

G検定対策講座 第7回

教師あり学習

正解付きのデータから学ぶ



今日学ぶこと

1

教師あり学習とは

正解ラベルとのペアから学ぶ

2

分類と回帰

出力が離散値か、連続値か

3

線形モデルの仲間

線形回帰・ロジスティック回帰・自己回帰モデル

4

木の仲間

決定木

5

SVM

マージン最大化とカーネル

6

アンサンブル学習

バギングとブースティング

AIに答えを 教えるには？

正解付きのデータを、たくさん見せればいい



基本中の基本

教師あり学習とは

特徴量と**正解ラベル**のペアから学ぶ機械学習



ラベルのない教師なし学習、
報酬から学ぶ強化学習と区別する

具体例

猫を見分けられるようになるまで

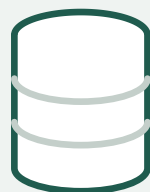


「猫」

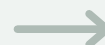
写真＋正解ラベル



大量に見せる



学習



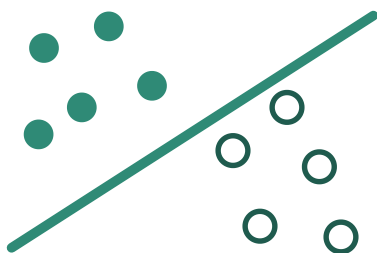
新しいデータの
正解を予測できる

POINT

「特徴量と教師データのペアが必要」という点が、学習目標に明記されている

分類 と 回帰

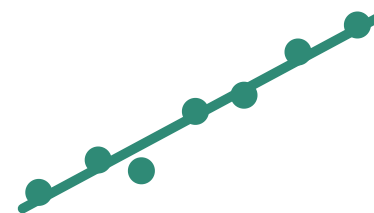
教師あり学習は、出力の種類でふたつに分かれる



分類

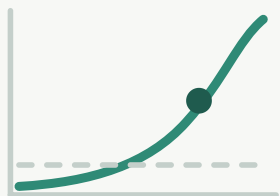
出力が離散値（カテゴリ）
スパムか否か・病気の有無
境界線を引いて仲間分けする

