

ARIMアカデミー
データ構造化ワークショップ2025
Python中級者向け

機械学習 基礎



Smart Solutions株式会社

機械学習の目的

機械学習の目的

- 機械学習モデルを用いて、入力されたデータから、何かの数値を予測したり、いくつかの選択肢からどれに該当するか分類したり、特徴ごとにクラスタリング(グループ分け)したりするのが目的
- 予測、分類、クラスタリングなどを行う仕組み(関数など)をコンピュータ上で実現したものを、機械学習モデルと呼ぶ



予測の例

- 地理的条件から不動産価格を予測

分類の例

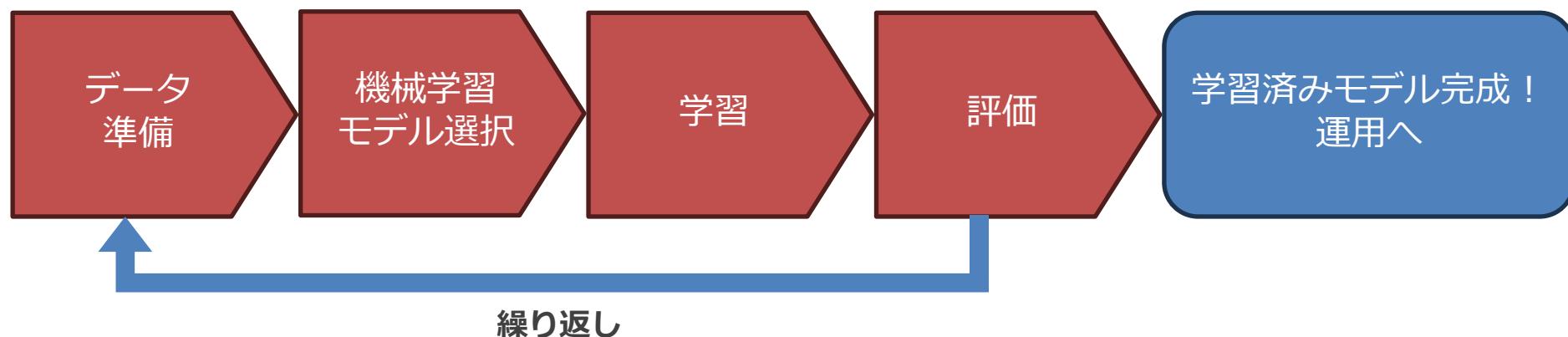
- メールの件名やタイトルから迷惑メールかどうかを分類

クラスタリングの例

- 顧客の属性や行動から、複数の顧客セグメントに分割

機械学習の大まかな流れ

1. 機械学習モデルの学習に用いる**訓練データ(学習データ)**と、機械学習モデルの評価に用いる**テストデータ**を準備する
2. 目的に合わせて機械学習モデルを選択する。このときの機械学習モデルはまだ学習を行っていないため、**未学習モデル(初期モデル)**と呼ぶ
3. 未学習モデルに訓練データを与えて学習させることで、**学習済みモデル**を作成する
4. テストデータを用いて学習済みモデルを評価する。目標とする評価結果が得られるまで、ここまで手順を何度も繰り返す



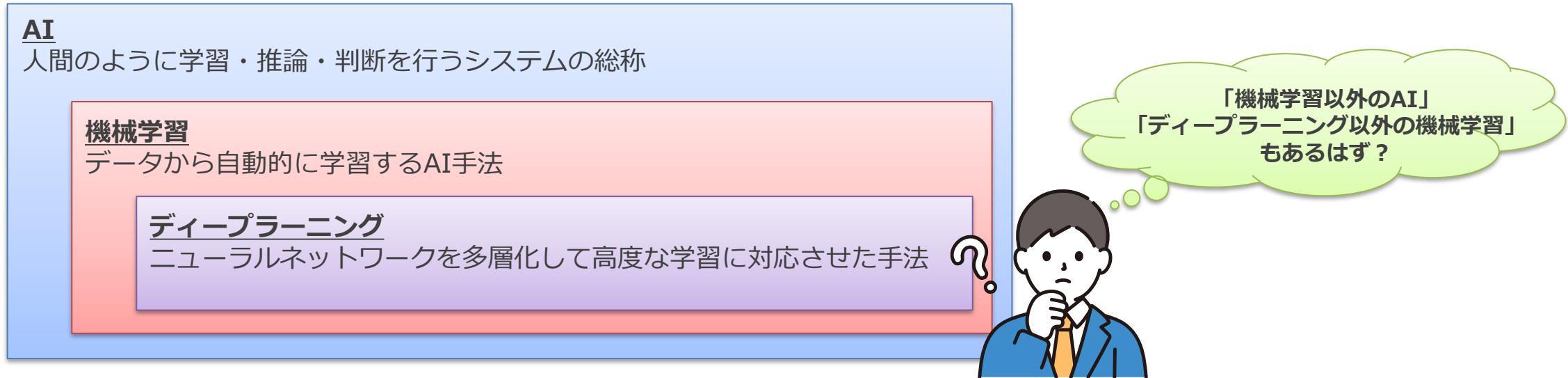
「モデル」について

- ここまでに登場した「機械学習モデル」「未学習モデル」「学習済みモデル」や、単に「モデル」などの用語が、機械学習の分野では頻出する
- とにかく定義がややこしい！使う人によって意味が異なっていることが多い
- 本講義では、以下の意味で統一します
 - 他の書籍やWeb記事では違う意味で使われることもあるので、注意してください。

