

# データ解析に関する講義

2024年7月23日(火)

埼玉大学 工学部

助教 杉浦 陽介

# 目次

## 1. データ解析概要

- 概要
- 解析ツールの紹介
- Pythonの紹介

## 2. 機械学習の基礎

- 回帰分析の基本：線形回帰・非線形回帰
- 識別分析の基本：ランダムフォレスト
- データの整形

## 3. 深層学習と生成AIの活用

- 深層学習の仕組み
- 生成AIとは
- 生成AIを利用したAI開発

1

---

# データ解析概要

スマホやセンサ技術の普及で,大量のデータの解析が重要となる.

## データ解析で重要なこと

1. 数学的な知識/専門的なプログラミングスキルは不要

⇒ **Python** を使えば簡単にデータ解析を始められる！  
(ただし最低限のプログラミングスキルは必要)

2. 解析の目的を明確にすること

⇒ 目的に応じて適切なアプローチも異なる.

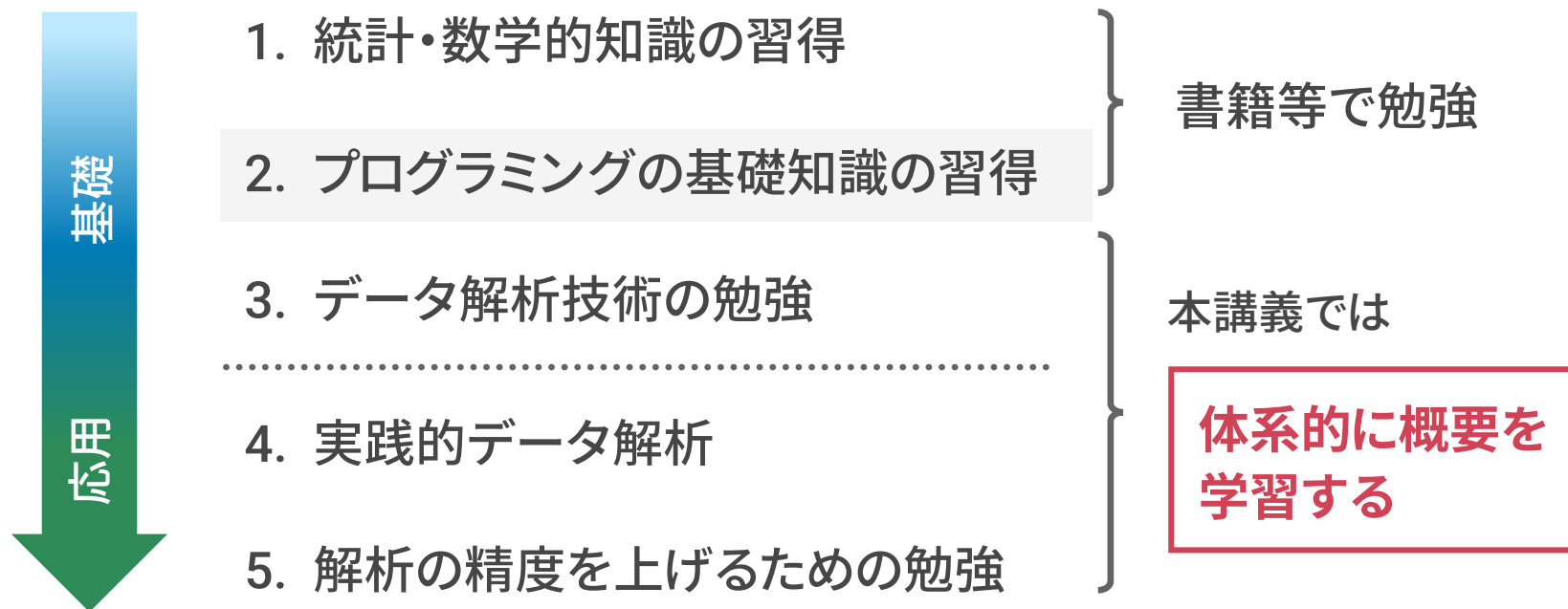
3. データを収集し,機械が処理しやすいよう整形・加工すること

⇒ 不要なデータを見極め,除去・加工する技術が求められる.

**実践的なデータ解析を行い,勘所を鍛えよう**

# この講義の立ち位置

データ解析を“本格的に”勉強するには...



実践的にデータ解析を行うにはプログラミングの基礎知識が必須

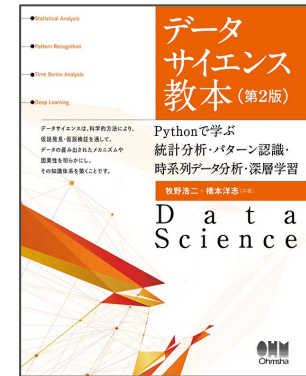
**データ解析ツールが“ひとまず使える”ところを目指す**

# データ解析のおすすめ教材

## データサイエンス教本

著：牧野浩二・橋本 洋志（オーム社）

統計の基礎知識と実践的な解析技術を網羅。  
原理は割愛されており, サンプルコードから  
解析技術の効果を直感的に理解できる。高校数学基礎レベル。



## データマイニング演習 (Github)

<https://github.com/Yosuke-Sugiura/datamining-exercise/tree/master>

埼玉大学の授業で使用する教材。機械学習に関わる知識を丁寧に解説。  
文系でも理解できるレベル。ただし演習問題の難易度は高い。

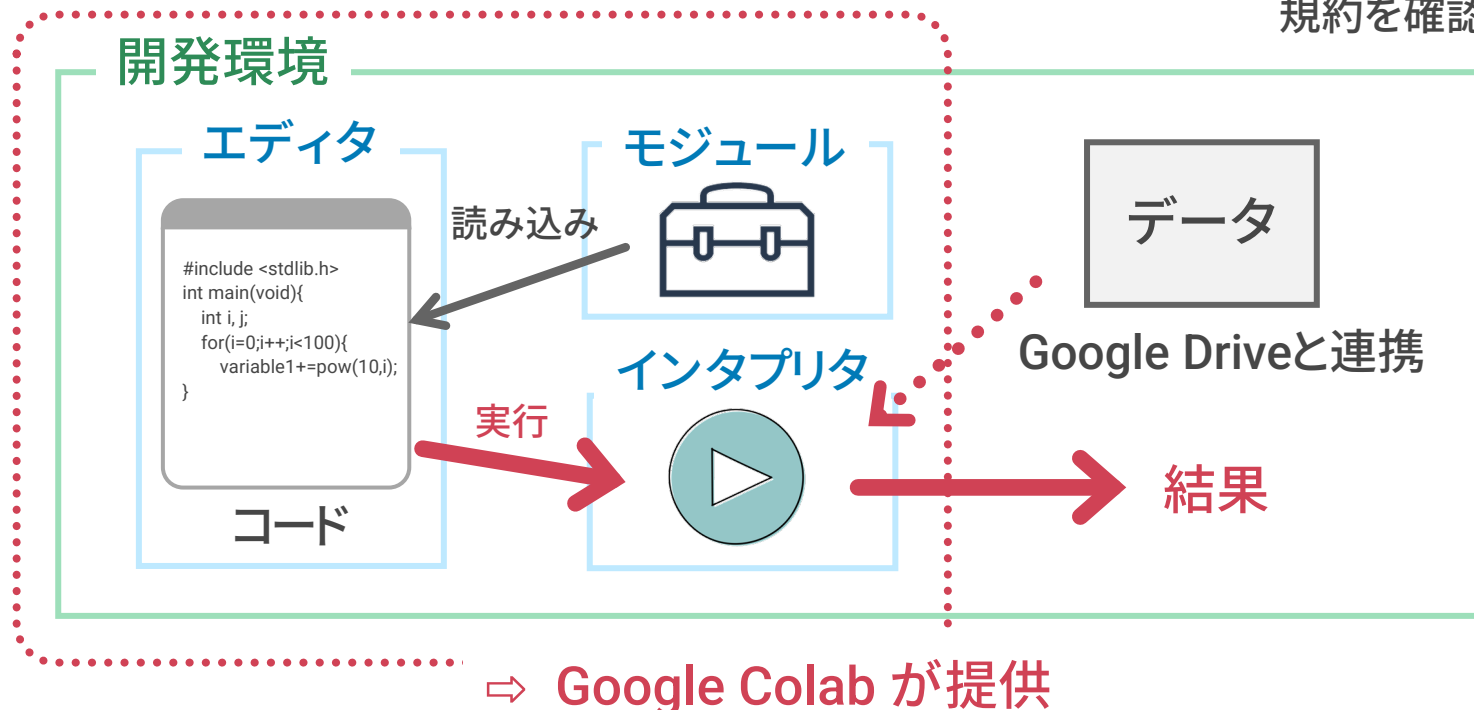
**本日はこれらの教材 + 深層学習を解説**

# 解析ツールについて

データ解析には簡単なプログラミングをする必要がある。  
プログラミングを実行する環境として **Google Colab** を利用する。

モジュールのインストール等の面倒な環境構築が全く不要。  
また無料で**高性能な GPU** を利用できる\*。

\* 一部機能に制限あり。  
規約を確認のこと。



Google ColabではPythonという言葉を利用してデータ解析を行う。

## 1. 豊富なモジュール(ライブラリ)

- 公式/サードパーティ問わず多数のモジュールが公開.
- 強力なデータ解析用モジュールが用意されている.  
⇒ 短い簡単なコードで高度なデータ解析を実現できる!