≠



回帰

出力が連続値（数値）
家賃の予測・売上の予測
点の傾向に線を当てはめる



いちばん基本

線形回帰

入力と出力を直線（または平面）で結ぶモデル

入力＝説明変数
出力＝目的変数

具体例

点の真ん中を通る直線を引く



点＝実データ／線＝学習した予測

単回帰分析

説明変数がひとつ（広さ だけ）

重回帰分析

説明変数が複数（広さ・駅距離・築年数）

POINT

直線の傾きと切片を学習で求める。解釈しやすく、いまもよく使われる



名前にだまされない

ロジスティック回帰

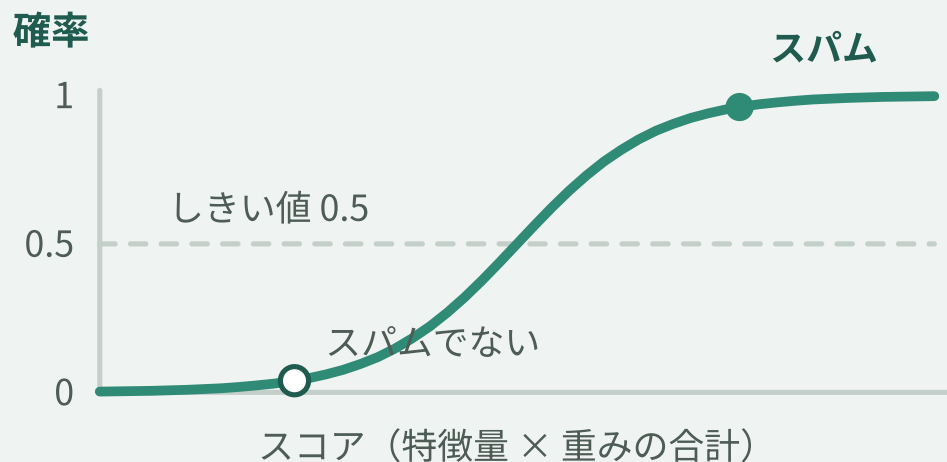
名前は回帰だが、実は分類に使うモデル



直線をそのまま使うと
確率が1を超えてしまう。
だからS字に押し込める

具体例

シグモイド関数で、出力を確率に変える



どんな入力でも
0~1 に収まる

S字なので
境界のあたりだけ大きく動く

POINT

「回帰」という名前だが用途は分類——ここが最大のひっかけどころ



1958年 コックス

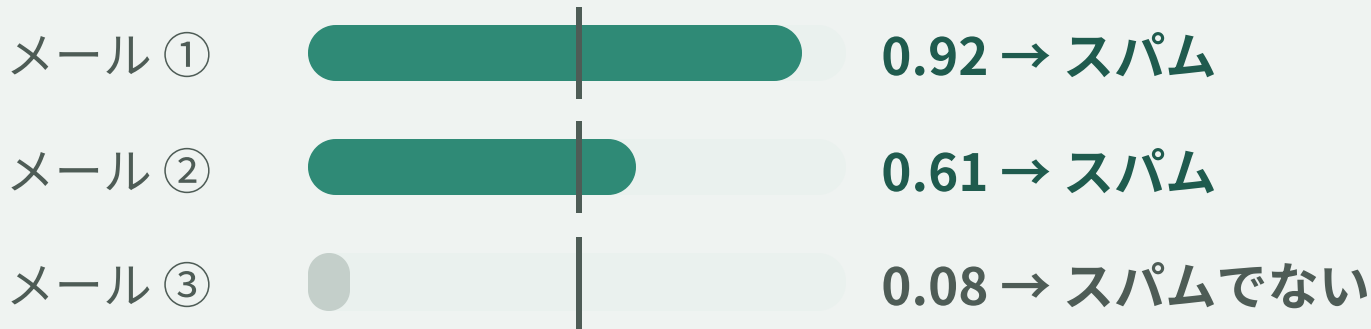
しきい値 0.5 で判定する

確率が0.5 より大きければスパムと決める



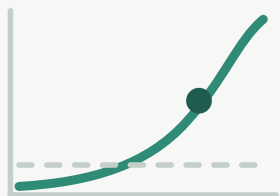
出力が確率なので
判定の信頼度も一緒に得られる

具体例 メールスパム判定



真ん中の縦線が しきい値 0.5

POINT ニクラス分類の代表。提唱はコックス（1958年）



時系列 ①

自己回帰モデル (AR)

