現場で使える機械学習・データ分析
基礎講座（DAY1）

SkillUP AI

本講座の方針

- 本講座は、予習、対面講義、通し課題で構成されます
- 予習では、予習動画を視聴し、Notebook演習に取り組んで下さい
- 対面講義では、予習に関する質疑応答とグループワークを行います
- 通し課題では、DAY1-DAY4までを通して、1つの課題に取り組んで頂きます
- 内容が非常に高度ですので、毎回必ず予習してきてください

本講座で必要になるPC環境

- 本講座では、Jupyter Notebook(言語はPython)を用います
- Jupyter Notebookは、Anacondaというソフトをインストールすることで利用できるようになります
- 対面講義では、Anaconda3系(バージョンは別途連絡)がインストールされていることを前提に講義を進めますので、事前にインストールをしておいてください
- その他必要なライブラリは、0.preparation.ipynbを参考に事前にインストールしておいてください

参考図書（全日程分．おおよそ日程の進行順に記載しています）

① 『ITエンジニアのための機械学習理論入門』(中井悦司、技術評論社)
② 『はじめてのパターン認識』(平井有三 、森北出版)
③ 『わかりやすいパターン認識』
（石井健一郎、前田英作、上田修功、オーム社）
④ 『機械学習　-データを読み解くアルゴリズムの技法-』
（Peter Flach 著、 竹村 彰通ら訳、朝倉書店）
⑤ 『深層学習 Deep Learning』 （人工知能学会、近代科学社）
⑥ 『これならわかる深層学習入門』 （瀧 雅人、講談社）
⑦ 『深層学習』
（Ian Goodfellow、 Yoshua Bengioら著、KADOKAWA）
⑧ 『続・わかりやすいパターン認識―教師なし学習入門―』
（石井健一郎、上田修功、オーム社）

本講座の構成

DAY1	DAY2
• 機械学習概論 　• 人工知能とは 　• 機械学習とは 　• 機械学習アルゴリズムの実装とワークフロー 　• 機械学習アルゴリズム概観 • 教師あり学習の基礎 　• 線形回帰 　• ロジスティック回帰 　• 多変量モデルへの拡張 • モデルの評価指標 　• 回帰問題（MAE/MSE/RMSE） 　• 分類問題（精度/適合率/再現率/F1-score）	• モデルの検証・正則化 　• 訓練誤差と汎化誤差 　• 過学習 　• 正則化（L2/L1） 　• ホールドアウト法・交差検証法 • 前処理 　• 正規化 / 標準化 　• 無相関化 / 白色化 • 教師あり学習の発展的トピック 　• サポートベクターマシン

本講座の構成

DAY3	DAY4
• 前処理 　• 特徴選択 • 教師あり学習の発展的トピック 　• 木モデル 　（決定木・ランダムフォレスト） 　• ニューラルネットワーク	• 教師あり学習の発展的トピック 　• 深層学習 　• k-最近傍法 • 教師なし学習 　• クラスタリング 　• 特徴抽出・次元削減 • モデルの改善 　• ハイパーパラメータ最適化

DAY1の目標とDAY4までの流れ

- DAY1では、講義内容をもとに、精度は気にせず、とにかく教師あり機械学習モデルを構築できるところまでを目指します
- DAY2以降で、精度を向上させる方法や様々なアルゴリズムを紹介し、教師あり機械学習モデル構築時の手札を増やしていきます
- DAY4で通し課題の最終発表の時間を設けているので、最初のモデルから何を変えてどの程度精度が向上したかを発表して頂きます
- DAY4では、教師なし学習や深層学習などについても触れる予定です

DAY1の目次

- 機械学習概論
  - 人工知能とは
  - 機械学習とは
  - 機械学習アルゴリズムの実装とワークフロー
  - 機械学習アルゴリズム概観
- 教師あり学習の基礎
  - 線形回帰
  - グループワーク1
  - ロジスティック回帰
  - 多変量モデルへの拡張
- モデルの評価指標
  - 回帰問題（MAE/MSE/RMSE）
  - 分類問題（精度/適合率/再現率/F1-score）
- グループワーク2
- 通し課題に関する説明と質疑

機械学習概論

1. 人工知能とは
2. 機械学習とは
3. 機械学習アルゴリズムの実装とワークフロー
4. 機械学習アルゴリズム概観

人工知能とは？ 機械学習とは？ 深層学習とは？

- 人工知能とは？SFの世界？
- 機械学習とは？機械が学習するってどういうこと？
- 深層学習とは？何が深いの？

人工知能（Artificial Intelligence）の定義

ジョン・マッカーシー*
(John McCarthy、 1927-2011)

Q. 人工知能とは何でしょうか？

知的な機械、特に、
知的なコンピュータプログラムを作る科学と技術です[1]

Q. 知能とは何でしょうか？

実際の目標を達成する能力の計算的な部分です
人間、動物、そして機械には、種類や水準が
さまざまな知能があります[1]

*：人工知能研究の第一人者．「Artificial Intelligence」という用語を初めて公の場で
使用した人物であり、プログラミング言語LISPの開発者でもある

[1] 人工知能学会：人工知能のFAQ、https://www.ai-gakkai.or.jp/whatsai/AIfaq.html (2018.09.12アクセス)

人工知能、機械学習、深層学習（ディープラーニング）のバズワード整理

「人工知能 ＝ 機械学習 ＝ 深層学習」ではないので要注意！！

- 機械学習→人工知能を実現するための技術領域のひとつ
- 深層学習（ディープラーニング）→機械学習の方法論のひとつ

人工知能（AI）

- 機械学習
  - 深層学習
  - SVM
  - 線形回帰
  - クラスタ分析
  - 決定木
  - etc…
- エキスパートシステム
- 探索アルゴリズム
- etc…

