機械学習では**検証用データ**で確かめる

# 分類モデルの評価方法①

## ～正解率(Accuracy)～

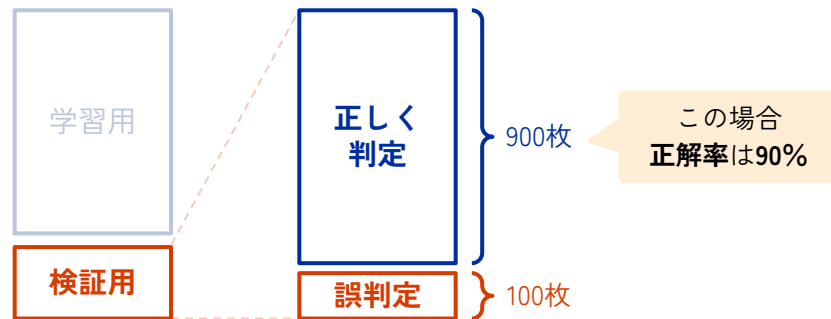
犬画像か猫画像かを判定するAI



この分類モデルの精度を  
どのように検証すればよいか？

正解率

最もシンプルな評価方法は  
**正解率**(Accuracy)を見ること

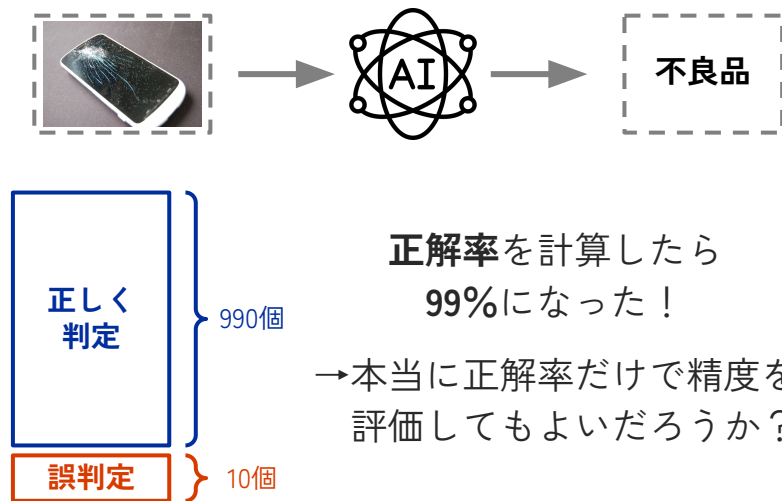


[1] DogBreedClassifier [https://github.com/srirammanikumar/DogBreedClassifier/blob/master/images/Labrador\\_retriever\\_06457.jpg](https://github.com/srirammanikumar/DogBreedClassifier/blob/master/images/Labrador_retriever_06457.jpg) から引用

[2] (OPTIONAL) EXPORTING A MODEL FROM PYTORCH TO ONNX AND RUNNING IT USING ONNX RUNTIME [https://pytorch.org/tutorials/advanced/super\\_resolution\\_with\\_onnxruntime.html](https://pytorch.org/tutorials/advanced/super_resolution_with_onnxruntime.html) から引用

## 分類モデルの評価方法② ～再現率(Recall)～

### 不良品を検出するAI



### 再現率

正解・不正解の内訳を詳しく見てみよう

		予測結果	
		不良品	良品
実際	不良品	0	10
	良品	0	990

不良品に対する  
正解率は0%...

全部「良品」と判定  
してしまっている...