## 2. デバッグのしやすさ

- 実行時にエラーが出た場合,エラーの原因を表示してくれる.

## 3. 高い普及率

- Pythonプログラミングコミュニティが発展している.  
⇒ ソースコードのリソース, FAQが充実!



# Pythonを習得するには

Python によるプログラミング技術の習得には以下の手順がおすすめ。

## Step 1. 基本的なプログラミング技術の習得

プログラミング全般に共通する”様式”を最低限学ぶ必要がある。

1. 配列・変数の使い方
2. Print文・If文・For文の使い方
3. 関数の使い方

おすすめ教材:

[https://github.com/YosukeSugiura/Introduction\\_to\\_Programming](https://github.com/YosukeSugiura/Introduction_to_Programming)

## Step 2. サンプルコード・既存コードの改修による実践

既存コードを動かしてみて, わからないところは調べて理解する。

プログラミングのすべてを知らないと開発できないわけではない  
知りたいところだけを重点的に調べて学習してくとよい

実際にデータ解析を行う際には、一からコードを組むことは少ない。  
先人たちによって**開発されたコード**を利用，改修することが多い。

## GitHub

- 自分でサーバを用意する必要がなく，公開も利用も無料！
- ソースコードは (基本的に) 全世界に公開される。  
⇒ 最新技術(ソースコード)の多くが GitHub で公開されている
- 利用規約(特に商用利用に関して)があるものも．要チェック．

他，Qiita や Zenn などの個人の技術ブログも大変参考になる。

**プログラミングコミュニティの恩恵を享受しましょう！**

2

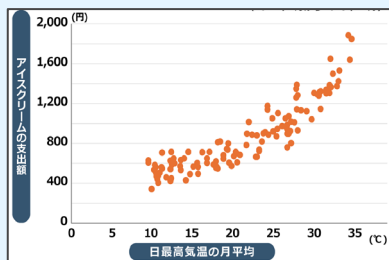
# 機械学習の基礎

データ解析技術は目的に応じて多岐にわたって存在する。  
ここでは機械学習の「**教師あり学習**」に焦点を絞って学習する。

## 教師あり学習

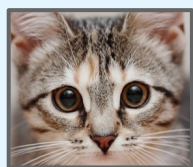
入力と正解(出力)の関係を学習

### 回帰



- 予測
- 異常検知

### 識別



画像



→ “猫”