機械学習（Machine Learning）の定義

明示的にプログラムしなくても学習する能力を
コンピュータに与える研究分野[2]のこと

具体的には…

コンピュータプログラムが、あるタスクにおいて、
用意されたデータを使い、性能の評価値を向上させること

アーサー・リー・サミュエル*
（Arthur Lee Samuel、 1901-1990）

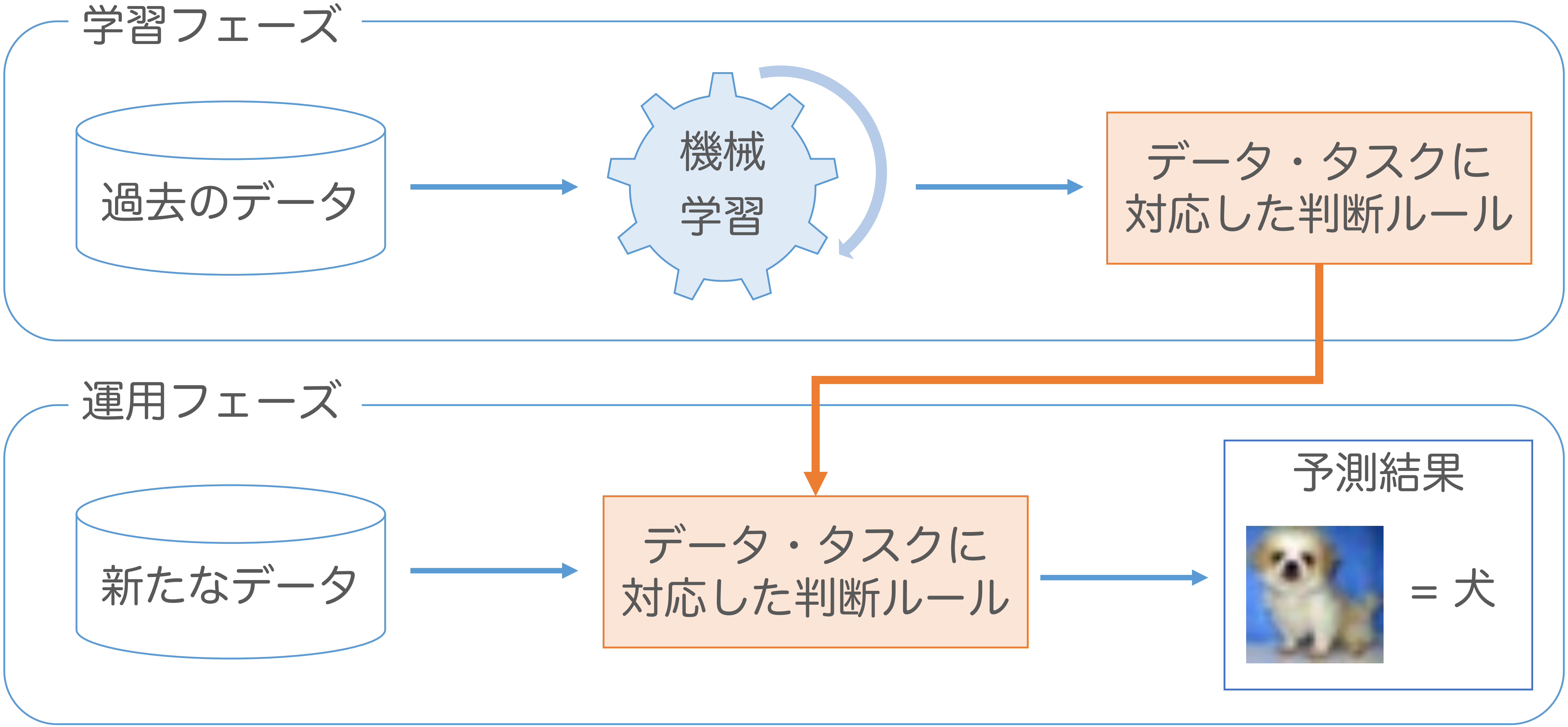
*：世界初の学習型プログラム「Samuel Checkers-playing Program」を開発した
計算機科学者．ハッシュテーブルの考案者でもある．

[2] Wikipedia：機械学習、 https://ja.wikipedia.org/wiki/機械学習 (2018.09.12アクセス)