用語	意味
機械学習モデル	予測、分類、クラスタリングなどを行う仕組み(関数など)を、コンピュータ上で実現したもの 学習前・学習済みの区別はしない
未学習モデル	訓練データで 学習する前の機械学習モデル
学習済みモデル	訓練データで 学習した後の機械学習モデル これを作成して、実際に予測、分類、クラスタリングなどで運用することが機械学習の目的
モデル	文脈次第で、上記のどれを指すこともある 便利な言葉ですが・・・混乱のないよう、本講義では基本的に使用しません

機械学習の位置づけ（1）

下記はAIや機械学習を学んでいると、よく見かける図ですが・・・ちょっと物足りない？



機械学習の位置づけ（2）

もう少し詳しく、書いてみました

AI

人間のように学習・推論・判断を行うシステムの総称

機械学習

データから自動的に学習するAI手法

古典的機械学習

回帰、SVM、決定木など、ニューラルネットワーク以外の従来手法

ニューラルネットワーク

人間の脳の神経回路(ニューロン)を機械学習モデルとして用いた手法

ディープラーニング

ニューラルネットワークを多層化して高度な学習に対応させた手法

単層パーセプトロンなど

浅いニューラルネットワークの手法

ルールベース、知識ベースなど

人間が事前に設定したルールやデータから判断を行うAI手法

機械学習の位置づけ（3）

もう少し詳しく、書いてみました

AI

人間のように学習・推論・判断を行うシステム

今回の講義はここ！

近年では、単に「機械学習」というと、古典的機械学習のみを指すことが多いです。

機械学習

データから自動的に学習するAI手法

古典的機械学習

回帰、SVM、決定木など、ニューラルネットワーク以外の従来手法

ニューラルネットワーク

人間の脳の神経回路(ニューロン)を機械学習モデルとして用いた手法

ディープラーニング

ニューラルネットワークを多層化して高度な学習に対応させた手法

単層パーセプトロンなど

浅いニューラルネットワークの手法

ルールベース、知識ベースなど

人間が事前に設定したルールやデータから判断を行うAI手法

古典的機械学習

「古典的機械学習」とは

- 回帰、SVM、決定木など、ニューラルネットワーク以外の従来手法による機械学習
- 学習データを手作業でいかに上手に設計するかが、予測精度に大きく影響する

古典的？・・・それって時代遅れで使いどころがないということ？？？ → **NO！！**

「古典的機械学習」の特徴

- ニューラルネットワークと比較して以下の強みがある
 - 少量データでも機能する（ディープラーニングのように何万、何十万といったデータがなくてよい）
 - 計算コストが比較的低い
 - 数学的に理解・解析しやすく、ブラックボックス性が低い

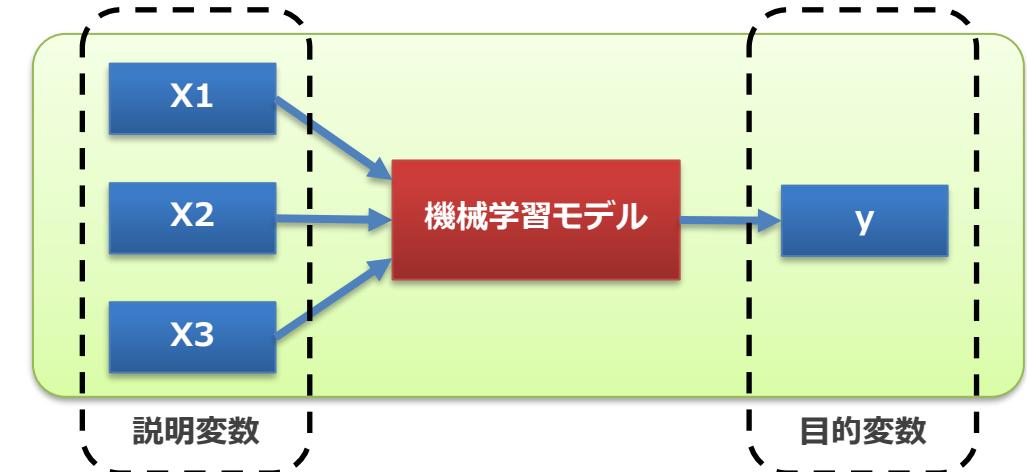
以降、講義中では古典的機械学習を「機械学習」と呼ぶことにします。

機械学習の詳細な流れ

順序	作業	内容
1	目的定義・データ収集	学習の目的を定義し、必要なデータ(生データ)を収集
2	探索的データ分析 (EDA)	データの構造や特徴を分析
3	データ前処理	生データを機械学習に使える形に成形
4	特徴量エンジニアリング	機械学習させやすい情報表現で訓練データを作成
5	アルゴリズム選択・ハイパーパラメータ設定	学習アルゴリズムとアルゴリズムのパラメータを決定
6	学習	訓練データを用いて学習
7	検証・評価	テストデータを用いて学習の結果を評価
8	フィードバック (誤差分析と改善)	評価結果に応じてこれまでのプロセスを再実行
9	運用	学習が完了したら、運用を開始

目的変数・説明変数・特徴量（1）

- 目的変数
 - 機械学習で予測・推論したいデータ（機械学習の出力）
- 説明変数
 - 目的変数を導く根拠とするデータ（機械学習の入力）
- 特徴量
 - 書籍やWebの解説を見ると、2つの異なった意味がある・・・
 - ① 学習対象の特徴を、データとして表現したすべての項目のこと（特徴量の中から目的変数と説明変数を選択）
 - ② 説明変数のこと
 - 機械学習では、（大手企業様の解説サイトも含め）②の意味で使う人が多い印象だが・・・「特徴量エンジニアリング」のプロセスでは①の意味にもなるなど、一貫性がない