不良品をどれだけ検出できるかが重要  
→不良品に対する正解率のことを  
**再現率**(Recall)という

# 混同行列(Confusion Matrix)とは

## 混同行列

前スライドで登場した、下のような表を  
**混同行列**(Confusion Matrix)という

		予測結果	
		陽性 (Positive)	陰性 (Negative)
実際	陽性 (Positive)	TP (True Positive)	FN (False Negative)
	陰性 (Negative)	FP (False Positive)	TN (True Negative)

## 陽性と陰性

どちらを陽性/陰性にするかは状況により異なる

(例) 不良品を検出するAI

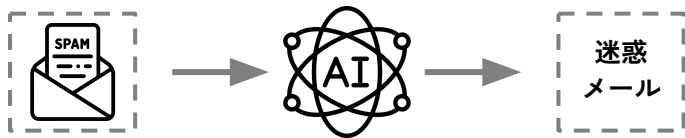
「不良品かどうか」が重要なので  
**不良品=陽性**(Positive)とする

(例) 迷惑メールを検出するAI

「迷惑メールかどうか」が重要なので  
**迷惑メール=陽性**(Positive)とする

## 分類モデルの評価方法③ ～適合率(Precision)～

### 迷惑メール判定AI



迷惑メールと判定されたものは自動的に  
迷惑メールフォルダへ振り分けられてしまう

→迷惑メールでないものを誤検出するのは避けたい

考えてみよう！

この例ではどんな指標で精度を評価すべきでしょうか？

### 適合率

		予測結果	
		迷惑メール	通常のメール
実際	迷惑メール	TP	FN
	通常のメール	FP	TN

迷惑メールと予測したメールのうちどれだけ正解したか？

陽性と予測したデータに対する正解率  
のことを**適合率**(Precision)という



## 分類モデルの評価方法④ ～F値～

### 迷惑メールの見逃しも減らしたい



普通のメールが  
迷惑メールフォルダ  
に入るのは嫌...

適合率(Precision)を  
高くしたい



迷惑メールを  
きちんと検出して  
くれないのも嫌...

再現率(Recall)を  
高くしたい

しかし...両者の間にはトレードオフの関係

Precision



適合率を高くすると  
再現率が低くなる

Precision Recall



バランスが良い

Recall



再現率を高くすると  
適合率が低くなる

### F値

適合率と再現率のバランスを重視した  
**F値**という評価指標がある

$$F値 = \frac{2 \times (\text{適合率}) \times (\text{再現率})}{(\text{適合率}) + (\text{再現率})}$$

適合率=0.1, 再現率=0.8

$$F = (2 \times 0.1 \times 0.8) / (0.1 + 0.8) = \underline{0.18}$$

適合率=0.5, 再現率=0.5

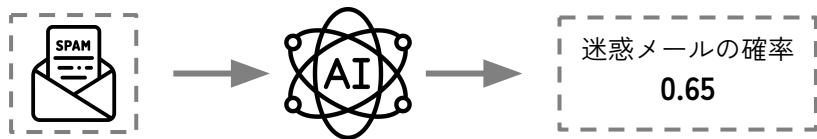
$$F = (2 \times 0.5 \times 0.5) / (0.5 + 0.5) = \underline{0.5}$$