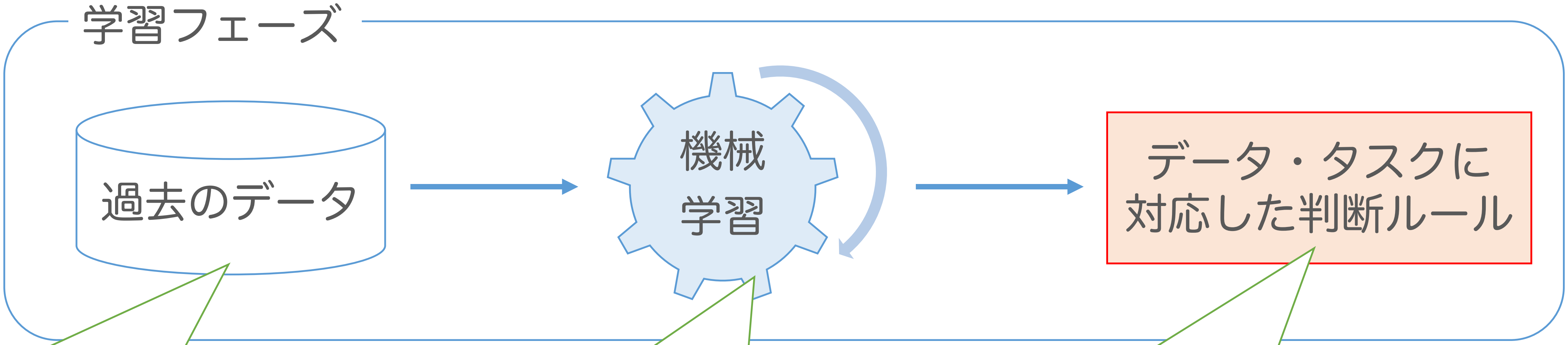
タスクとは

- 機械学習分野では、機械学習モデルが対象とする問題設定のことをタスクと呼ぶ
    - 例、株価予測、画像認識、画像物体検出、文章分類、機械翻訳

機械学習の全体像


学習フェーズ
過去のデータ
機械
学習
データ・タスクに
対応した判断ルール
運用フェーズ
新たなデータ
データ・タスクに
対応した判断ルール
予測結果
= 犬

機械学習の全体像

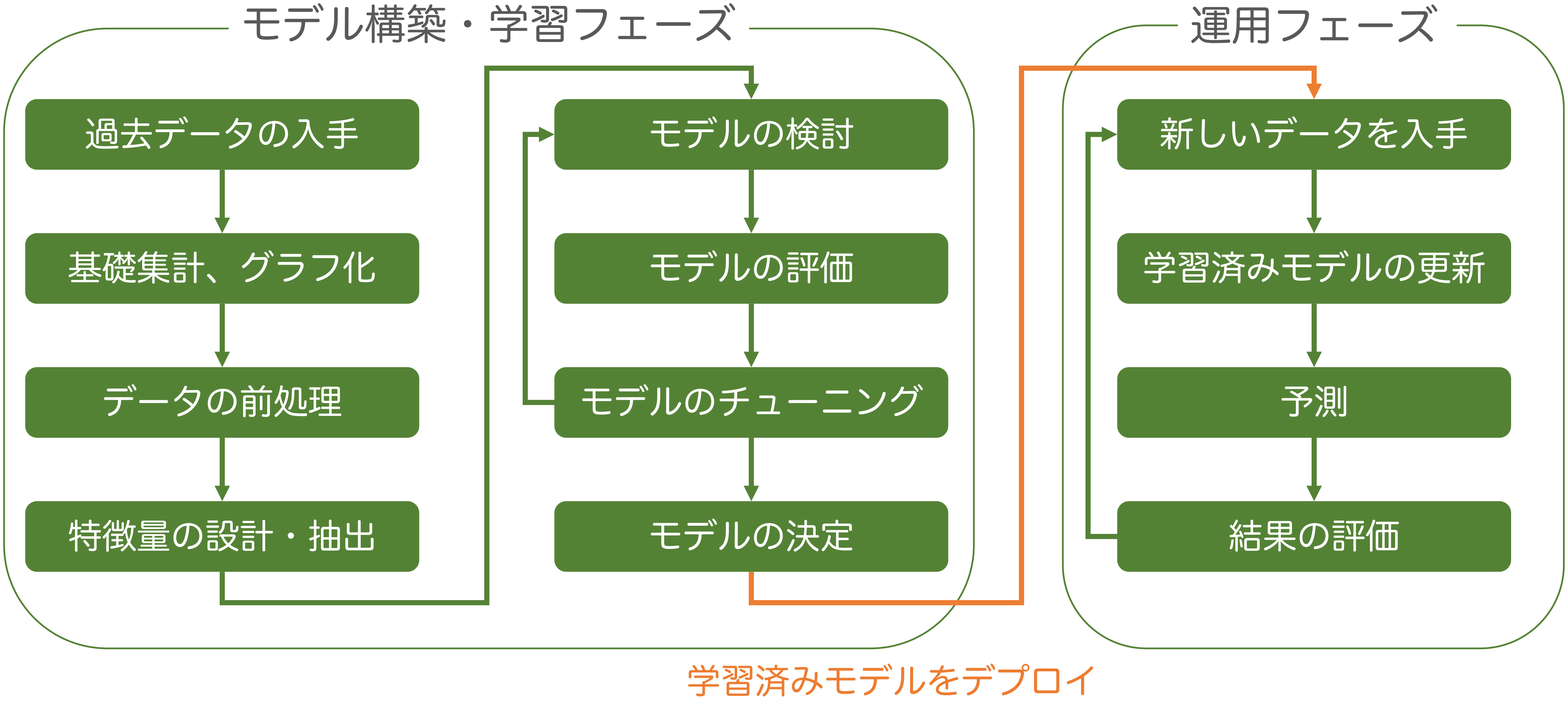


データの理解がないと
意味のあるデータが選別不可

性能向上には
理論の理解が不可欠

ビジネスの理解がないと
判断ルールを活かせない

機械学習システムの開発・運用フロー



機械学習システムの開発・運用フロー ｜ビジネス観点で重要なこと

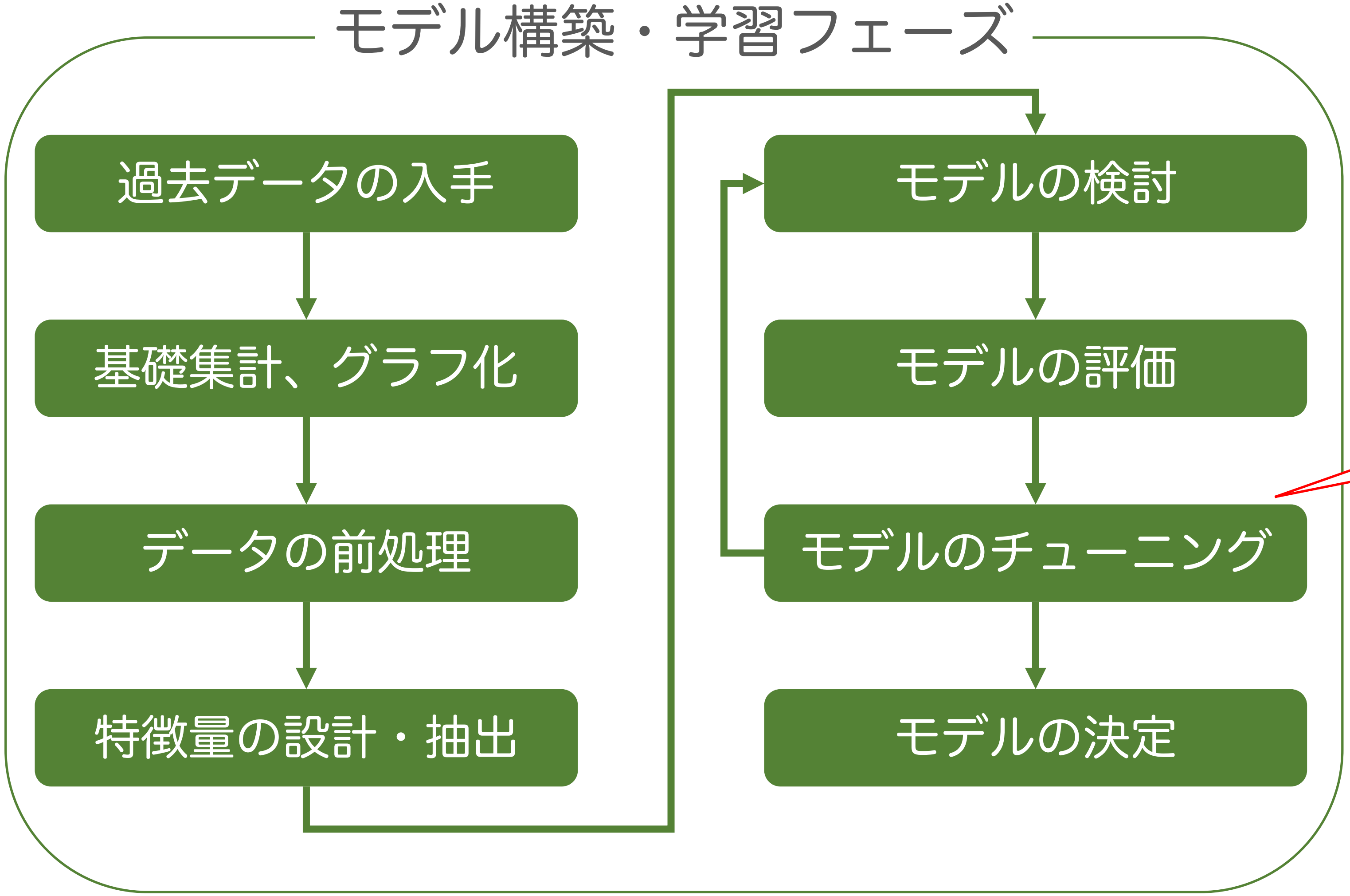
モデル構築・学習フェーズ

- 過去データの入手
- 基礎集計、グラフ化
- データの前処理
- 特徴量の設計・抽出

課題定義

- 機械学習で何を解くか？
- どれほどのビジネスバリューをもたらすか？
- その課題は本当に機械学習で解く必要がある課題か？

機械学習システムの開発・運用フロー ｜ビジネス観点で重要なこと



ROI観点でどこまで
チューニングすべきか

- わずかの改善に多大な工数を割いてはならない
- 精度1%の改善でも50%→51%より90%→91%のほうがより工数がかかる

機械学習システムを実装する際に必要な知識とスキル

機械学習アルゴリズム	基礎知識	基本的なITスキル
教師なし学習のアルゴリズム	数学記号	Python
教師あり学習のアルゴリズム	微分積分	Linux
強化学習のアルゴリズム	線形代数	bash
	確率	ssh
	統計学	MySQL
	最適化手法	json
		encoding
		git

RとPython、どっちがいいの？

	R	Python