- 認識

## 教師なし学習

入力から潜在的な特徴の共通項を学習

### クラスタリング



## 強化学習

環境適合する機械学習

教師あり学習は，入力とそれに対応する出力(真値)のペア(=**データセット**)を用いて入力と出力の間の関係を学習する。

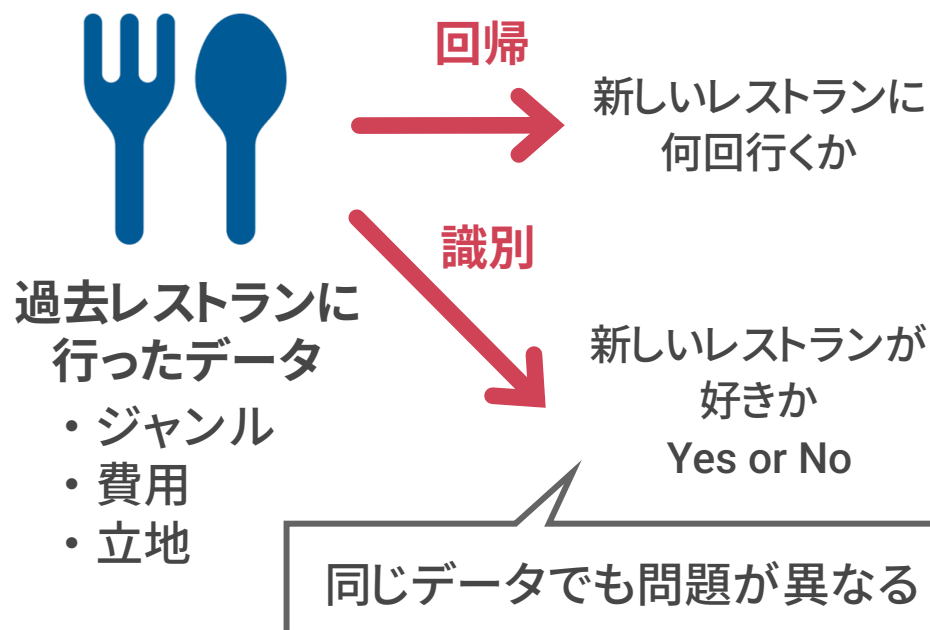
## 1. 回帰分析

出力=数値である問題．

## 2. 識別

出力=クラスである問題．

一般に回帰のほうが難しい



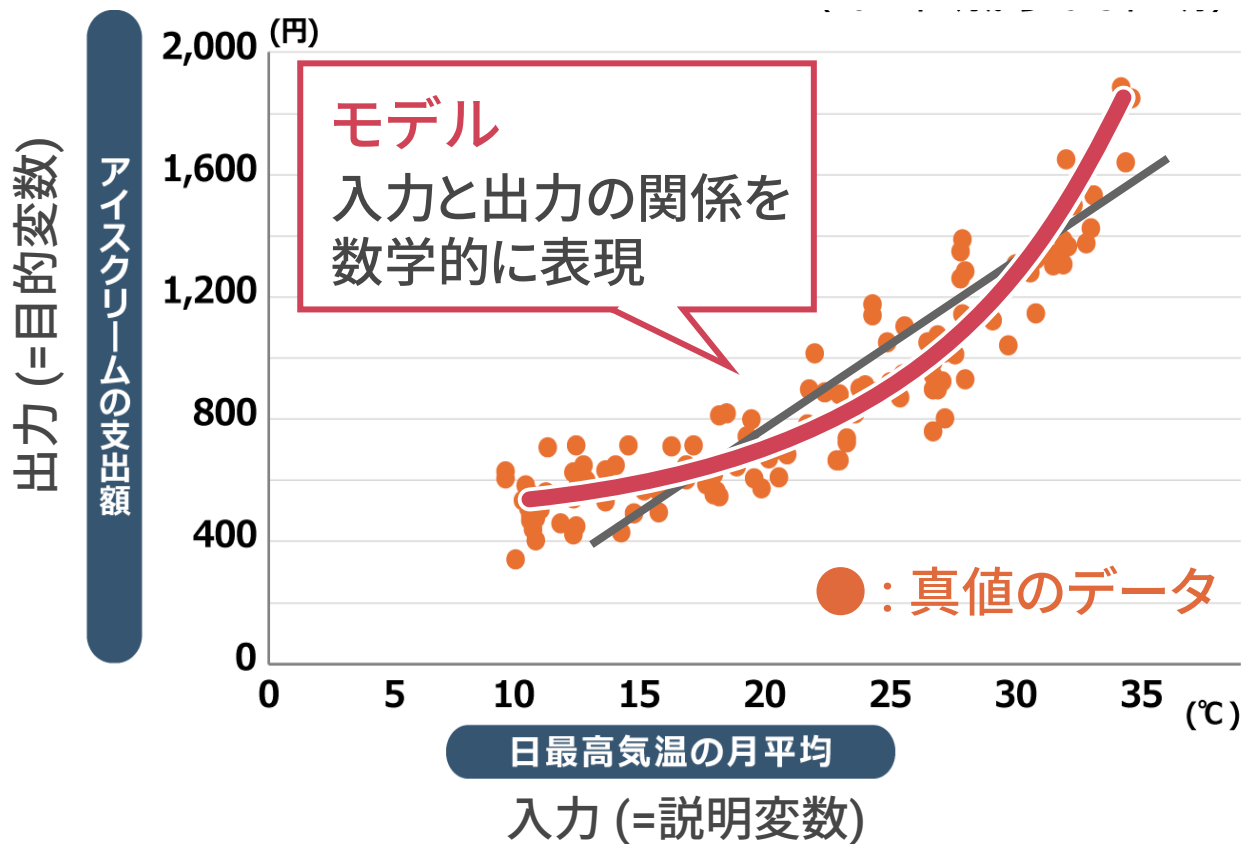
どういう問題(正解)を設定するか?がとても重要

2.1

# 機械学習の基礎

## 回帰分析

# 回帰分析の概要



未知のデータに対しても適切な出力を得るモデルを求めることが目標

データ解析者はより妥当なモデルを選択する必要がある

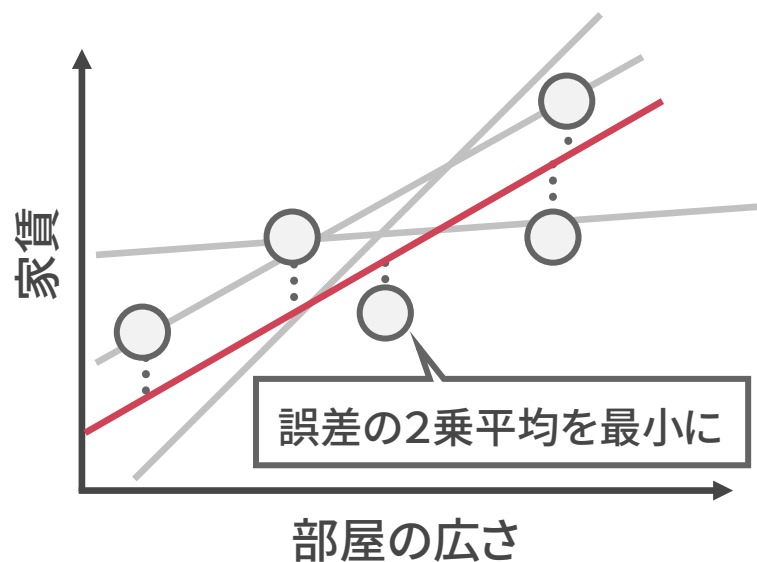
# 回帰分析の流れ

回帰問題は学習・推論の2ステップで行われる。

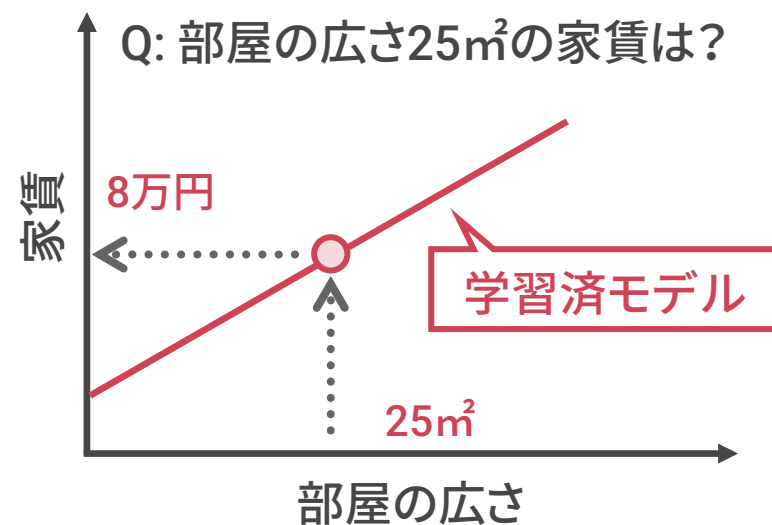
“学習”ステップでは、**モデルパラメータ**を学習する。

真値とモデルの出力の誤差が最小になるようモデルを調整する。

“推論”ステップでは、未知の入力から出力を得る。



## 1. 学習ステップ



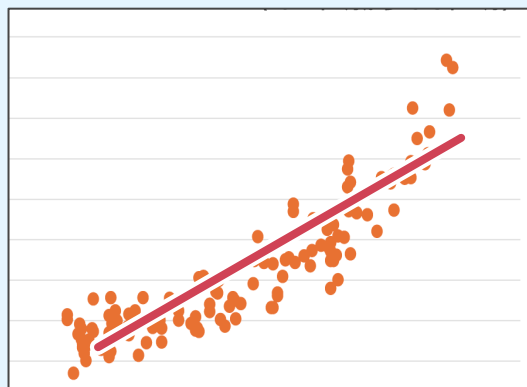
## 2. 推論ステップ



# 回帰分析の種類

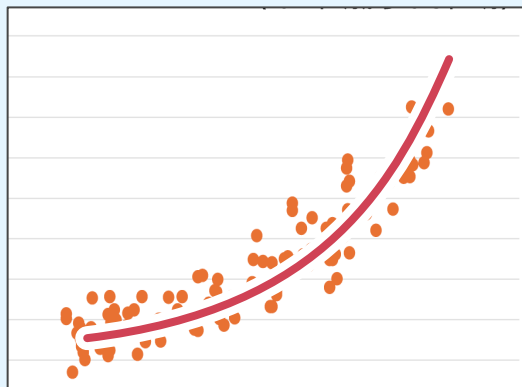
回帰分析は、設定するモデルによって様々な種類が存在する。

## 1. 線形回帰



直線モデル

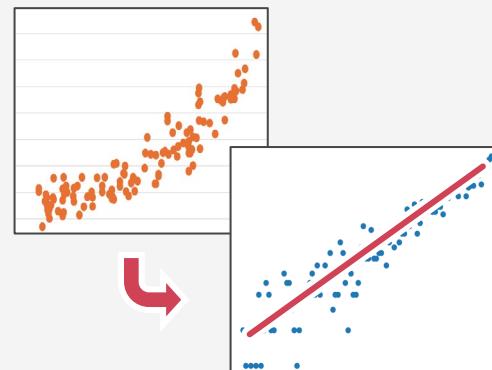
## 2. 多項式回帰



曲線モデル

よく用いられるモデル

## 3. 一般化線形回帰



データを変形して回帰

# 1. 線形回帰

線形回帰は，入力と出力の関係を直線で表したものの．

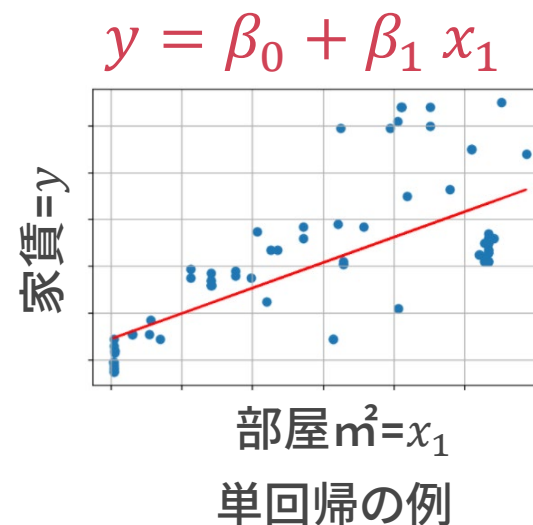
## 線形回帰モデル

### 入力の特徴量が1つ = 単回帰

$$y = \beta_0 + \beta_1 x_1$$

出力   切片   傾き   入力

例) (家賃) =  $\beta_0 + \beta_1$ (部屋 $\text{m}^2$ )



### 入力の特徴量が複数 = 重回帰

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots$$

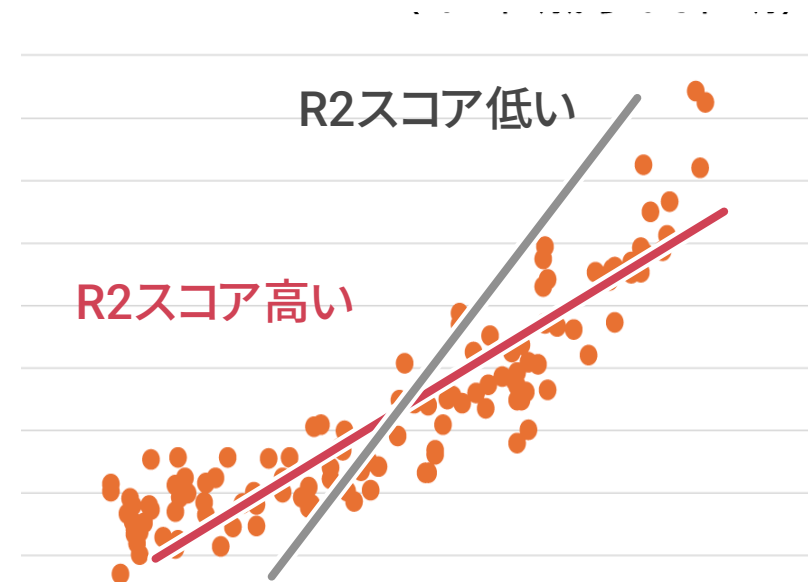
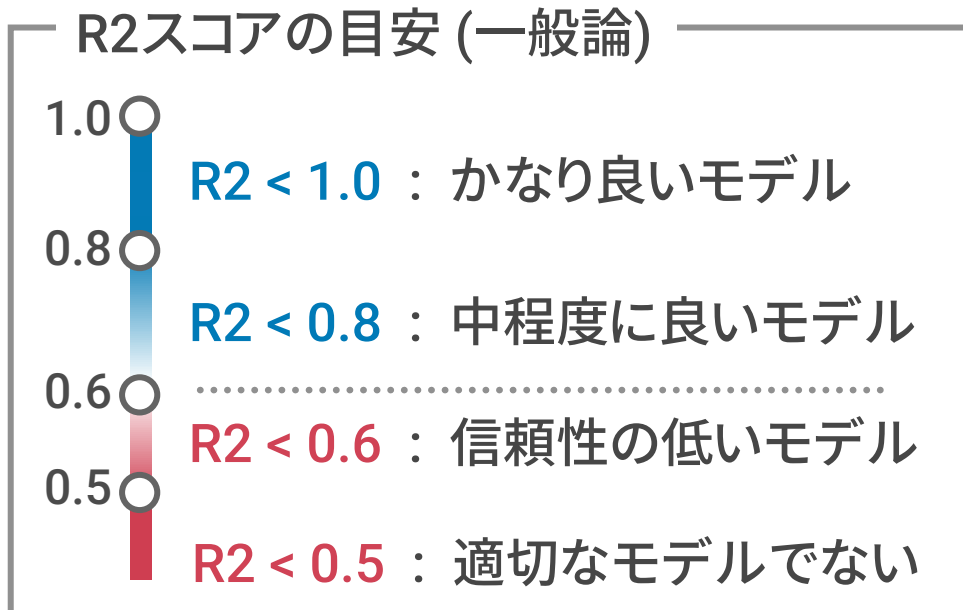
出力   切片   傾き   入力   傾き   入力