これらの用語には国際規格や標準仕様がないので、文脈で判断するしかないのが現状です。

本講義では、より広い概念をカバーできる①に統一いたします。

目的変数・説明変数・特徴量（2）

例：住宅に関する各種データのうち、「面積」と「駅までの距離」から、「住宅価格」を予測する

特徴量 ※定義① 本講義ではこちらの定義を採用！

面積 (m ²)	駅までの距離 (km)	築年数 (年)	住宅価格 (万円)
60	0.5	5	3,500
80	1.2	10	3,800
100	3.0	30	2,500

説明変数 (=特徴量)
※定義②

目的変数

データ尺度

- データを「尺度」という基準で分類できる
- 「尺度」により、適切な前処理やアルゴリズムが異なる

データ種別	尺度	特徴・意味	計算できること	例
カテゴリデータ	名義尺度	名前による区別のみ。順序や数値的な意味はない	一致・不一致	性別(男/女/その他)、血液型(A/B/O/AB)
	順序尺度	順序関係はあるが、間隔は一定でない	大小比較	満足度(高/中/低)、等級(初級/中級/上級)
数量データ	間隔尺度	順序と間に間に意味があるが、絶対的なゼロがない (割合などは意味がない)	加減算	温度(°C)、西暦(年)
	比例尺度	順序・間に加えて絶対的なゼロを持つ (割合などにも意味がある)	四則演算	体重、身長、距離、金額、時間

探索的データ分析(EDA)