数学的には、F値は適合率と再現率の調和平均とも言える

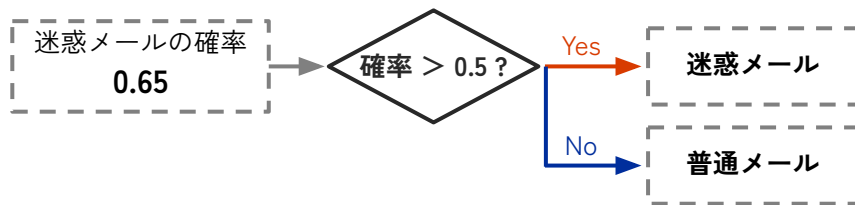
# 分類モデルの評価方法⑤

## ～ROC曲線とAUC～

### 分類モデルは確率を出力する



人が設定した**閾値**に基づき、迷惑メールかどうか判定

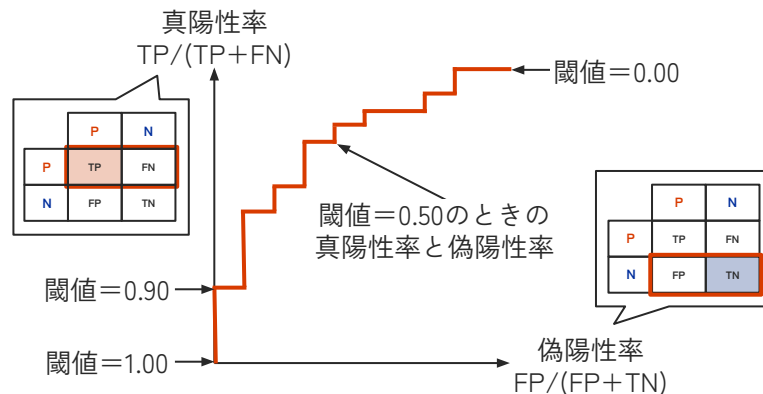


閾値によってモデルの精度が変わってしまう  
→閾値の影響を受けずに精度評価したい！

### ROC曲線

**ROC曲線**とは...

分類の閾値を変えていったときの  
真陽性率と偽陽性率の関係をプロットしたもの

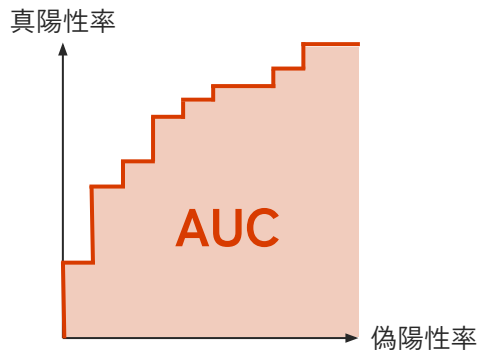


## 分類モデルの評価方法⑤

### ～ROC曲線とAUC～

#### AUC

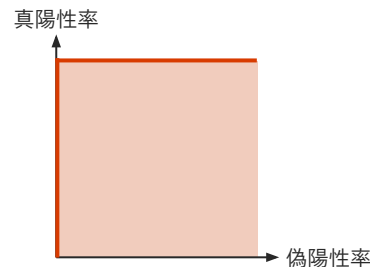
ROC曲線の下面積のことを  
**AUC**(Area Under the ROC Curve)という



#### AUC値のめやす

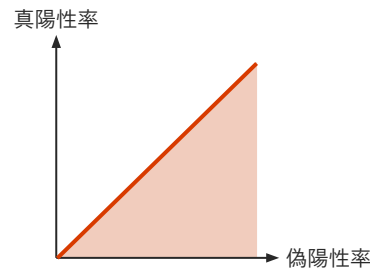
AUC=1.0の場合

完璧に分類できている



AUC=0.5の場合

コイン投げで  
予測しているのと同じ



# 分類モデルの評価方法のまとめ

## 混同行列

Accuracy

**正解率**：検証データに対して何%正解したか

Recall  
**再現率**

陽性のデータに対して  
何%正解したか

Precision  
**適合率**

陽性と予測したデータに対  
して何%正解したか

		予測結果	
		Positive	Negative
実際	Positive	TP	FN
	Negative	FP	TN

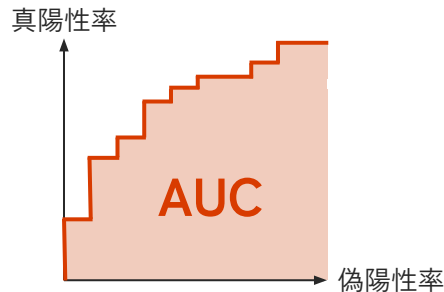
		予測結果	
		Positive	Negative
実際	Positive	TP	FN
	Negative	FP	TN