例) (家賃) =  $\beta_0 + \beta_1$ (部屋 $\text{m}^2$ ) +  $\beta_2$ (築年数)

重回帰のプロットには  
3次元以上の空間が必要．  
一般にはプロットは難しい

# モデルの性能評価指標

R2スコアは、モデルがどれだけデータにフィットしているかを測る。  
R2スコアが1に近いほど、モデルがデータにフィットしていると言える。



データによっては  $R^2 > 0.6$  とならない場合も.あくまで参考に

## 2. 多項式回帰

多項式回帰は，入力のn乗を特徴量に加えること．  
直線ではなく曲線をフィットさせることができる．

### 多項式回帰モデル

#### 単回帰 (再掲)

$$y = \beta_0 + \beta_1 x_1$$

例) (家賃) =  $\beta_0 + \beta_1(\text{部屋m}^2)$



#### 多項式回帰

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \dots$$

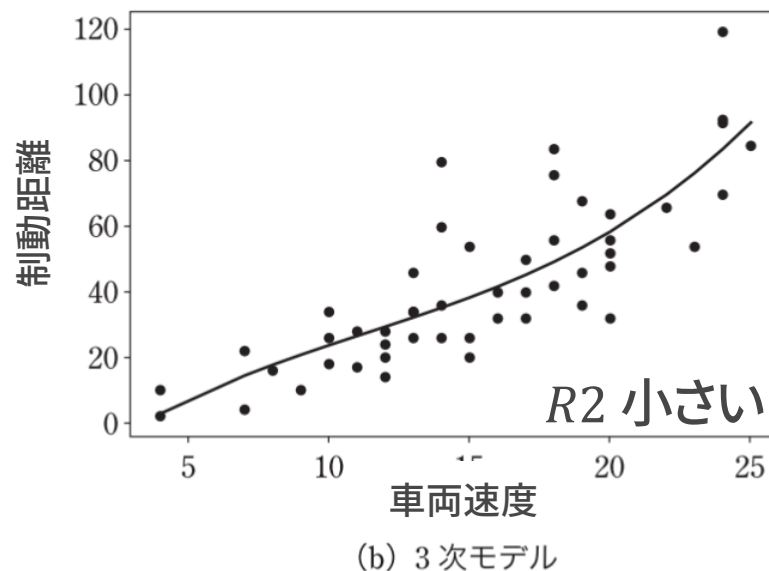
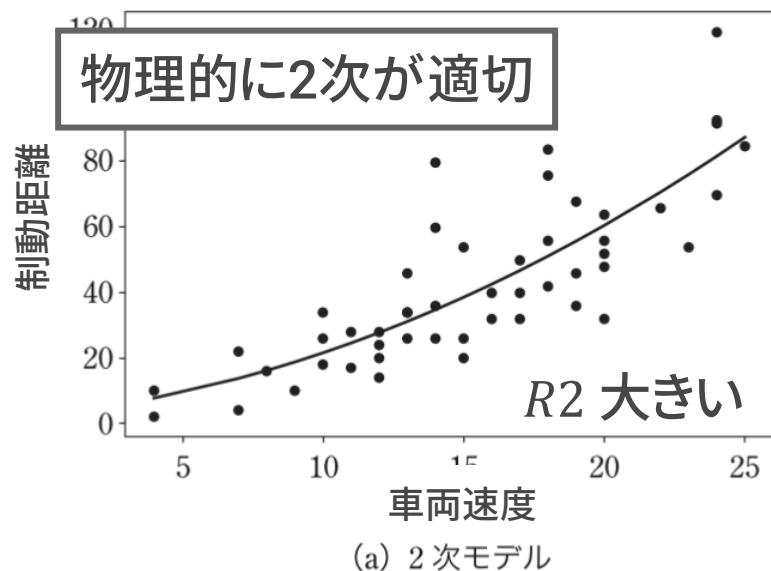
出力   切片   傾き   入力   傾き   入力

何次まで追加するかは任意

例) (家賃) =  $\beta_0 + \beta_1(\text{部屋m}^2) + \beta_2(\text{部屋m}^2)^2$

# 過学習の危険性

多項式回帰では、 $n$ 乗を増やしていくとモデルの形は複雑になる。  
複雑なモデルになるほど、 $R^2$ スコアが小さくなるが、  
未知データに対して予測性能が低下する**過学習**が発生する。