- データの傾向・分布・欠損・外れ値などを把握するプロセス
- 仮説を立て、次の処理（前処理・特徴量設計）の方針を決定

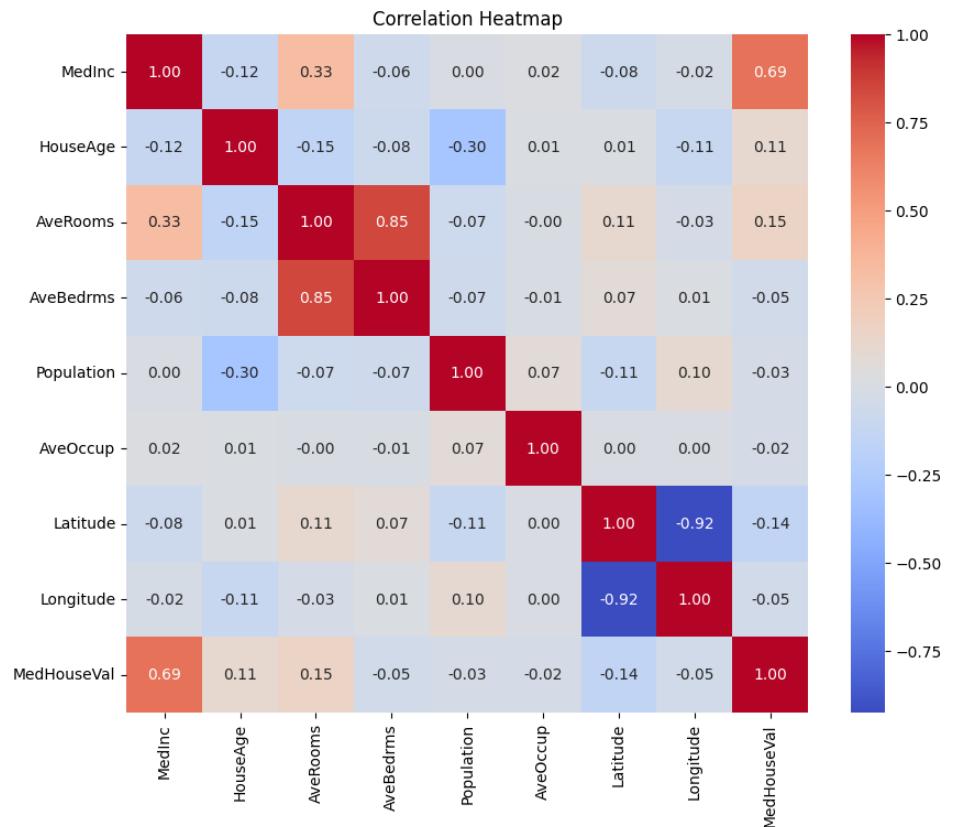
代表的な手法

- 統計量の確認（最大値／最小値、平均値、中央値、四分位数など）
- 欠損や外れ値のパターン確認
- データ間の依存関係の確認

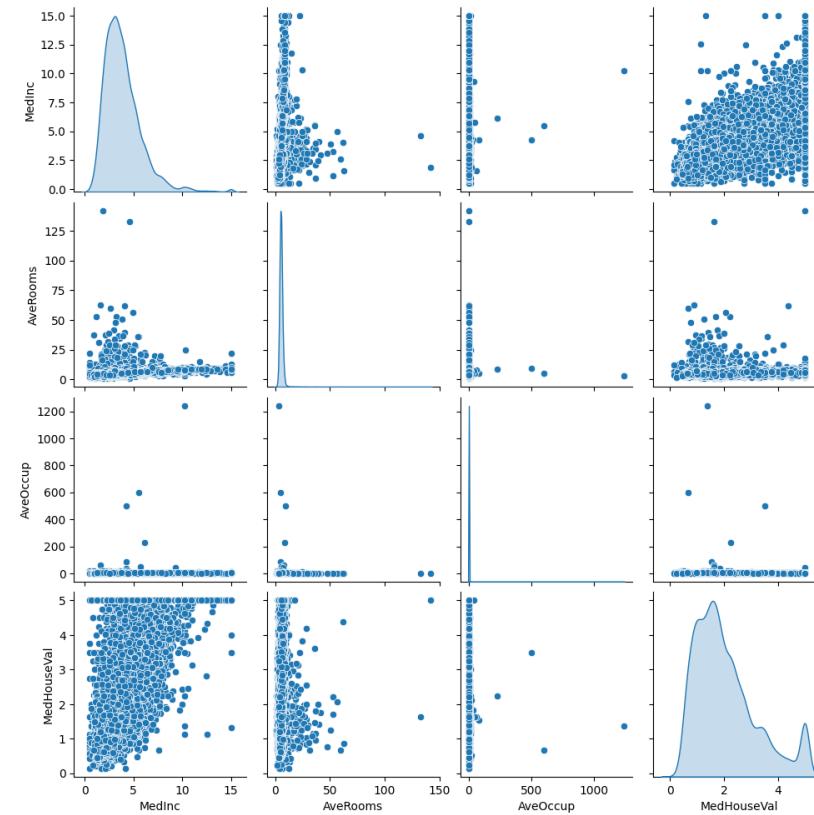
分類	指標	特徴・意味	確認ポイント
線形に強い指標	ピアソン相関	相関関係(一緒に増減する度合い)	値が ± 1 に近いほど、強い線形関係を示す
	スピアマン順位相関	値の大小関係の一致度合い	値が ± 1 に近いほど、強い順序関係を示す
非線形に強い指標	相互情報量 (MI)	依存関係の情報量を確率論で導出	値が大きいほど、依存性が高い
	MIC	データ間の様々な関係性(線形・非線形・周期など)を表す	値が1に近いほど、何らかの規則的関係を持つ

EDAによる特徴量間の相関チェック例

ピアソン相関のヒートマップ図



特徴量間のペアプロット散布図



データ前処理

- 生データを機械学習に使える形に成形するプロセス

主な処理

- データ分割：データ全体を訓練データとテストデータに分割
- 欠損値削除：明らかな不当データの削除
- 欠損値補完：平均／中央値補完、前方補完など
- 外れ値処理：除外、ワインザー化
- スケーリング：標準化、正規化
- カテゴリ変数のエンコーディング：One-Hotエンコーディング、ラベルエンコーディング

特徴量エンジニアリング

- 機械学習させやすい情報表現で訓練データを作成するプロセス

代表的な手法

- 既存特徴量の組み合わせ：例) 面積 × 部屋数 を新たな特徴量に追加
- 集約特徴量：グループごとの平均、最大、標準偏差
- 次元削減：PCA、LDA など
- 時系列特徴量：移動平均、ラグ特徴量

ポイント

- 高い学習効果が見込まれる特徴量を、少ない数でシンプルに作成すると成功しやすいです。

EDA ~ データ前処理 ~ 特徴量エンジニアリング

という一連のプロセスは、機械学習の流れの中で最重要のプロセスと言えます。

全プロセスに要する時間の80%以上を占める、と言われるほどです。

学習タスクと代表的なアルゴリズム

- 機械学習の手法は**教師あり学習**と**教師なし学習**に分けられる
 - 例外的に、それの中間ともいえる強化学習なども存在
- ここまでご紹介した回帰、分類、クラスタリングなどは**学習タスク**と呼び、教師あり学習・教師なし学習のいずれかに属する

教師あり学習

入力と正解ラベル(答え)があるデータで学習

回帰

過去のデータからパターンを学習し、未知の結果を予測

分類

過去のデータから分類方法を学習し、未知のデータを分類

教師なし学習

正解ラベル(答え)がないデータを構造化

クラスタリング

未知のデータを、「類似性」をもとにグループ化

次元削減

データを表現するために必要な値の数を削減

強化学習(ご参考)