ひとつの変数の過去の値から、未来を予測する



時系列データ＝
時間順に並んだデータ

具体例

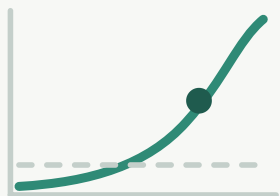
過去の値だけを手がかりにする



きのうまでの株価から
あすの株価を予測する

自分自身の過去に「回帰」す
るので
自己回帰

POINT AR は Auto-Regressive (自己回帰) の略



時系列 ②

ベクトル自己回帰 VAR

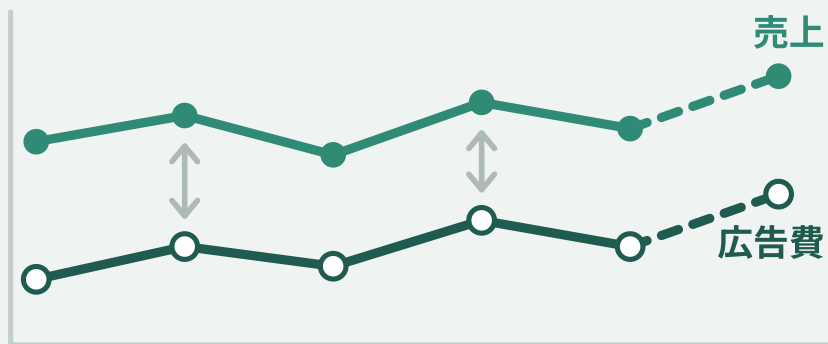
複数の変数を、互いの関係もふくめて予測



派生モデルに
ARIMA・SARIMA

具体例

変数どうしが影響し合う



時間 →

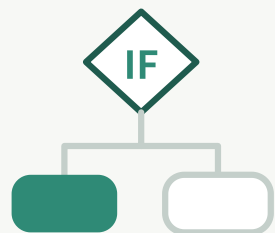
互いの影響もふまえて予測する

広告費が増えると売上が伸び
売上が伸びると広告費も増える

AR = ひとつの変数
VAR = 多変数

POINT

「AR はひとつの変数／VAR は多変数」という対比が頻出



1980年代

決定木

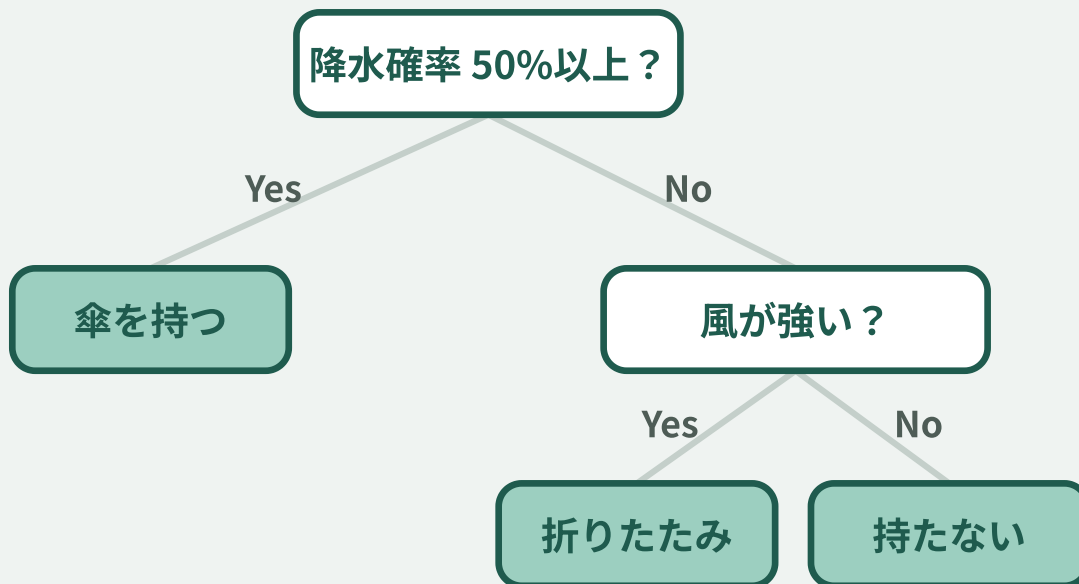
Yes / No の分岐でデータを分類していくモデル



分岐をたどるだけなので
なぜその答えかを説明できる

具体例

きょう傘を持つか、を決める木



構造がフローチャートそのもの

だから解釈性が高い
人間が判断の理由をたどれる

POINT ブラックボックスになりがちなAIの中で、説明できることが決定木の価値

弱点



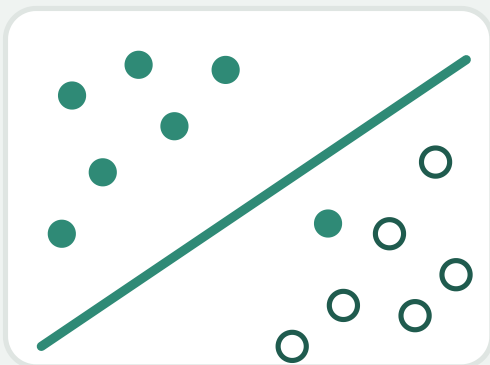
決定木は過学習しやすい

深くしすぎると、訓練データに合わせすぎる

代表例は ID3 (キンラン 1986)
と CART (ブライマン 1984)