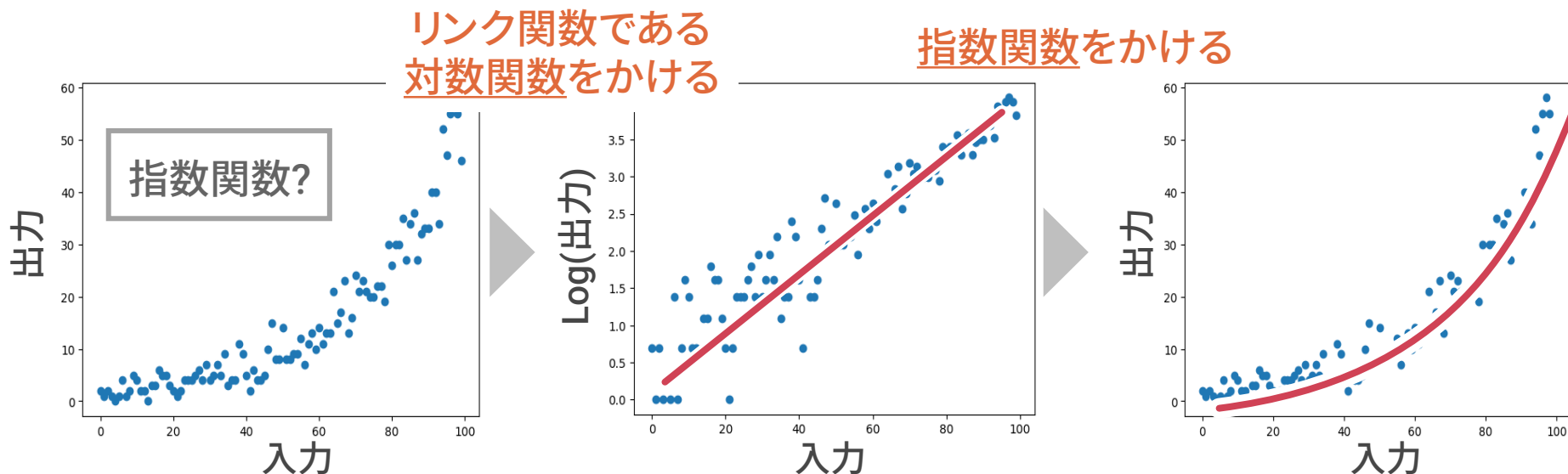
物理的な観点などでデータをよく観察し、モデルを選別すること

### 3. 一般線形回帰

自然現象の多くで、入力と出力に線形な関係にないことが多い。

その場合、出力をリンク関数(逆関数)で補正して予測する  
**一般線形回帰**が用いられる。

一般に、データが指数関数に従うことが多く、リンク関数として  
対数関数を採用することが多い。



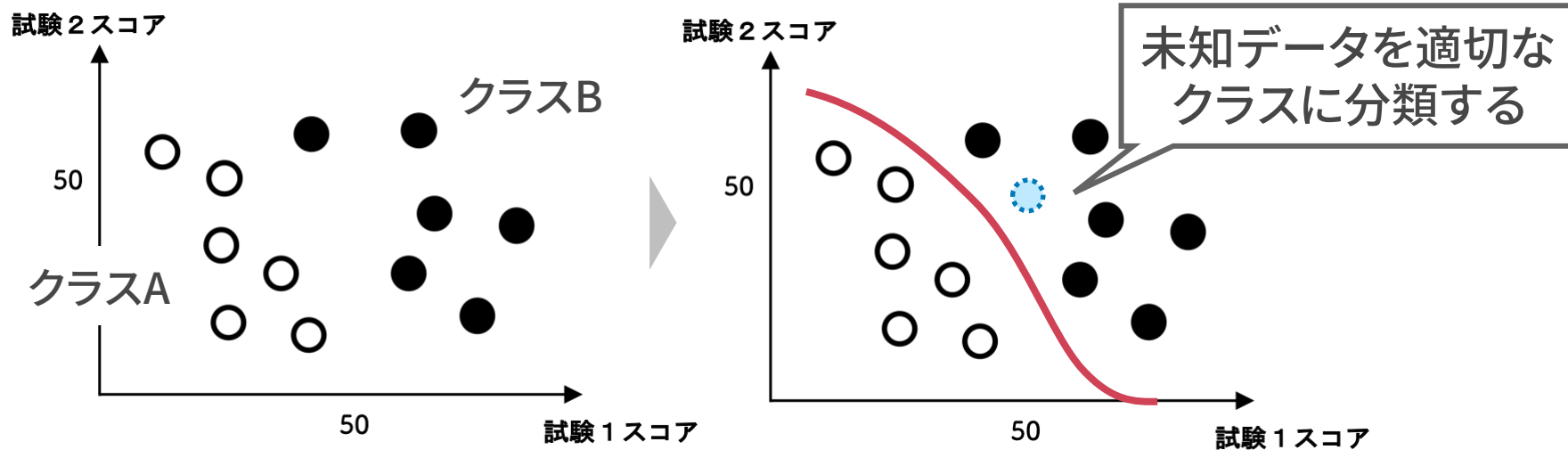
データの観測を通じて、最適なリンク関数を見つける必要がある

## 2.2

# 機械学習の基礎

## 識別

識別は、それぞれのクラスを分別する**識別境界**を求めること。



## 識別の種類:

- サポートベクターマシン (SVM)
- ロジスティック回帰
- **ランダムフォレスト**

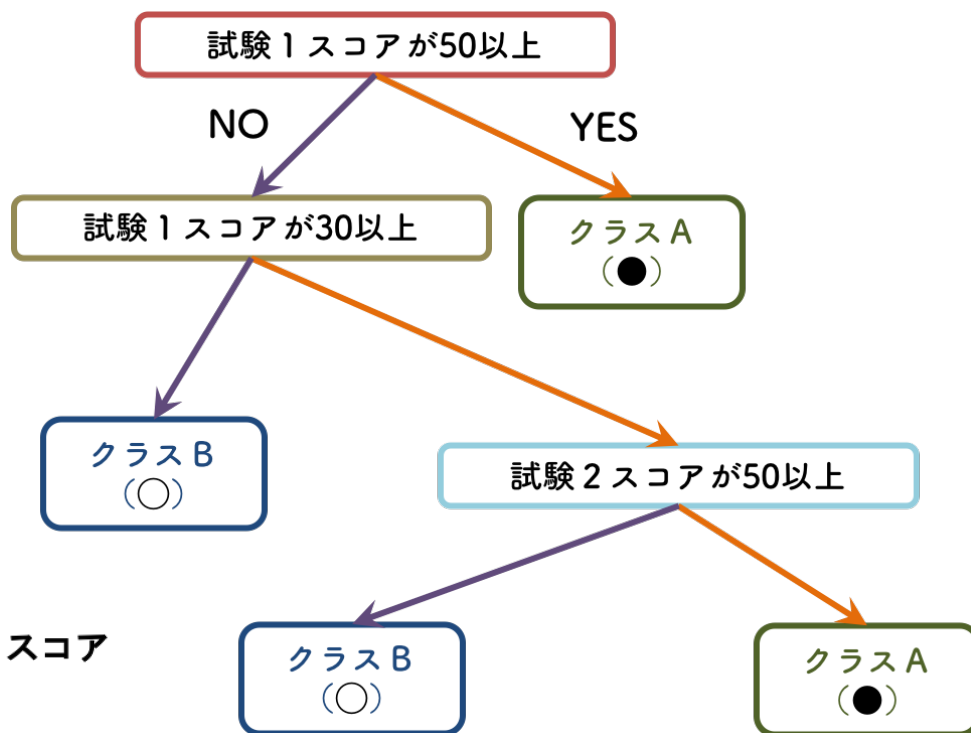
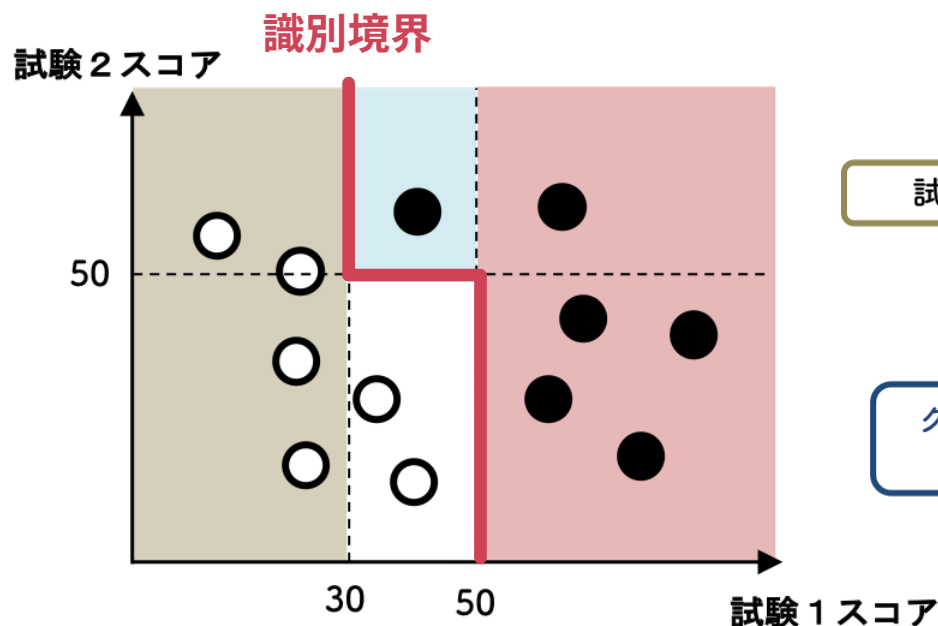
RFは高速に処理でき、  
データを標準化しなくとも  
性能が劣化しない。