## F値・AUC

**F値**：再現率と適合率のバランスを表す

$$F値 = 2 \cdot \frac{(適合率) \times (再現率)}{(適合率) + (再現率)}$$

**AUC**：ROC曲線の下面積



## 回帰モデルの評価例①

### ～MAEとMSE・RMSE～

#### MAE

**MAE**(Mean Absolute Error)

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

$\hat{y}_i$  : 予測値  
 $y_i$  : 正解値

#### メリット

- ❑ 人間にとって**解釈しやすい**
- ❑ **外れ値**の影響を受けにくい

#### MSE・RMSE

**MSE**(Mean Squared Error)

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

- ❑ 誤差が大きいほど過大に評価

二乗しているため単位が変わり、解釈しづらい...  
→ルートをとって元の単位に戻す

**RMSE**(Root Mean Squared Error)

$$\text{RMSE} = \sqrt{\text{MSE}}$$

## 回帰モデルの評価例② ～RMSLEとMAPE～

### RMSLE

**RMSLE**(Root Mean Squared Logarithmic Error)

$$\text{RMSLE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \{\log(1 + \hat{y}_i) - \log(1 + y_i)\}^2}$$

1を足しているのは「log 0」となるのを防ぐため

#### メリット

- 下振れの誤差を過大評価したいときに有用

#### デメリット

- ❖ 予測値や正解値に負の数があると使えない

### MAPE

**MAPE**(Mean Absolute Percentage Error)

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

#### メリット

- スケールが異なるデータに対応できる

#### デメリット

- ❖ 正解値に0があると使えない
- ❖ 正解値が0に近いと値が大きくなりやすい

## まとめ

01

学習モデルを構築する前に、  
**前処理**や説明変数の選択を行っておくことが重要

02

分類モデルの評価指標には  
**正解率・再現率・適合率・F値・AUC**などがある

03

回帰モデルの評価指標には  
**MAE・RMSE・RMSLE・MAPE**などがある

1 機械学習って何？

2 モデル開発の流れ

3 アルゴリズム概論



## はじめに

Q

機械学習ではどのようにして  
学習・予測を行っているの？

## はじめに

Q

機械学習ではどのようにして  
学習・予測を行っているの？

A

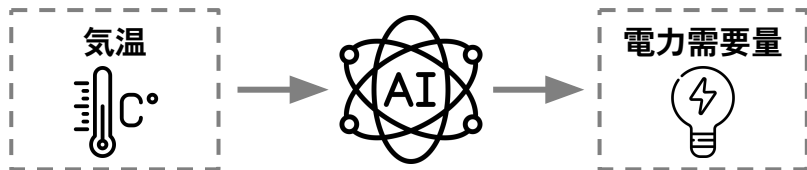
これから代表的な機械学習の  
アルゴリズムを紹介していく！

## 回帰タスクのアルゴリズム①

### ～単回帰分析～

#### 気温から電力需要量を予測したい

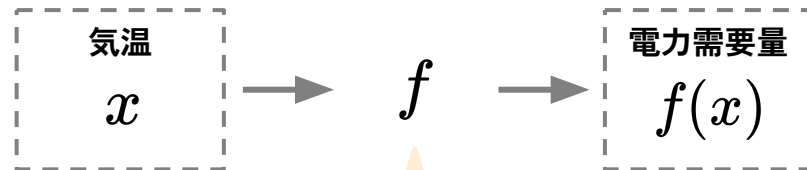
ある日の予想される**平均気温**から  
その日の**電力需要量**を予測したい