習得の容易さ	◎ データ分析に特化しているため、記述が簡潔	○ 汎用言語のため、プログラミングの基本を理解する必要がある
応用発展性	○	◎ データ分析以外のライブラリも充実
開発環境	○ Rstudioは使いやすい	◎ Jupyter Notebookはもっと使いやすい
情報量	◎ 日本語の書籍やサンプルコードが多い	○ Stack OverFlowを見れば、大概解決する。(英語)
統計関連のライブラリ	◎ 書籍やサンプルコードが多い	○ ライブラリはあるが、使用例や情報が少ない
機械学習関連のライブラリ	◎	◎
日本語対応	○ 不便を感じることはない	△ グラフ化など、たまに不便を感じる
レポーティング	○ knitrでレポートは作成できるが、使いづらい。NotebookでRカーネルをインストールして使う方がいいかも。	◎ Notebook自体がレポートになる。柔軟にいろいろできる

主要ツールやライブラリ概観

プログラミング言語	python™
機械学習ライブラリ	scikit learn　K + TensorFlow
ベクトル・行列計算に特化したライブラリ	NumPy
計算可視化ライブラリ	matplotlib
データ処理ライブラリ	pandas
科学計算ライブラリ	SciPy

より使いこなせるようになる方は『機械学習・ディープラーニングのためのPython基礎講座』へ！

機械学習アルゴリズムの分類

教師あり学習（Supervised Learning）

- 入力とそれに対する正しい出力（＝教師信号）が学習用データとしてモデルに与えられる
- モデルは出力を教師信号に近づけるように学習

教師なし学習（Unsupervised Learning）

- 入力のみが学習用データとして、モデルに与えられる
- モデルは入力の関係性や構造をうまく表現するように学習

強化学習（Reinforcement Learning）

- 学習用データは与えられず、試作錯誤を繰り返すことによって学習していく
- 学習用データを与えるかわりに、行動の良さを評価できる報酬を与える
- モデルは累積報酬を最大化するように学習