# 決定木の原理

ランダムフォレストは決定木に基づいている。

決定木は、**二分木のルール**をつなげて識別を行う。

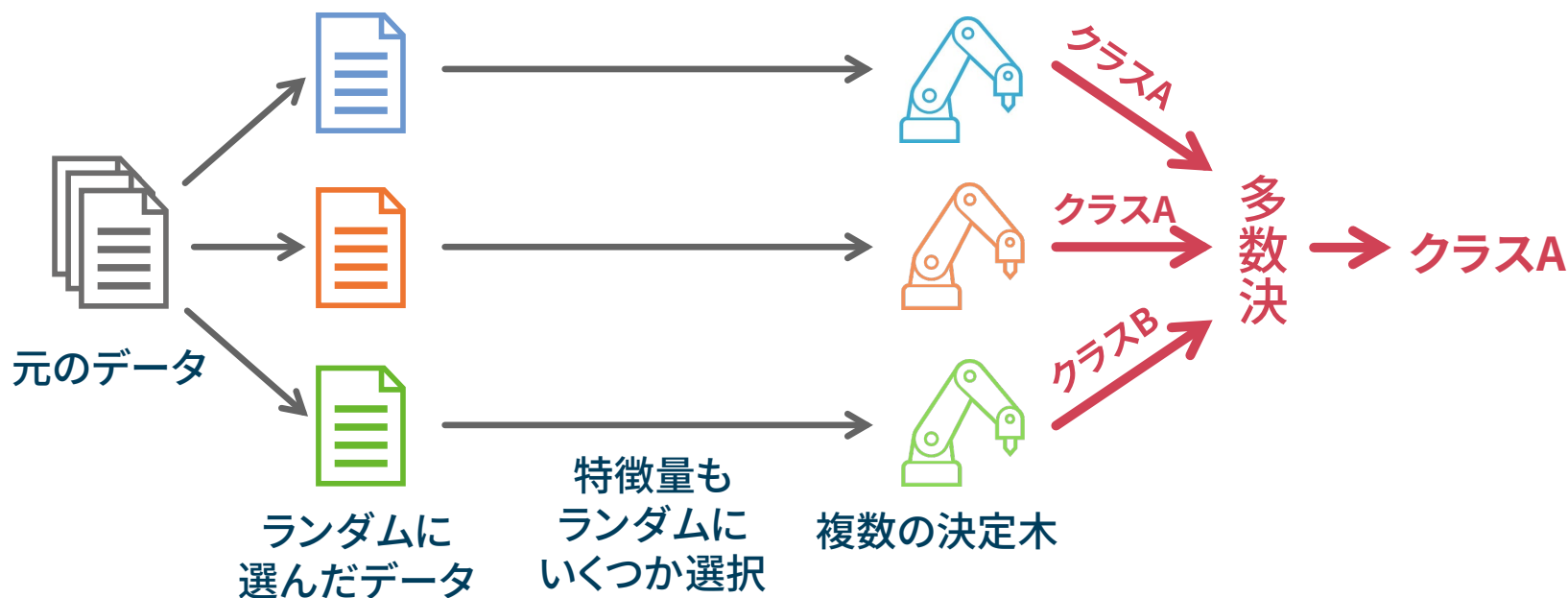


単純な Yes or No の組み合わせで識別できる

# ランダムフォレストの原理

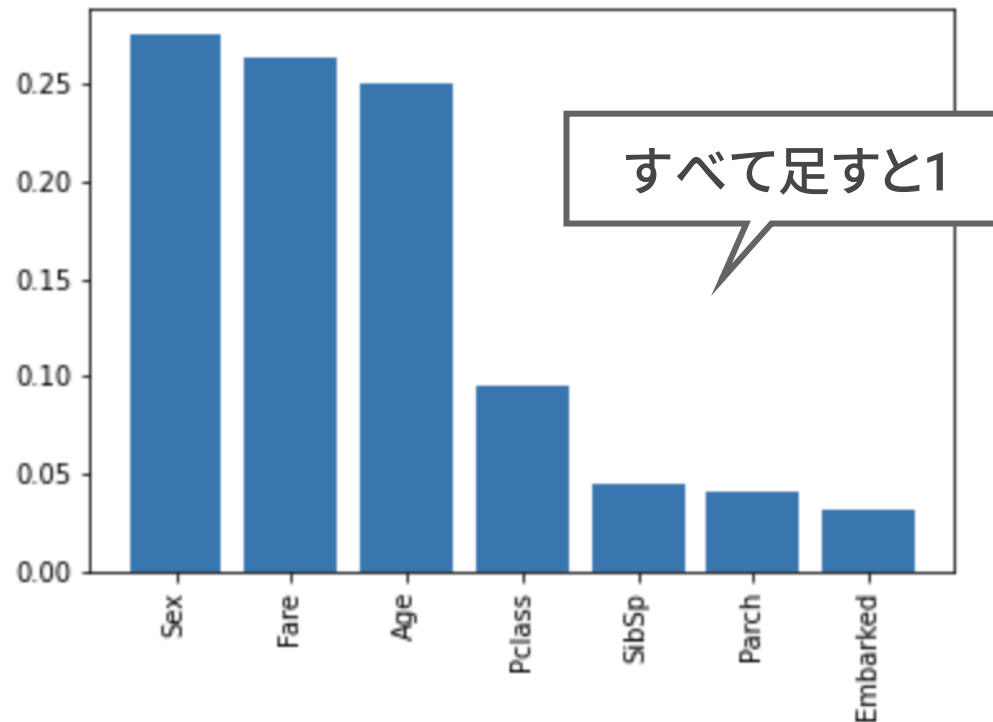
決定木は過学習が発生しやすく、汎化性が低い。

ランダムフォレストはランダムに選んだデータ・特徴量から複数の決定木を作り、多数決を行うことで汎化性を向上させる。



# 特徴量重要度

ランダムフォレストにおいて、特徴量ごとに識別結果にどれくらい寄与するかを測る指標(=**特徴量重要度**)を算出できる。



タイタニック号の生存者に関するデータ

機械学習の判断根拠( = 解釈性)を探る参考に

2.4

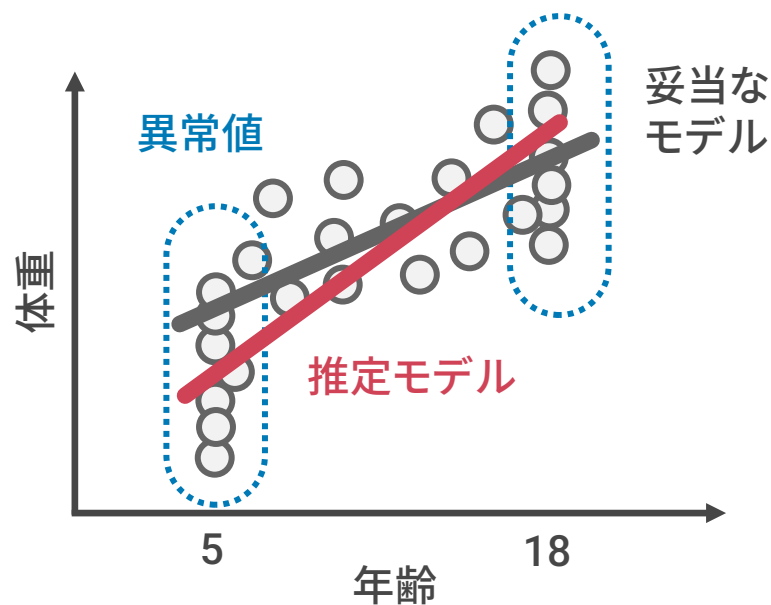
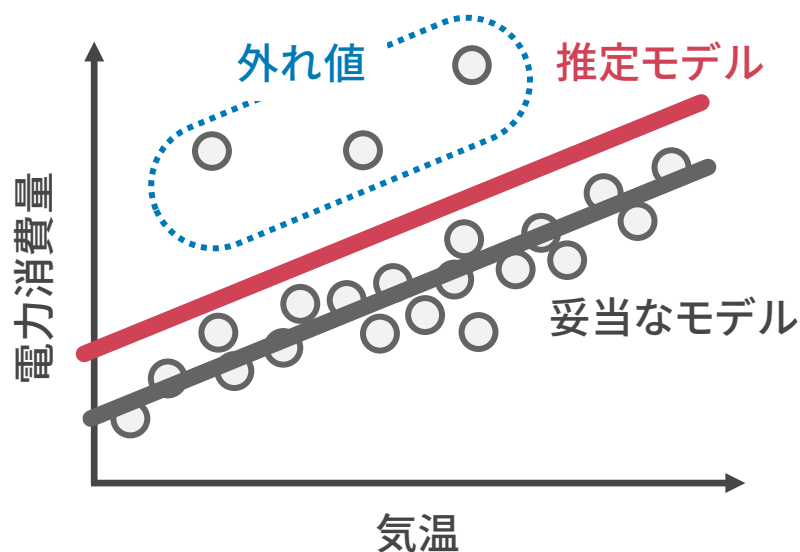
# 機械学習の基礎

## データ整形

# 外れ値の影響

回帰・識別ともに外れ値・異常値があると、その影響で推定されたモデルが妥当でない場合がある。

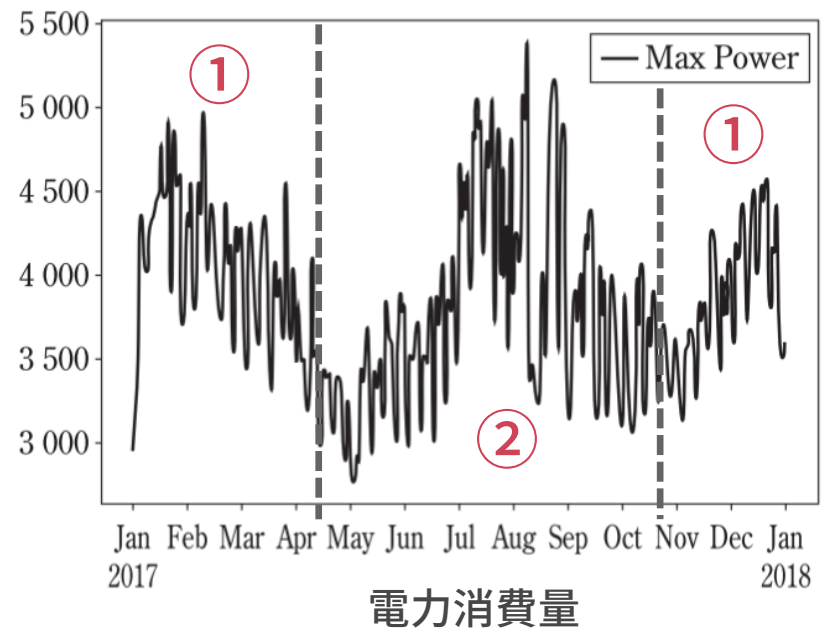
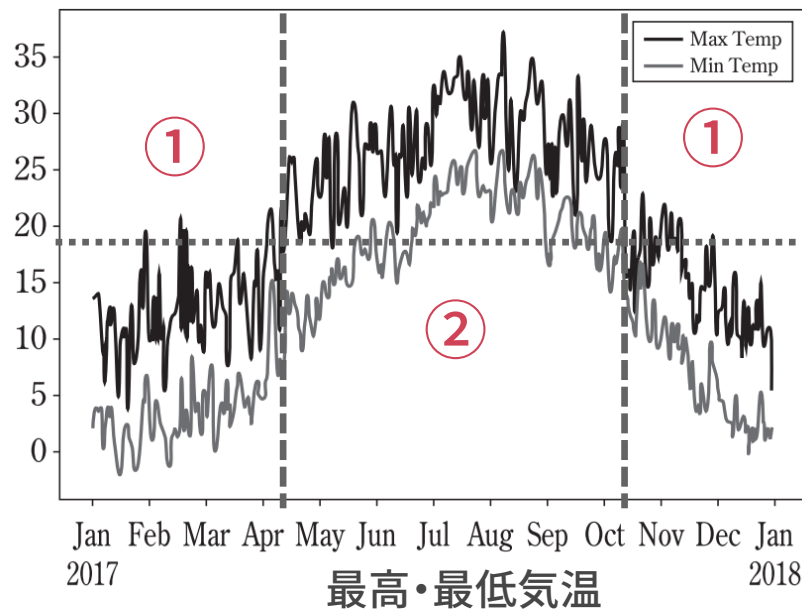
データをよく観察して、適切でないデータは予め削除したほうがよい。