どんな関数にするか？

#### 単回帰分析

**単回帰分析**では気温と電力需要量  
の関係が**1次関数**であると仮定する



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

係数a,bはデータから学習して求められる

## 回帰タスクのアルゴリズム② ～重回帰分析～

### 単回帰分析のメリットとデメリット

#### メリット

- ❑ モデルがシンプルで解釈がしやすい

#### デメリット

- ❖ 1つの説明変数しか考慮しないため低精度

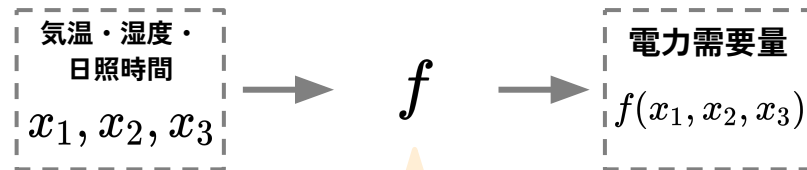
湿度や日照時間も  
電力需要に関係しよう！



重回帰分析

### 重回帰分析

**重回帰分析**では複数の説明変数から  
目的変数の値を予測することができる



$$f(x_1, x_2, x_3) = w_0 + w_1x_1 + w_2x_2 + w_3x_3$$

係数はデータから学習して求められる

# 重回帰分析で説明変数の影響度を知る

## 重回帰分析のメリット

### メリット

- ❑ 複数の説明変数を考慮することができ、単回帰分析よりも高精度に予測が可能
- ❑ 係数の大小から説明変数の影響度がわかる

### デメリット

- ❖ 説明変数が正規分布に従っていない場合や変数間に強い相関がある(=多重共線性)場合、適切に予測できないことがある

## 重回帰分析の係数の意味

重回帰分析の結果が下のようになったとする

$$f(x_1, x_2, x_3) = -280x_1 + 5.39x_2 - 50.0x_3$$

$x_1$ : 気温  
 $x_2$ : 湿度  
 $x_3$ : 日照時間

↑  
気温が下がると  
電力需要は増える

↑  
湿度が上がると  
電力需要もやや増える

↑  
日照時間の影響は  
気温に比べ小さい

係数を見ることで、説明変数が目的変数に  
**正負**どちらの影響を**どれだけ**与えるかがわかる

# 機械学習でタイタニックの生存予測

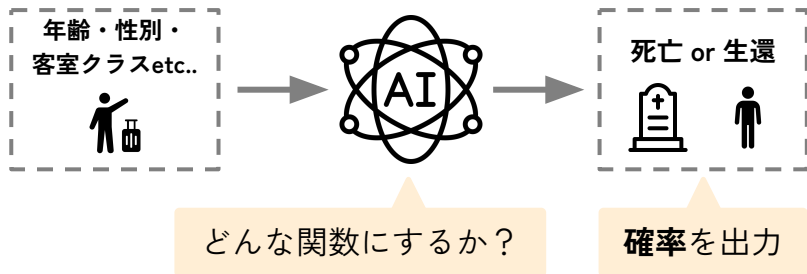
## タイタニックの生存予測



### タイタニック号沈没事故(1912)

この事故で1,514人が死亡、710人生還

→乗員乗客に関するデータを使い生存予測を行う



## 重回帰分析ではうまくいかない

重回帰分析の結果が下のようになったとする

$$f(x_1, x_2, x_3) = 0.2x_1 + 0.9x_2 - 0.3x_3$$

例えば  $x_1 = 3.0, x_2 = 0.8, x_3 = 0.1$  とすると  
1.29 となり、**確率にならない!**

→カテゴリ変数では重回帰分析できない

# 分類タスクのアルゴリズム①

## ～ロジスティック回帰～

### ロジスティック回帰

**ロジスティック回帰**とは...

重回帰分析の左辺を**ロジット**にしたもの

$$\underbrace{\log\left(\frac{p}{1-p}\right)}_{\text{ロジット}} = w_0 + w_1x_1 + w_2x_2 + \dots$$