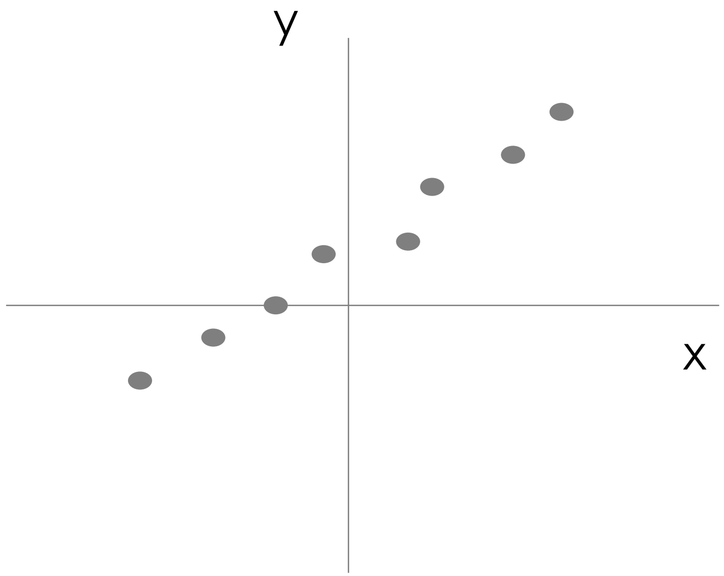
※モデル：データを説明する判断ルール（関数や確率分布）のこと

教師あり学習の代表的なアルゴリズム

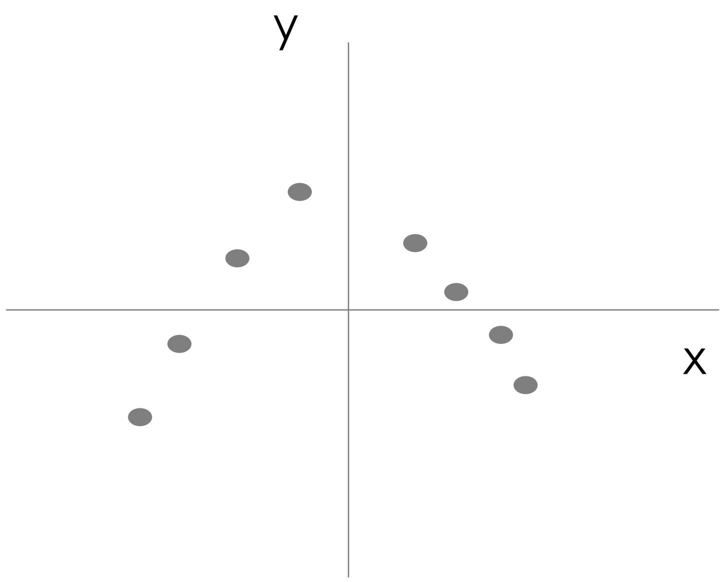
↓ 詳しくはDAY2にて！

アルゴリズム	目的	線形/非線形	正規化、標準化
線形回帰	回帰	線形	必要
ロジスティック回帰	回帰、分類	線形	必要
サポートベクターマシン	回帰、分類	非線形	必要
k近傍法	回帰、分類	非線形	必要
決定木	回帰、分類	非線形	不要
ランダムフォレスト	回帰、分類	非線形	不要
ブースティング	回帰、分類	非線形	不要
ナイーブベイズ	分類	非線形	不要
状態空間モデル、階層ベイズモデル	回帰、分類	非線形	必要
ニューラルネットワーク(深層学習)	回帰、分類	非線形	必要

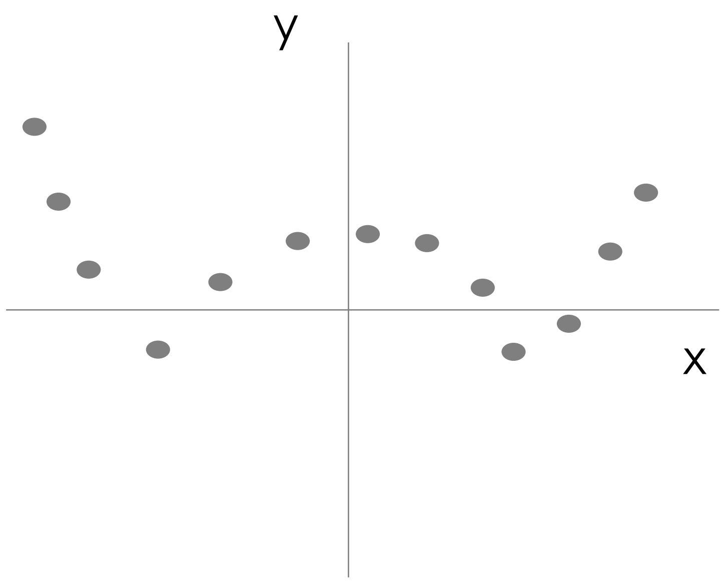
これは線形?非線形?



これは線形?非線形?



これは線形?非線形?