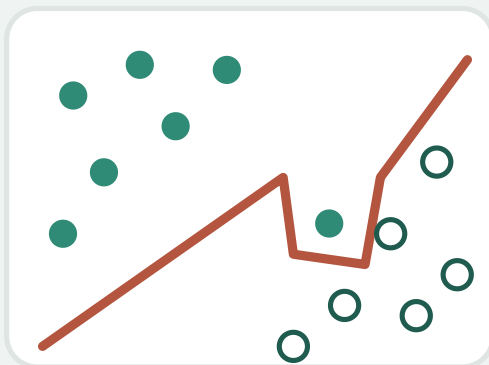
対比

ちょうどよい / 深くしすぎ



ちょうどよい

1つ外しても、境界はまっすぐ



合わせすぎ = 過学習

外れ値ひとつまで困りにいく

訓練データは完璧に分けられるが
新しいデータで外すこの弱点がアンサンブル学習へ
発展する動機になった

POINT

「決定木は解釈しやすいが、過学習しやすい」が頻出



1963年～

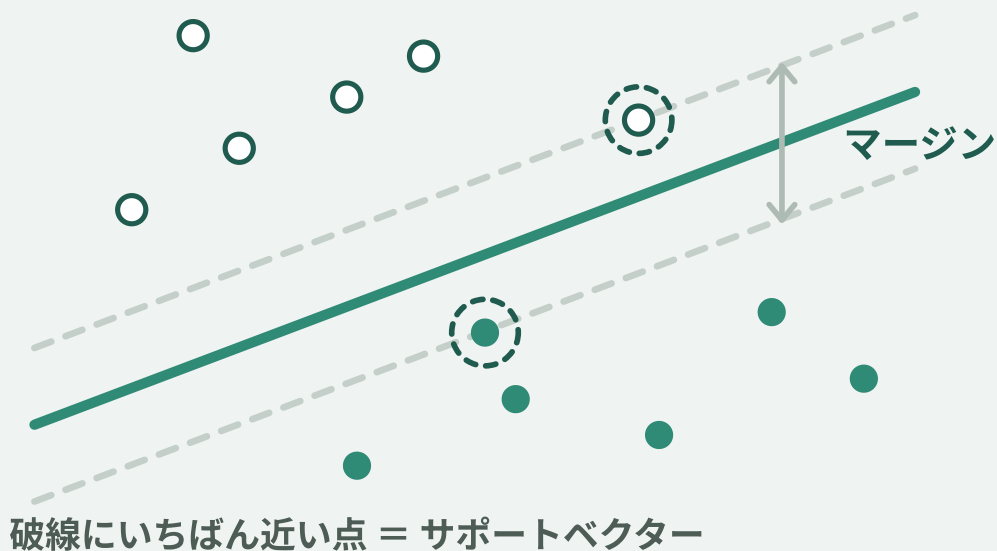
SVM | マージン最大化

境界線といちばん近い点との距離を、最大にとる



ぎりぎりの境界より
まん中を通る境界のほうが
新しいデータに強い

具体例 境界とデータの距離を、できるだけ大きく



破線のすきま = マージン

思いきり離して引くほど
汎化性能が高くなる

POINT 境界を決めているのは、ごく少数のサポートベクターだけ

非線形も分けられる



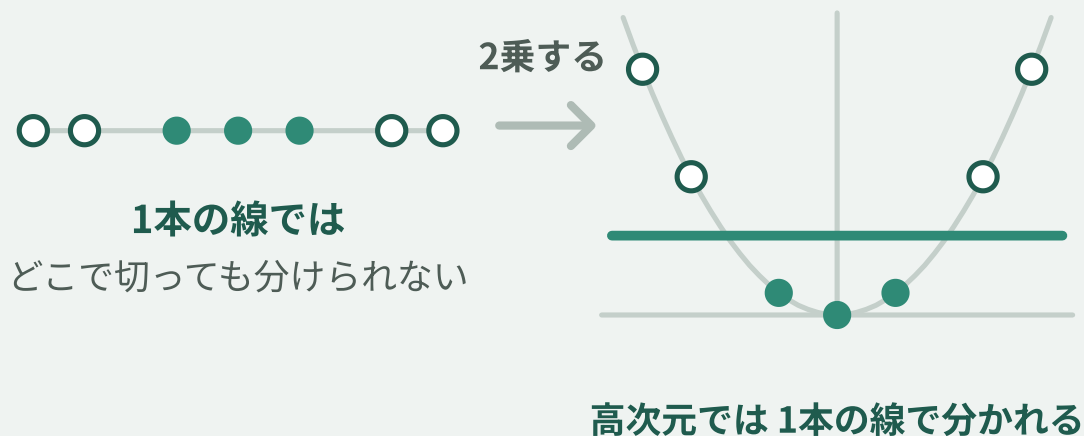
SVM | カーネルトリック

次元を上げると、直線で分けられるようになる



バプニクラが1963年に線形版
1990年代にカーネル拡張で完成

具体例 1次元では無理 → 2次元に持ち上げる



カーネル

データを高次元に写像する関数

カーネルトリック

高次元の計算をせずに
同じ答えを出す巧みな技

POINT

「マージン最大化とカーネルトリックで、非線形も分類できる」が頻出