$p$ : 目的変数が1である確率

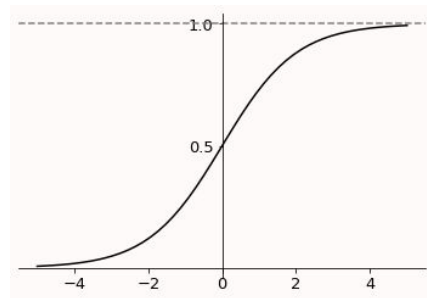
今知りたいのは確率なので、 $p$ について解くと

$$p = \frac{1}{1 + e^{-(w_0 + w_1x_1 + w_2x_2 + \dots)}}$$

### ロジスティック関数

ロジスティック回帰では  
必ず0～1に収まるようになっている

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



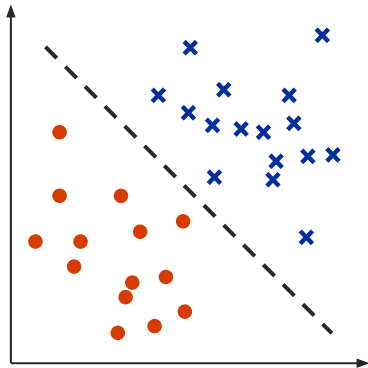
## 分類タスクのアルゴリズム②

### ～SVM～

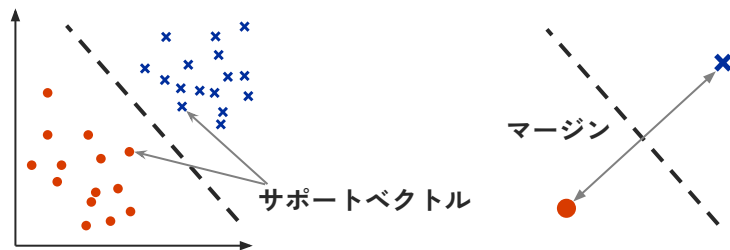
#### SVM

**SVM**(Support Vector Machine)とは...

データごとの説明変数の分布図に  
直線(平面)を引いて分類を行うアルゴリズム



#### 線形分離可能な場合



**サポートベクトル**：直線に最も近いデータ点

**マージン**：サポートベクトルと直線との距離

→ マージンが最大となるような直線を引く

➡ **ハードマージン**