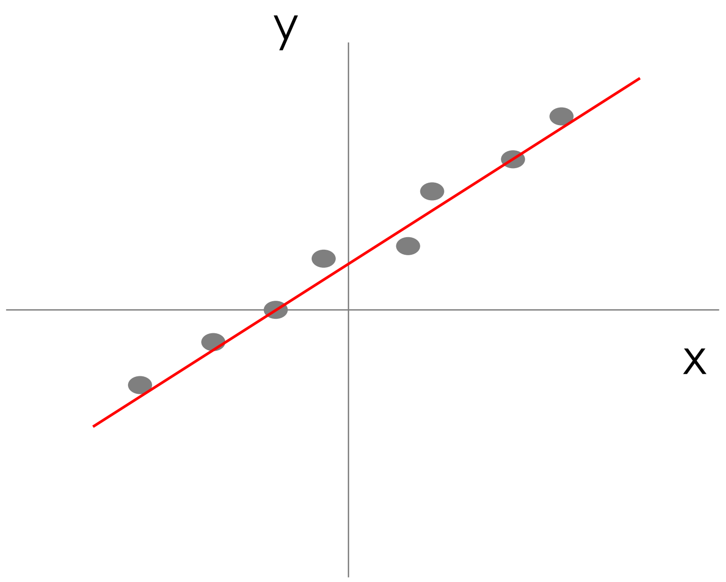


線形と非線形

「線形」「非線形」という言葉は、

データ分析でどのように使われるでしょうか？

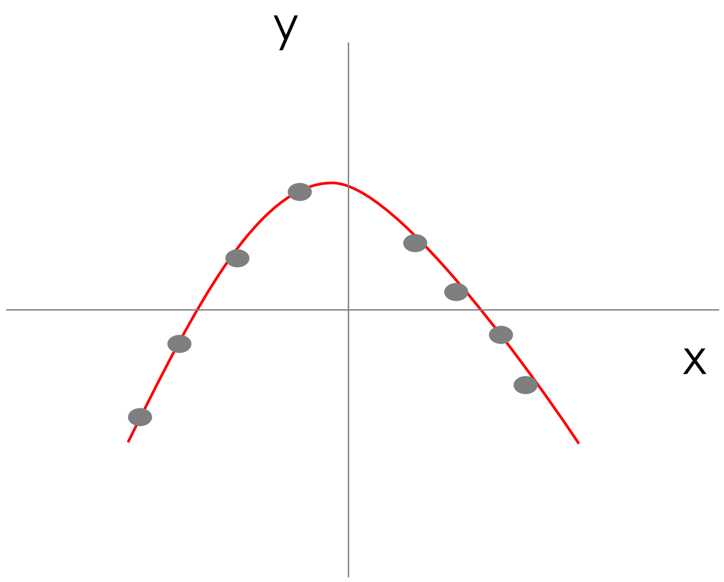
線形と非線形



$$y = ax + b$$

データでは線形の関係が見られる。
線形の手法を使って回帰モデルをつくる。

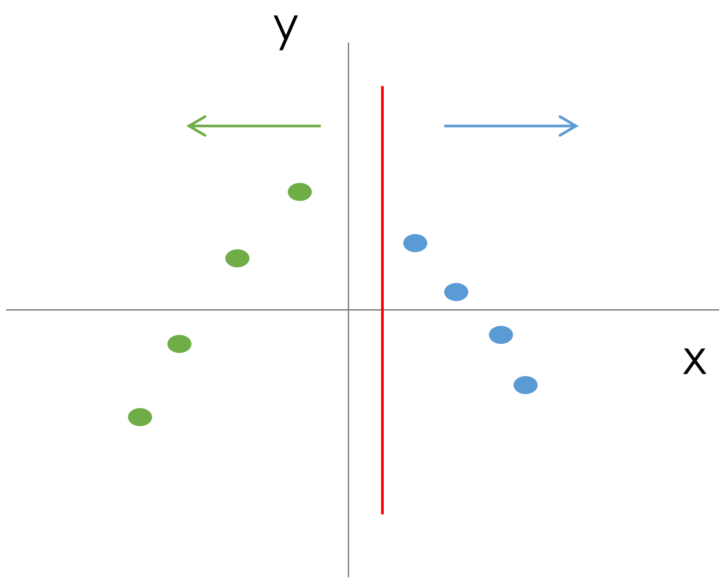
線形と非線形



$$y = ax^2 + bx + c$$

データでは非線形の関係が見られる。
線形の手法(3つの項の線形和)を使って回帰モデルをつくる。

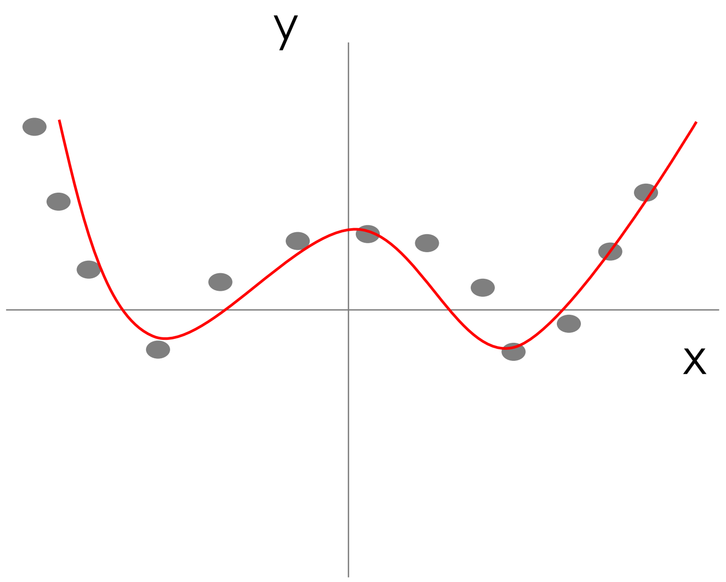
線形と非線形



$$x = a$$

データでは非線形の関係が見られる。
線形の識別境界をつくる。

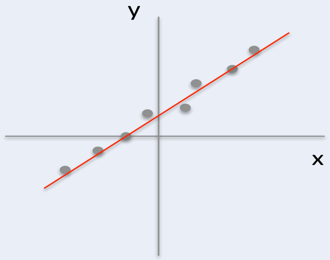
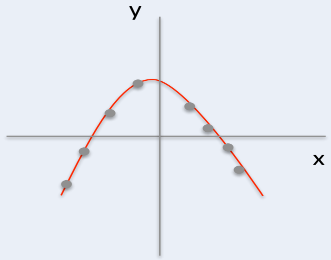
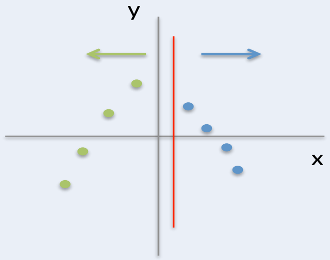
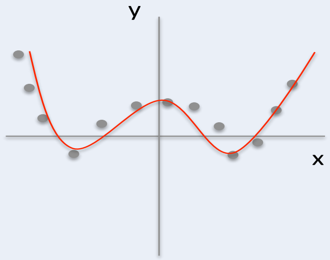
線形と非線形



SVRなど

データでは非線形の関係が見られる。
非線形の回帰モデルをつくる。

線形と非線形

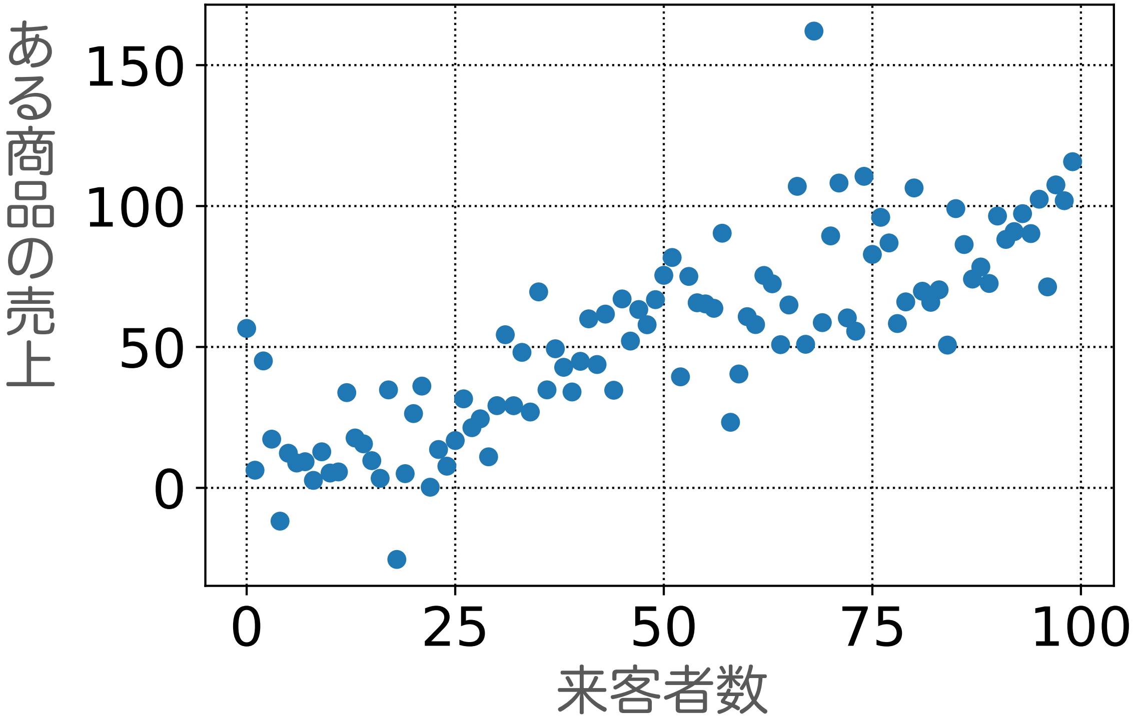
例	 $y = ax + b$	 $y = ax^2 + bx + c$	 $x = a$	 SVRなど
データ間の関係性	データでは線形の関係が見られる。	データでは非線形の関係が見られる。	データでは非線形の関係が見られる。	データでは非線形の関係が見られる。
手法	線形の手法を使って回帰モデルをつくる。	線形の手法を使って回帰モデルをつくる。	線形の識別境界を作る。	非線形の手法を使って回帰モデルをつくる