与えられる報酬を最大化するよう行動を選択

ロボティクス

ロボットの動作を調整しながら状態を維持

ゲームAI

状態に基づいて次の最適な行動を選択

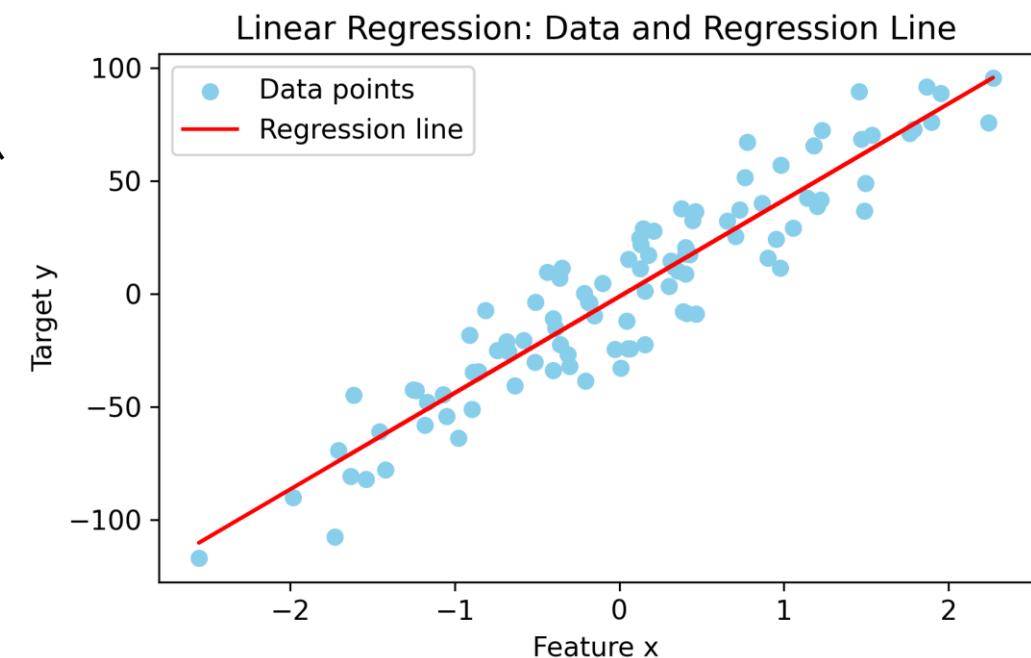
回帰

過去のデータからパターンを学習し、未知の結果を予測する手法

- 主なアルゴリズム：線形回帰、Lasso回帰 など
- 用途例：不動産価格予測、電力需要予測、為替レート予測

例：線形回帰

- 説明変数 x と目的変数 y の関係を「直線・平面」で近似
- 一次方程式： $y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$
- 目的：誤差(二乗誤差)を最小にする係数(w_n)を求める



分類 (1)

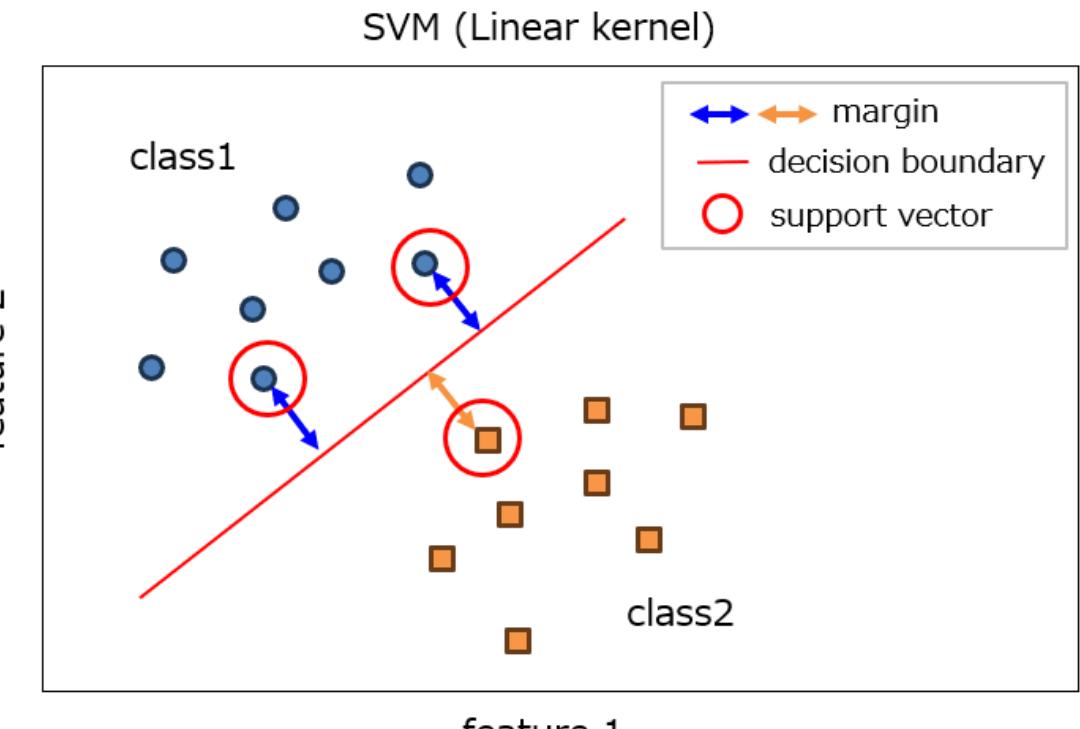
過去のデータから分類方法を学習し、未知のデータを分類する手法

- 主なアルゴリズム：サポートベクターマシン(SVM)、ランダムフォレストなど
- 用途例：医療診断、クレカ不正利用検知、迷惑メールフィルタ

例：サポートベクターマシン

- クラスを線や面(決定境界)で分割
- 決定境界に最も近い点(サポートベクター)までの距離(マージン)を最大化
- 決定境界をはみ出すデータの許容度をハイパープラメータ $\ast C$ (正則化パラメータ)で決定