可視化などを通じてデータをよく分析し、不必要なデータは削除する

# データの統計的性質が変化する場合

全体のデータを観察すると傾向が見えなくとも、  
局所的にデータを区切ると傾向が見えることがある。



① : 気温が低いほど電力消費量高い    ② : 気温が高いほど電力消費量高い

データを区切って解析すれば精度向上することも

3

---

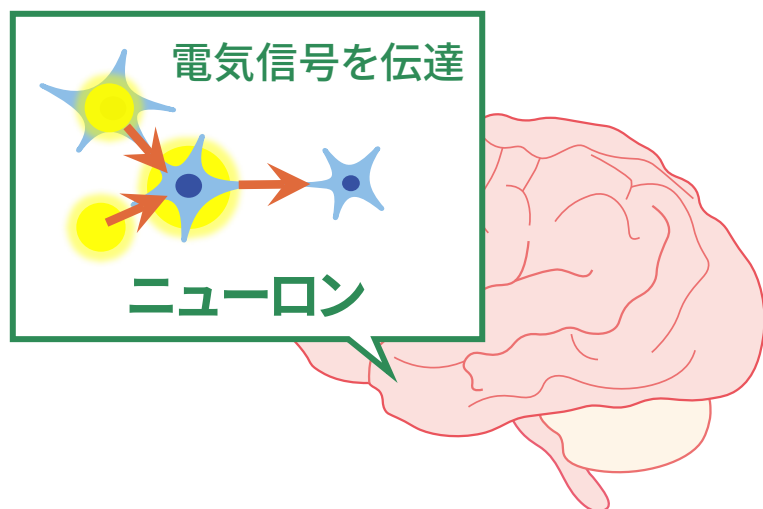
# 深層学習と生成AIの活用

# 深層学習について

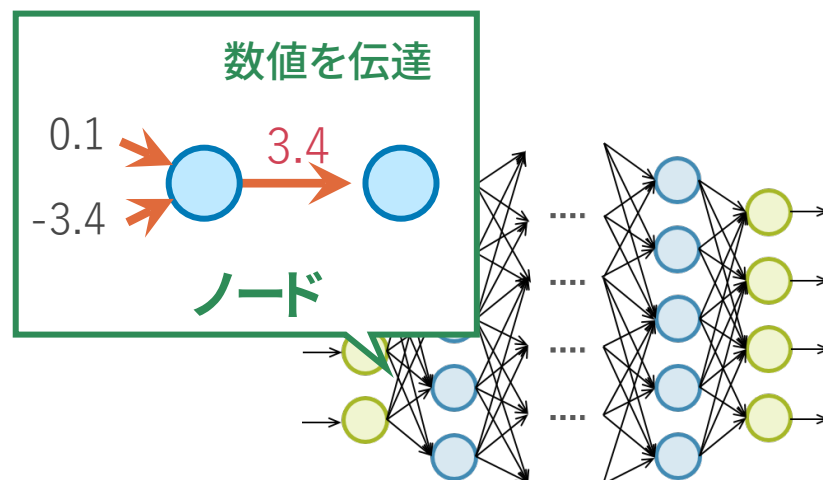
最近では深層学習が優れた回帰・認識性能を発揮する例も多い。

深層学習とは**深層ニューラルネットワーク**を特定のタスクをこなすよう、内部のパラメータを学習させたもの。

⇒ 脳の構造をコンピュータ上で再現したもの。



脳



深層ニューラルネットワーク (DNN)

深層学習を使った処理をみてみよう



3.1

## 深層学習と生成AIの活用

### 深層学習の仕組み

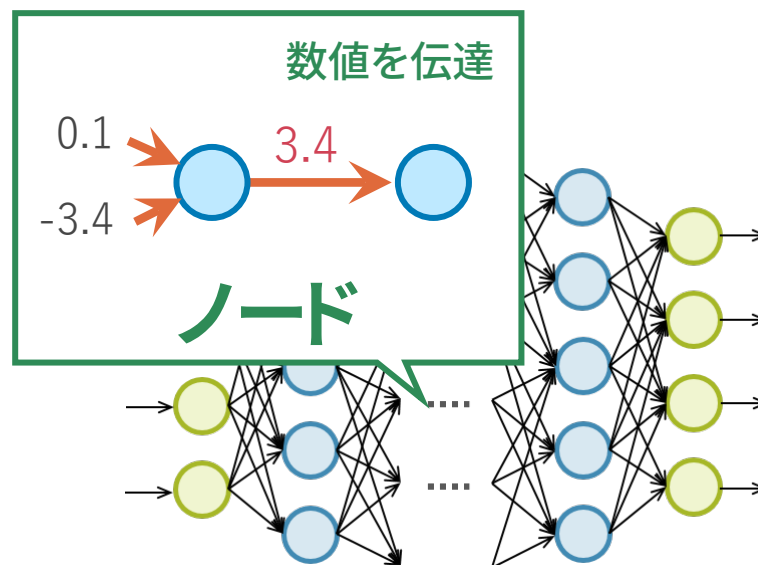
# 深層学習とは？

深層ニューラルネットワークを特定のタスクをこなすよう学習したもの。

⇒ 脳の構造をコンピュータ上で再現したもの。



脳

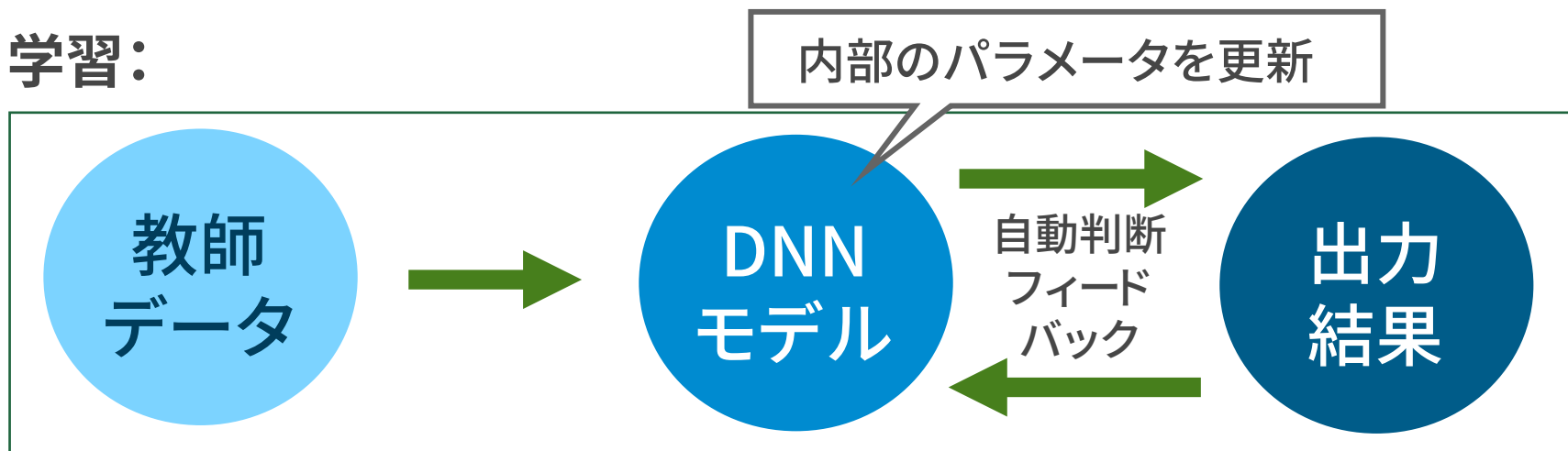


深層ニューラルネットワーク (DNN)

人間と同等の処理ができると期待

# 深層学習の概要

学習:



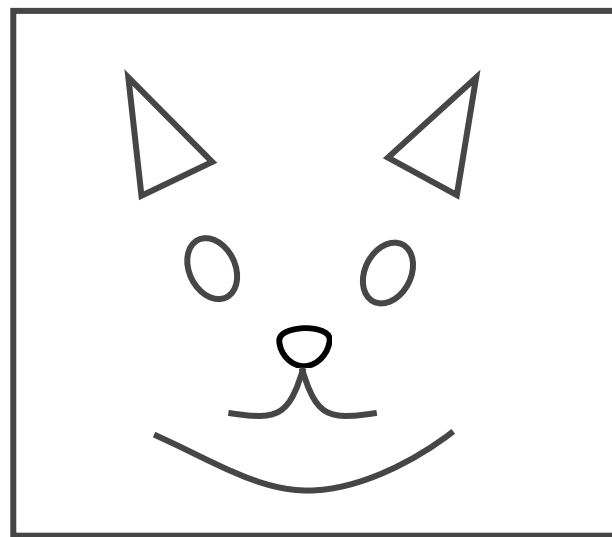
推論:



特徴量抽出・処理手順を自動で獲得する

# 例：画像認識

画像認識であれば、画像から特徴量(パーツ)を抽出して、その位置関係によって物体を認識する。



認識に用いるパーツ(特徴量)は少ない!