データの関係が線形か非線形かは意識しましょう

1. データが線形であれば、線形で解いた方が楽
2. データが非線形でも、線形で解ける場合がある
   - 例）$y = ax^2 + bx + c$のように多項式の線形和の関係に落とし込める場合
3. アルゴリズムを選択する際には、対象とするデータが線形か非線形かを常に意識して選ぶことが重要

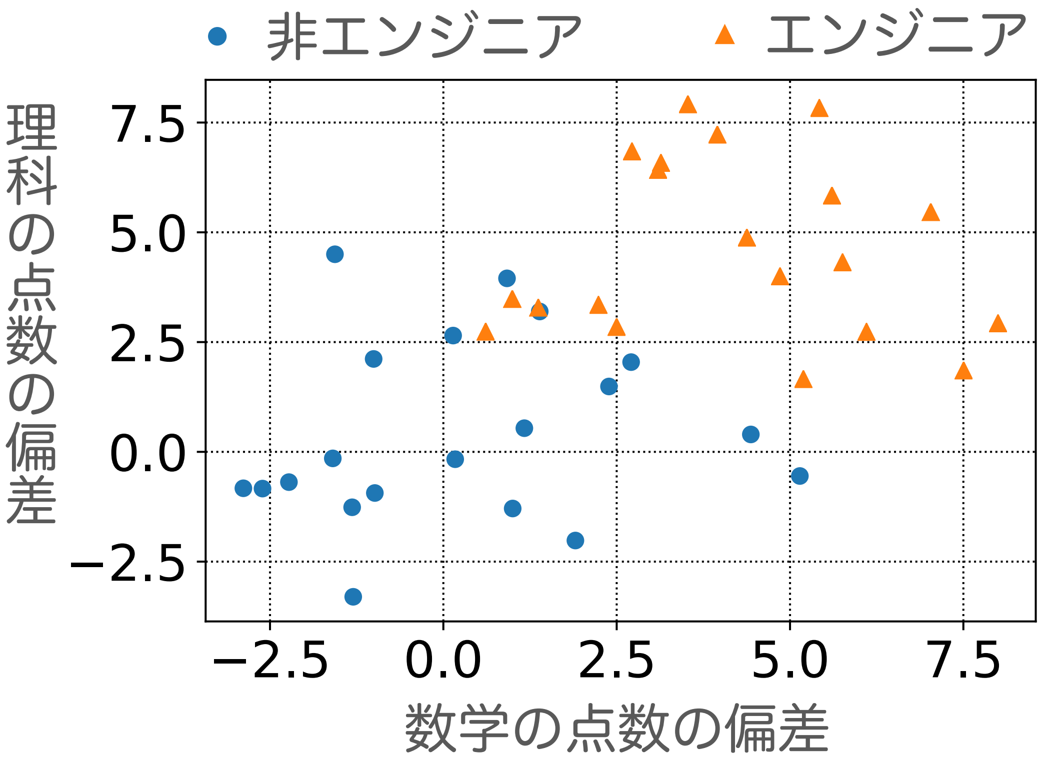
回帰（Regression）とは

- 回帰とは、数値を予測すること
  - 教師信号が連続値のケースは回帰問題に属する
  - 例）来客数からある商品の売上を予測



分類（Classification）とは

- 分類とは、カテゴリを予測すること
  - 教師信号がカテゴリを示す離散値のケースは分類問題に属する
  - 例）理科と数学の偏差から将来エンジニアになるか予測（2カテゴリ、二値分類）



回帰と分類の違い

- 回帰と分類は、目的変数が異なる

<table>
<tr><th></th><th>回帰</th><th>分類</th></tr>
<tr><td>説明変数
(特徴量、特徴ベクトル、素性、素性ベクトルとも呼ばれる)</td><td colspan="2">$\mathbf{x} = \left(x_1, x_2, \dots, x_p\right)^{\mathrm{T}} \in \mathbb{R}^p$
($\mathbf{x}$はp個の実数からなるベクトルという意味の数式)</td></tr>
<tr><td>目的変数</td><td>$y \in \mathbb{R}$
(yは実数という意味の数式)</td><td>$y \in \{0、1、\dots、C-1\}$
(yはC個のカテゴリを示す離散値という意味の数式)</td></tr>
</table>