※「ハイパープラメータ」については3日目の講義で解説します。

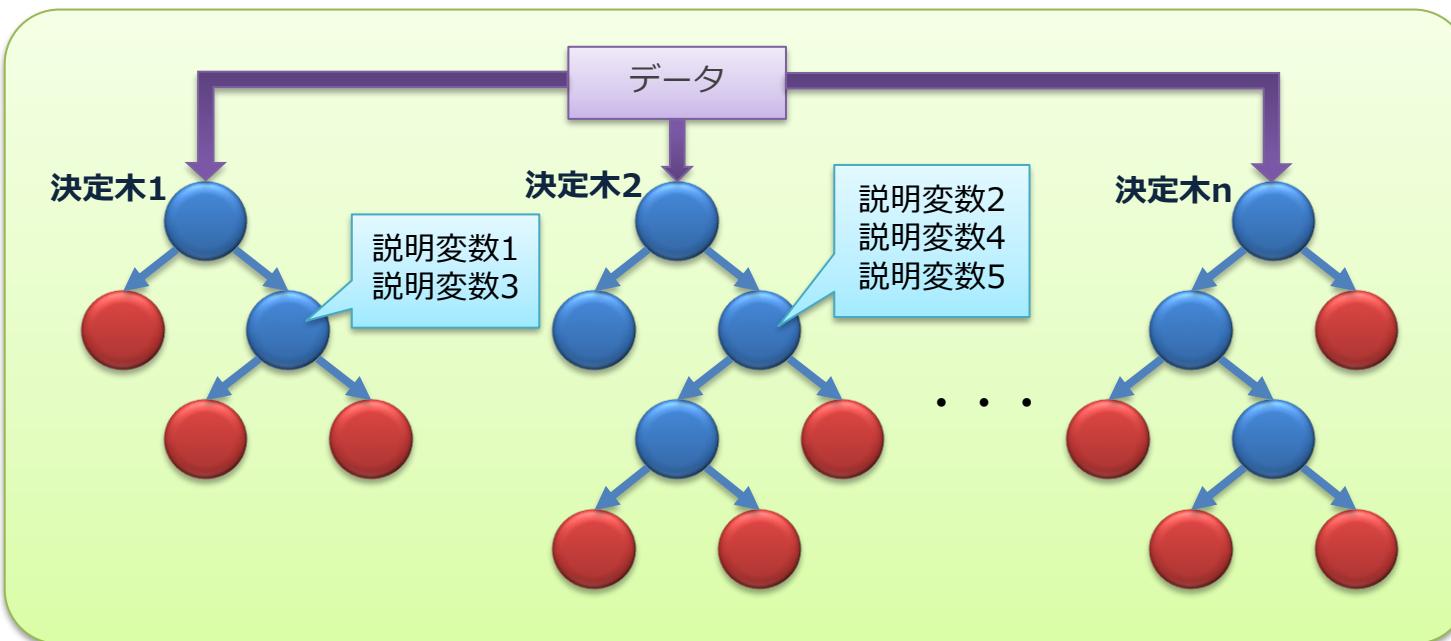


分類 (2)

例：ランダムフォレスト

多数の「決定木」を組み合わせたアンサンブル学習

- それぞれの決定木は、データと説明変数をランダムに選び、末端に向かって分岐をたどっていく
- 最終結果は、多数決(分類)や平均(回帰)などの方式で決定



注意：
サポートベクターマシンやランダムフォレストは、回帰タスクでも利用できます。これらに限らず、各アルゴリズムは、特定のタスク専用とは限らないので注意してください。

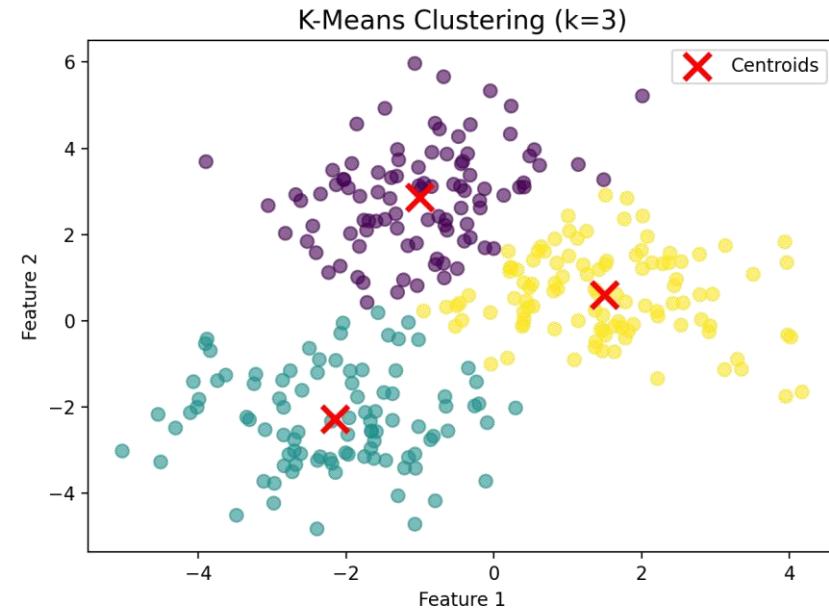
クラスタリング

未知のデータを、「類似性」をもとにグループ化する手法

- 主なアルゴリズム : k-means、スペクトラルクラスタリング
- 実用例 : 顧客セグメンテーション、交通パターン分析、楽曲グルーピング

例 : k-means

- データをk個のグループ(クラスタ)に分ける
- 各クラスタの中心(Centroid)との距離を最小化
- ステップ：
 1. データをランダムにクラスタに分けする
 2. 各クラスタの中心を求める
 3. 求めた中心に最も近いデータを、そのクラスタに割り当てなおす
 4. 中心が変化しなくなる、または変化量が一定値以下となるまで、2.と3.を繰り返す