組み合わせる

アンサンブル学習

複数のモデルを組み合わせで**精度を上げる**方法



ひとつのモデルを使うより
頑健で高精度になる

対比

バギング / ブースティング

並

バギング

並列に学習させて多数決をとる

それぞれが独立に学ぶ

直

ブースティング

直列に学習させて誤りを補強していく

前のモデルの失敗を次が学ぶ

POINT

「バギングは並列、ブースティングは直列」——この対比を覚える



並列

バギング／ランダムフォレスト

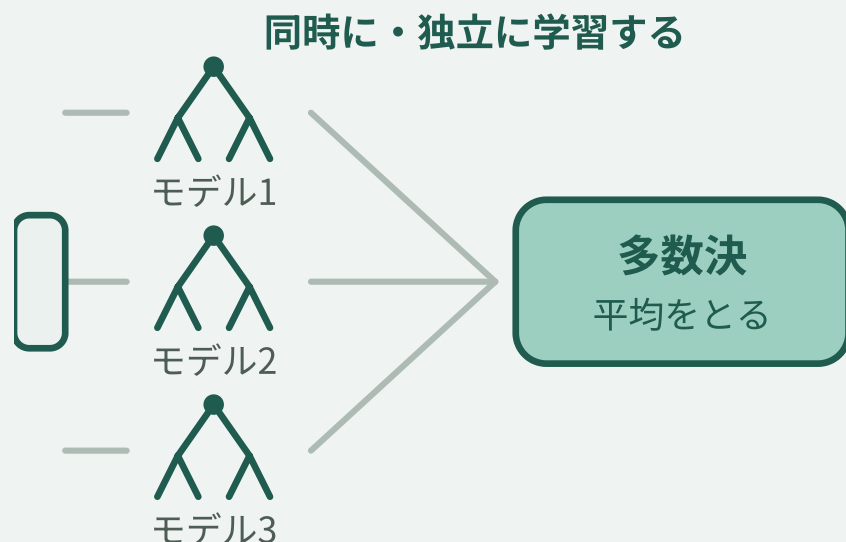
データを分けて**並列に学習**し、最後に多数決をとる



バギング＝ブートストラップ・
アグリゲーティングの略

具体例

同時に学ばせて、答えを合わせる



元データから復元抽出で
少しずつ違うデータを作る

ランダムフォレスト
＝ 決定木 ＋ バギング
ブライマンが2001年に提唱

POINT

「ランダムフォレスト＝決定木 ＋ バギング」が頻出

直列



ブースティング

弱いモデルを**直列**に学習させ、前の誤りを次が直す



代表例は **XGBoost**・**LightGBM**
データ分析コンペで強い

具体例

間違えたところを、次のモデルが重点的に学ぶ



勾配ブースティングは
誤差の勾配を使って次を改善

高精度だが過学習には注意

POINT バギングは独立に学ぶ／ブースティングは**前のモデル**を見て学ぶ