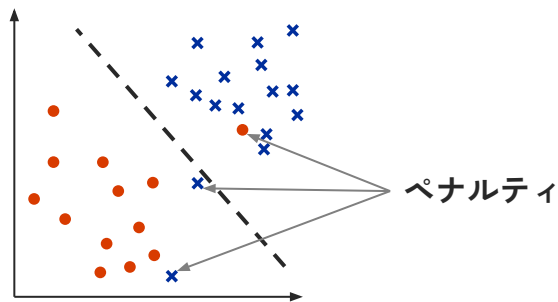


## 分類タスクのアルゴリズム②

### ～SVM～

#### 線形分離不可能な場合

直線(平面)を超えてしまってもOKとする  
ただし、その場合は**ペナルティ**を与える

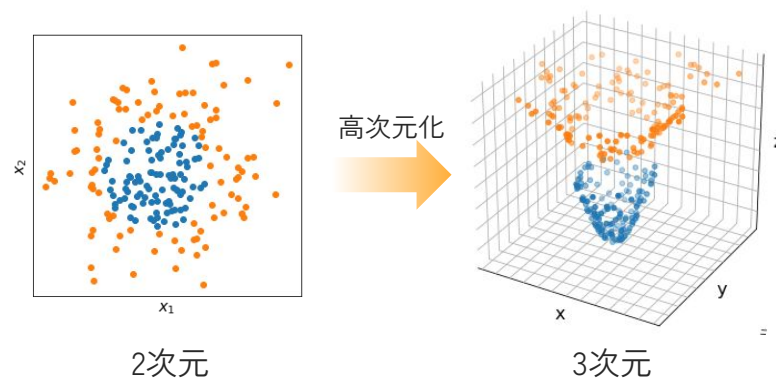


→ ソフトマージン

#### カーネル法

**カーネル法**とは...

高次元なデータに変換することで  
うまく分離できるようにする手法



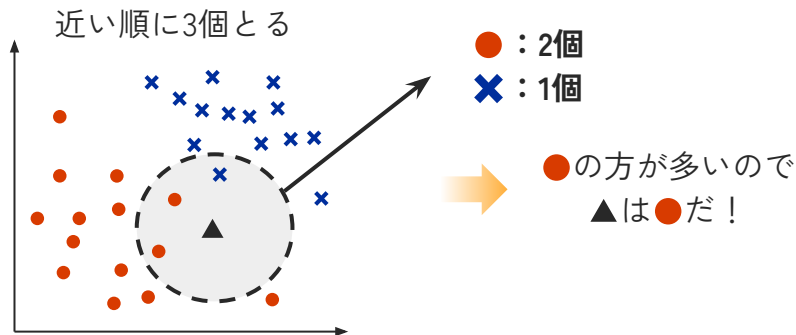
## 分類タスクのアルゴリズム③

### ～k-NN法～

#### k-NN法

**k-NN法**(k-Nearest Neighbors)とは...

最も近いデータ k 個の多数決で  
クラスを決定するアルゴリズム



#### メリットとデメリット

##### メリット

- ❑ 学習を行う必要がない
- ❑ シンプルなので様々な問題に適用しやすい

##### デメリット

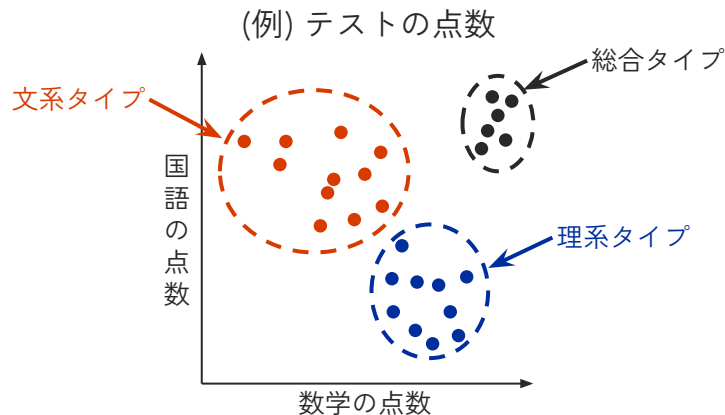
- ❖ データ量が多いと計算時間がかかる
- ❖ 次元の高いデータに弱い(次元の呪い)