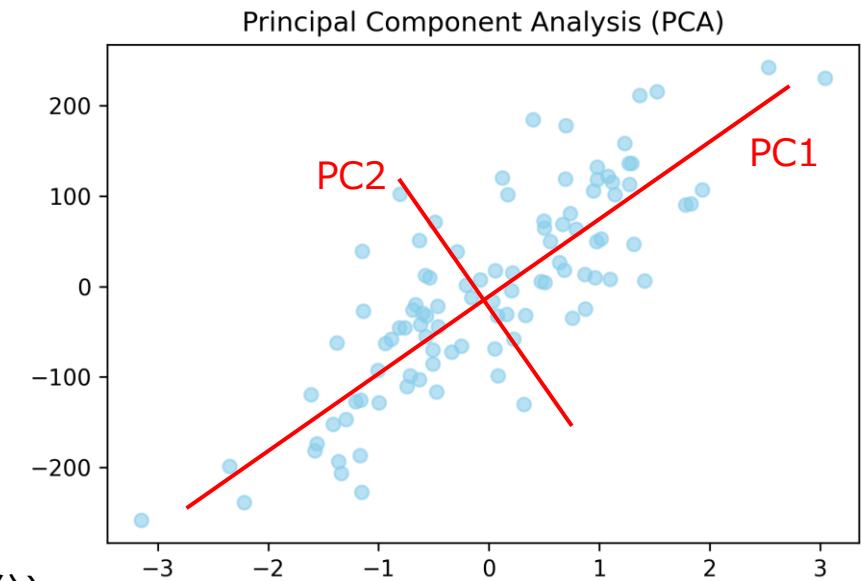
次元削減

データを表現するために必要な特徴量の数を削減する手法

- ・ データ圧縮、可視化、ノイズ除去などに有効
- ・ 主なアルゴリズム：主成分分析(PCA)、t-SNE
- ・ 実用例：遺伝子発現データ・センサーデータ異常検知などの前処理

例：主成分分析(PCA)

- ・ データの分散が最大となる方向(主成分軸)を見つける
- ・ 高次元の特徴を少数の主成分にまとめる
- ・ ステップ：
 1. すべてのデータの重心を求める
 2. 重心からデータの分散が最大となる方向を見つける(第1主成分)
 3. 第1主成分と直角(直交)で、次に分散が最大となる方向を見つける(第2主成分)
 4. 上位の主成分のみを、特徴量として採用する



検証・評価（1）

- 検証・評価のプロセスでは、学習済みモデルにテストデータを入力してタスクを実行し、タスクの結果と正解ラベルとを比較する
- 学習タスクやアルゴリズムごとに、様々な評価指標がある

タスク	評価指標の例	評価ポイント
回帰	MSE RMSE R2スコア（決定係数）	予測値と実際の値のズレの大きさ
分類	正解率（Accuracy） 適合率（Precision） 再現率（Recall） F1スコア 混同行列	当たり具合、誤りの傾向
クラスタリング	ARI（Adjusted Rand Index） シルエットスコア	グループのまとまり・分離の良さ
次元削減	累積寄与率 分離の見やすさ（例：クラスタの可視化）	元の情報をどれだけ保ったまま小さくできたか

検証・評価 (2)

- このあとのハンズオンでも登場する指標について、評価方法を次に示す

タスク	評価指標の例	説明	評価方法
回帰	MSE (Mean Squared Error)	平均二乗誤差 予測値と実測値の差の二乗平均	小さいほど良い
	R2スコア (決定係数)	モデルがどの程度うまくデータのばらつきを説明できているか	1に近いほど良い
分類	正解率 (Accuracy)	全データのうち正しく分類できた割合	1に近いほど良い
	混同行列 (Confusion Matrix)	予測結果と正解ラベルを表で比較	一致数が多いほど良い
クラスタリング	ARI (Adjusted Rand Index)	真のラベルとクラスタ結果の一致度を評価 (ラベルがある場合のみ適用可)	1に近いほど良い
	シルエットスコア	各データが自分のクラスタにどれだけ適しているか (ラベル不要)	1に近いほどクラスタが明確
次元削減	累積寄与率	選択した主成分の寄与率の合計	1に近いほど良い

検証・評価（3）

指標を選ぶコツ

- ひとつの数値で判断しない
 - 単に「正解率が高い」だけでは不十分な場合がある
 - 例：健康／病気を分類する学習タスクにおいて、病気の人が1%しかいないデータでは、常に「健康」と予測すれば正解率は99%になってしまう・・・（不均衡データ）
- ビジネスや目的の背景を意識する
 - 誤検出と見逃し、どちらが困るか？など

機械学習のプロセスは「1回通して終わり！」ではありません。

評価結果から得られた見解をフィードバックしてEDAに戻り、目標とする結果が得られる学習ができるまで何度も何度も、プロセスを繰り返すことになります。

よくある落とし穴

- 機械学習で陥りがちな、よくある落とし穴と主な対策をまとめます

落とし穴	説明	主な対策
過学習 (オーバーフィットィング)	<ul style="list-style-type: none">学習結果が訓練データに特化しすぎた状態訓練データでは完璧に近い結果が出せても、未知のテストデータではボロボロの結果に	<ul style="list-style-type: none">十分な量のデータを準備特徴量や学習アルゴリズムの単純化交差検証でのチェック
データリーク	<ul style="list-style-type: none">テストデータの情報を学習に使ってしまうことテストデータによる検証が、実態と乖離した良好な結果に見えててしまう（具体例を後述）	<ul style="list-style-type: none">データ分割後のテストデータのみ前処理を行う説明変数が未来情報を含まないかをチェック
データ不足／ データバイアス	<ul style="list-style-type: none">訓練データが極端に不足していたり、偏ったりしている状態外れ値に過剰に反応してしまうなど、安定した結果が得られない	<ul style="list-style-type: none">十分な量のデータを準備EDAで分布の偏りをチェック
多重共線性	<ul style="list-style-type: none">説明変数同士が強い相関性を持った状態どの変数が本当に目的変数を説明できているのか判断しづらく、特に回帰での学習結果が不安定になりやすい	<ul style="list-style-type: none">説明変数の集約や削減リッジ回帰やLasso回帰などの機械学習モデルを選択