重要な特徴量の自動獲得が強みの一つ

# 深層ニューラルネットワークの種類

深層NNには様々なモデルが提案されている。  
データの性質・目的に応じて適切に選択する必要がある。

## 畳み込みニューラルネットワーク(CNN)

特に画像認識において優れた性能を発揮するモデル。  
局所的な特徴を抽出し、徐々に広い範囲を解析する。

## RNN / LSTM

特に時系列データの識別・回帰において優れた性能を発揮するモデル。

## Transformer

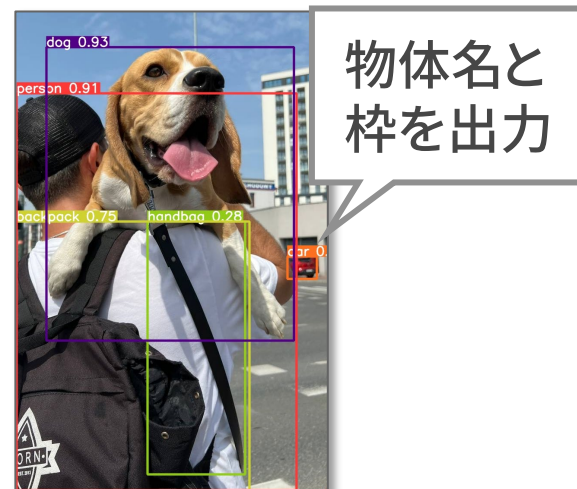
特に時系列データの識別において優れた性能を発揮するモデル。

# 代表的な State-of-The-Arts モデル

最近では最新で最高性能のモデルもGithub等から簡単に利用できる。

## Yolo (ヨーロ)

物体認識において特に優れた性能を発揮。  
Yolo v9 は超大規模セットで学習した、  
軽量かつ高性能なモデル。



## Segformer

画像セグメンテーション用の優れたモデル。  
リアルタイムに物体位置を推定できる。

ピクセル単位で  
位置を特定



学習済モデルも公開されている

# バーニーおじさんのルール

深層学習において、性能を発揮するには**大規模なデータ**が必要。  
大規模データセットがないと、汎化性が著しく低下する。

## バーニーおじさんのルール

機械学習を効果的に学習するには、以下の関係を満たす必要がある。

$$(\text{パラメータ数}) < 10 \times (\text{データ数})$$

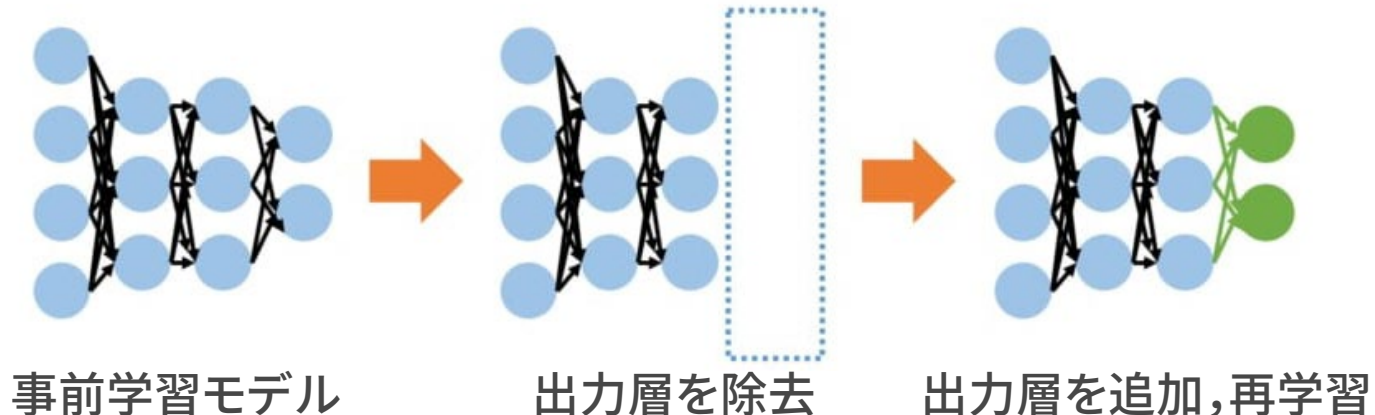
現実的に大量のデータを収集, アノテーション, 整形するのは困難

**転移学習**を活用しよう

優れた深層学習モデルの多くはソースコードが公開されており、また事前学習モデルも公開されていることが多い。

⇒ 事前に大規模データセットで学習されたモデル

事前学習モデルを使いまわしてもよいし、自身のデータセットで再学習(=**転移学習**)するとなお性能が向上する。





# 3.1

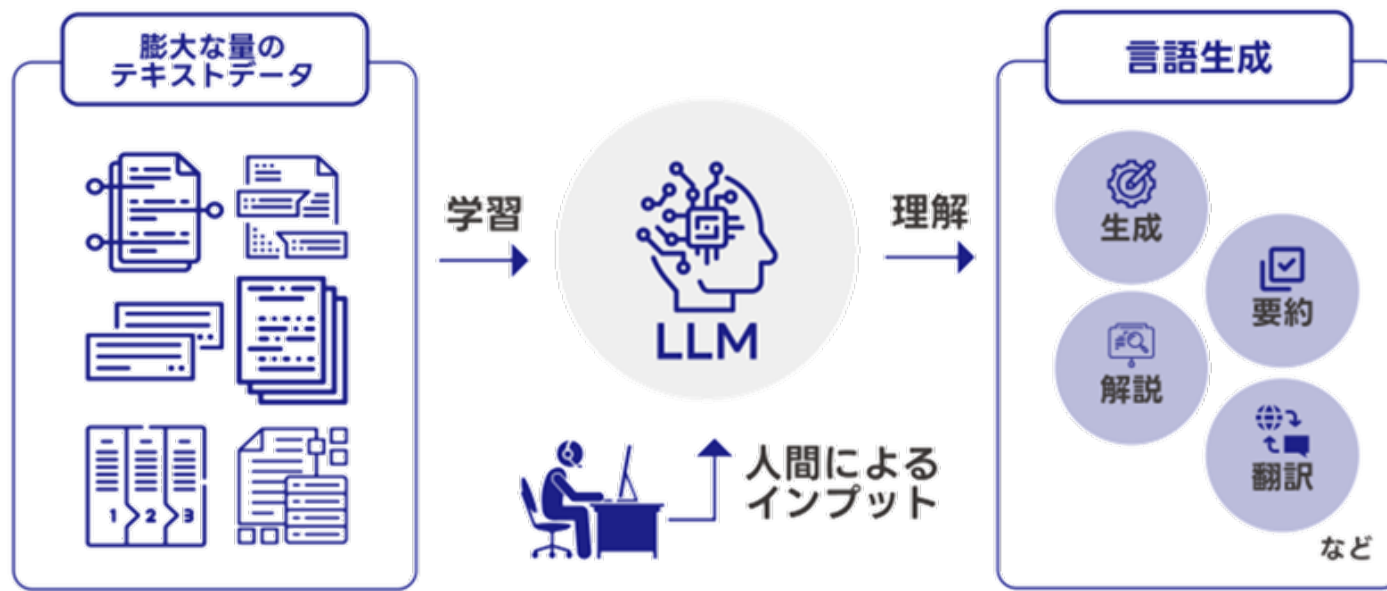
## 深層学習と生成AIの活用

### 生成AIとAI開発

# 生成AI (LLM)とは

ChatGPTをはじめとする**大規模言語モデル(LLM)**は、  
文章作成・要約・翻訳・ソースコード生成など、ありとあらゆる  
文章生成タスクを行うことのできるモデル。

⇒ 数100億程度の文章(Wiki, ニュース等)を収集して学習



業務の効率化に大きく役立つ

# LLMの活用例 - 1. プロンプトの設定

LLMサービスを活用すれば、ソースコードの開発も容易に。

プロンプト (指示する文章)

Pythonで機械学習を行うコードを作成してください。条件は以下に示します。

1. 今からデータセットをcsv形式で渡します。最後の列を出力に,それ以外を入力とする回帰問題を解きたいです。
2. Google colabを利用しています。
3. 入力データは標準化してください。
4. R2スコアによりモデルの性能を評価してください。
5. 訓練とテストでデータを分けてください。

Point 1

質問の目的を明確に示す。

Point 2

入力・出力・方法を示す。

できればデータも渡す(有料)。

Point 3

条件もできるだけ細かく。

具体的に・明確にプロンプトを書けば適切な回答が得られる

# LLMの活用例 - 2. ChatGPTの回答例



データセットが読み込まれ、構造が確認できました。次に、条件に基づいて機械学習のコードを作成します。

python

📄 コードをコピーする

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

# Load the dataset
file_path = '/path/to/your/boston.csv'
data = pd.read_csv(file_path)

# Split the data into input (X) and output (y)
X = data.iloc[:, :-1]
```



具体的に・明確にプロンプトを書けば適切な回答が得られる

# LLMの活用例 - 3. 回答の修正

もし回答が気に入らなければ,その都度投げかければ良い.

コードを実行してエラーが出れば,エラー文をプロンプトに貼り付けて実行すれば,修正したコードを提示してくれる.

続きのプロンプト

先程のコード,結果を図示したいのですが,いいやり方を教えて下さい.



結果を図示するためには,回帰モデルの予測値と実際の値を比較するプロットを作成すると良いでしょう。例えば,散布図で実際の値と予測値をプロットし,予測値がどれだけ実際の値に近いかを視覚的に確認できます。

以下に,回帰モデルの予測結果を図示するためのコードを追加します:

```
python
```

📋 コードをコピーする

**対話できるのが強み.曖昧な質問でも提案してくれる**