次元が高いと、データ間の距離を測定する際に  
どのデータとの距離も同程度になってしまう  
という問題が生じやすくなる

# 教師なし学習のアルゴリズム ～k-means法～

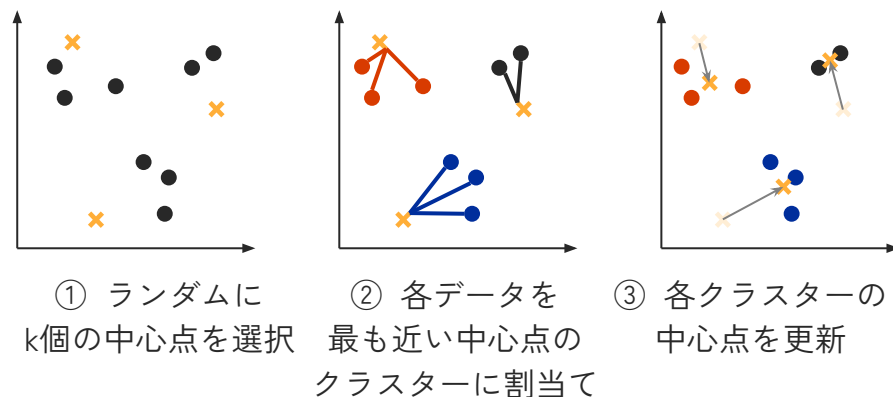
## クラスター分析

**クラスター分析**(クラスタリング)とは...  
データを似たもの同士でグループ分けすること



## k-means法

**k-means法**とは...  
平均を用いてk個のクラスターに分類する手法



## k-means法のメリットとデメリット

### メリット・デメリット

#### メリット

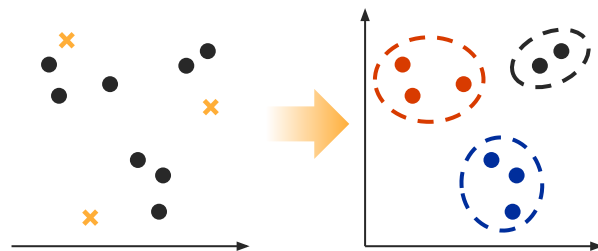
- 計算量が比較的小さく済む

#### デメリット

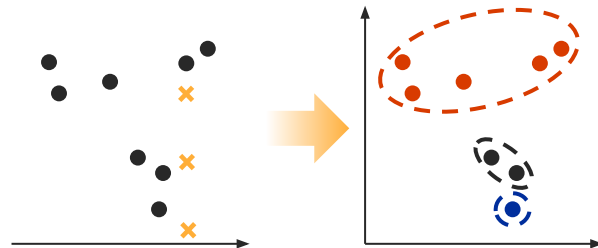
- ❖ 初期値によって結果が大きく変わりやすい

### 初期値依存性

“良い”初期値  
の場合



“悪い”初期値  
の場合



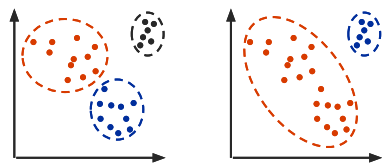
# 機械学習の”設定” ハイパーパラメータ

## ハイパーパラメータ

**ハイパーパラメータ**とは...

機械学習のアルゴリズムでは最適化できず、  
手動で設定する必要があるパラメータ

(例) k-means法



クラスター数は  
人間が決める

(例) SVM

$$\frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^N \xi_i$$

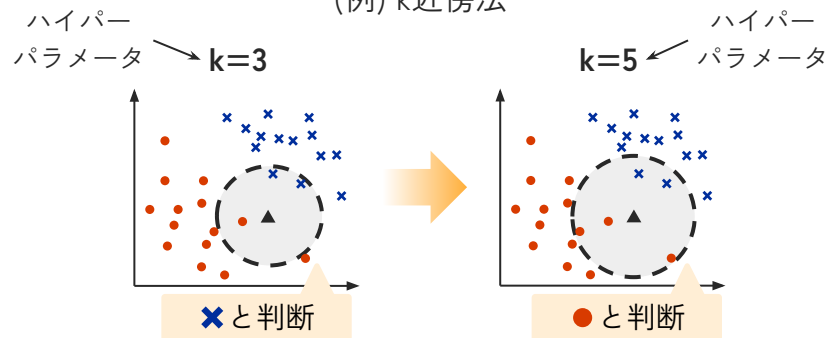
どれだけ誤分類を許すかは  
人間が決める

## チューニング

**ハイパーパラメータチューニング**とは...

適切なハイパーパラメータを探して  
機械学習モデルの精度を上げること

(例) k近傍法



# 2種類以上のハイパーパラメータを チューニングする方法

## グリッドサーチ

### グリッドサーチとは...

各パラメータの候補を列挙して全組合せを試し  
最も精度が高いものを探し出す方法


パラメータ①

		1	2	3	4	5
パラメータ②	0.5	90%	92%	93%	94%	93%
	1.5	91%	92%	92%	95%	91%
	2.5	89%	90%	93%	90%	92%
	3.5	90%	95%	97%	92%	94%
	4.5	92%	92%	94%	93%	90%

## ランダムサーチ

### ランダムサーチとは...

全組合せを試すのではなく、  
ある確率分布に従ってランダムに探索する方法



		1	2	3	4	5
パラメータ②	0.5			93%	94%	
	1.5		92%	92%	95%	91%
	2.5			93%		
	3.5	90%	95%	97%	92%	94%
	4.5		92%	94%	93%	

## まとめ

01

回帰タスクのアルゴリズムには  
**単回帰分析・重回帰分析**などがある

02

分類タスクのアルゴリズムには  
**ロジスティック回帰・SVM・k-NN法**などがある

03

モデルの**ハイパーパラメータ**を調整することで  
精度を向上させることができる