データリーケの具体例

- データリーケのよくある例として、データ前処理でのスケーリング手順誤りがある

誤った手順

- データ全体の統計情報(最大値／最小値、平均／標準偏差など)を計算
- データ全体を、計算した統計情報でスケーリング
- データ全体を、訓練データとテストデータに分割

学習時には本来知りえない、テストデータの分布情報が混入してしまう！！

正しい手順

- データ全体を、訓練データとテストデータに分割
- 訓練データの統計情報を計算
- 計算した統計情報で、訓練データとテストデータをスケーリング

scikit-learnとは

- Pythonで最も広く使われる機械学習ライブラリのひとつ

特徴：

- シンプルなAPI設計 (fit → predict)
- 回帰・分類・クラスタリング・次元削減・前処理・評価指標など、機能が一通り揃っている
- 教育用にも実務用にも適している

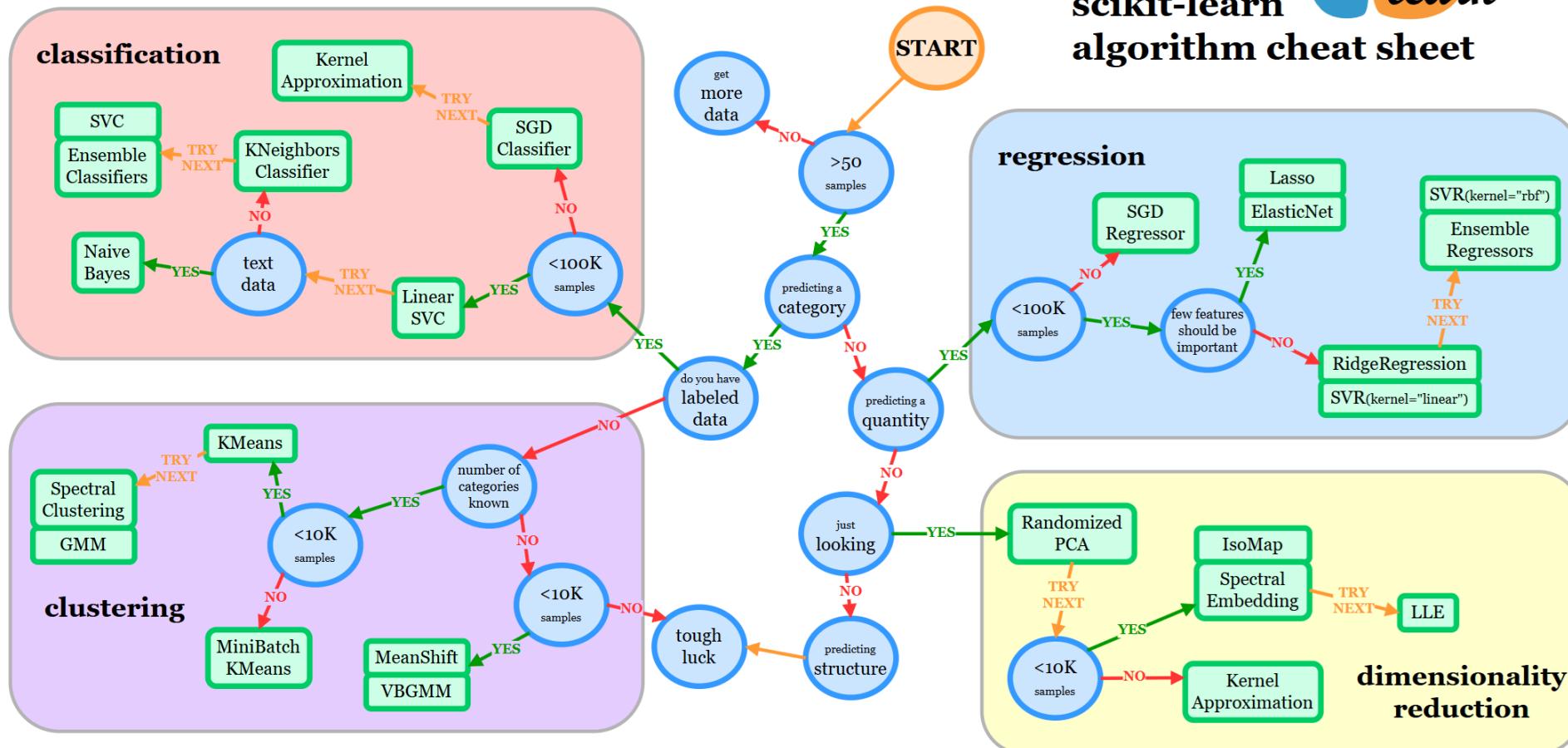
基本的なコード例：

- たったこれだけのコードで線形回帰による学習・予測が完了！

```
from sklearn.linear_model import LinearRegression  
  
model = LinearRegression()  
model.fit(訓練データ, 正解ラベル)  
pred = model.predict(テストデータ)
```

アルゴリズム選択のご参考

- scikit-learnのアルゴリズムチートシートが便利です。



出展：https://scikit-learn.org/stable/machine_learning_map.html

ハンズオンにチャレンジ

このあとは、scikit-learnによる機械学習にチャレンジしていただきます！！

Google Colaboratory URL :

https://colab.research.google.com/drive/1wH3hONX0daFu2abC7V_eeLD6ZyrVbPk7?usp=sharing