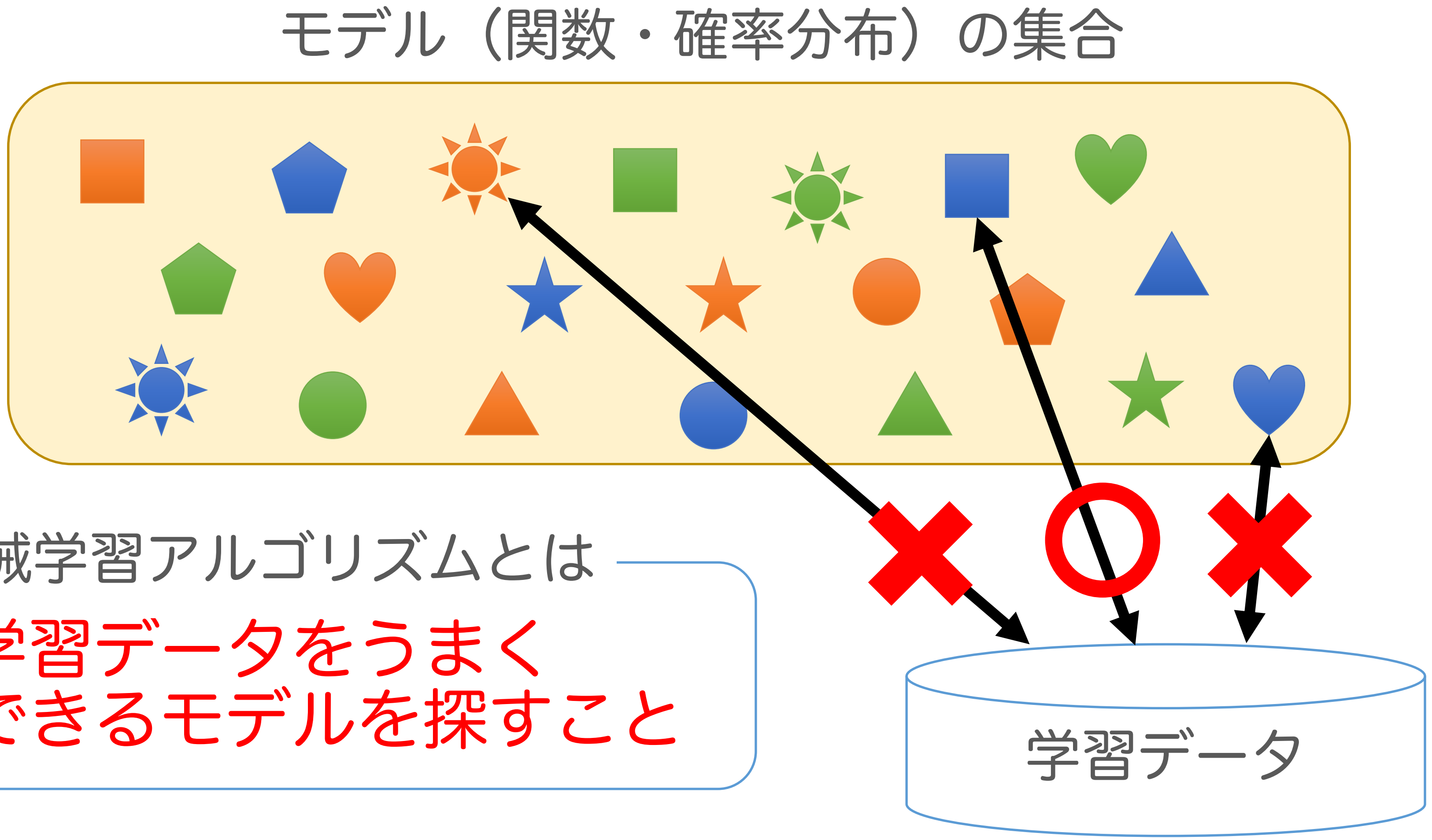
数学記号は、Appendixを参照

教師なし学習(クラスタリング)の代表的なアルゴリズム

<table>
<tr><th>日本語名</th><th>英語名</th><th>クラスタ数の
決め方</th><th>分散</th><th>手法の概要</th></tr>
<tr><td>K-平均法</td><td>K-means</td><td>分析者が決める</td><td rowspan="2">等分散なデータでないとうまく分類できない</td><td>各データとクラスタ中心の距離の総和が最小になるように分類する方法</td></tr>
<tr><td>X-means</td><td>X-means</td><td>BICが決める</td><td>BICでクラスタ数を決めるようにしたK-means</td></tr>
<tr><td>混合ガウスモデル</td><td>Gaussian Mixture Model</td><td>分析者が決める</td><td rowspan="2">等分散でないデータもうまく分類できる</td><td>ガウス分布の重ね合わせによる確率分布を求め、属するクラスタを求める方法</td></tr>
<tr><td>ディリクレ過程混合ガウスモデル</td><td>Dirichlet Process Gaussian Mixture Model</td><td>クラスタリングと同時に推定</td><td>クラスタに分けられるデータ要素の確率だけなく、クラスタ数も推定する方法

小さなクラスタとして判定され得るような微妙なデータがあると、それをクラスタリングしてしまい、真のクラスタ数よりも多いクラスタ数を推定してしまうことがある。</td></tr>
</table>

機械学習アルゴリズムのポイント



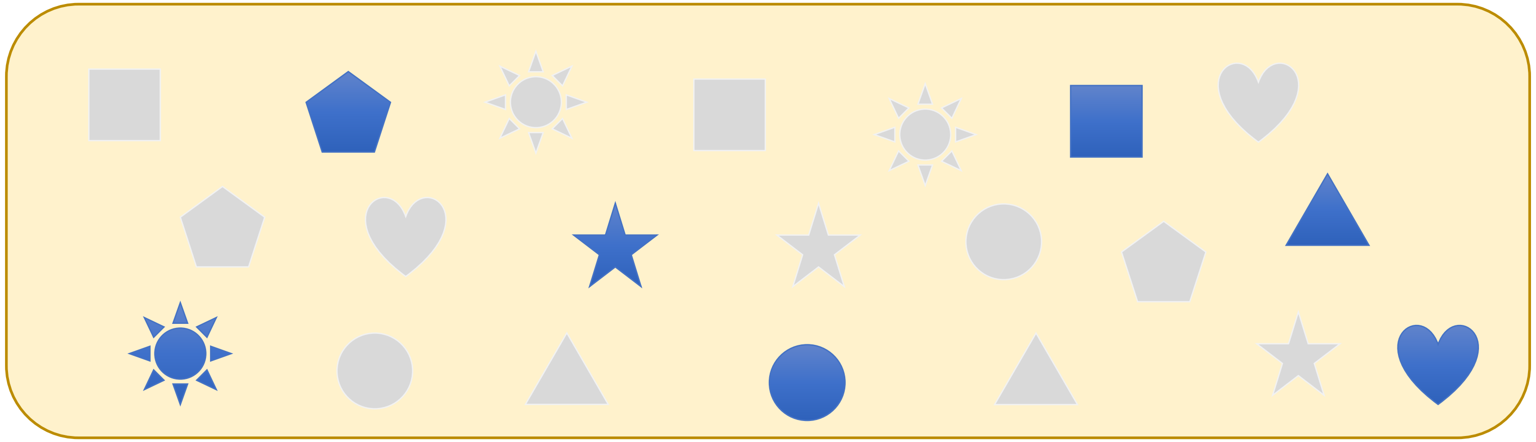
機械学習アルゴリズムとは

学習データをうまく説明できるモデルを探すこと

しかし候補となるモデルが多すぎて、最適なものを見つけるのは困難…

機械学習アルゴリズムのポイント

モデル（関数・確率分布）の集合

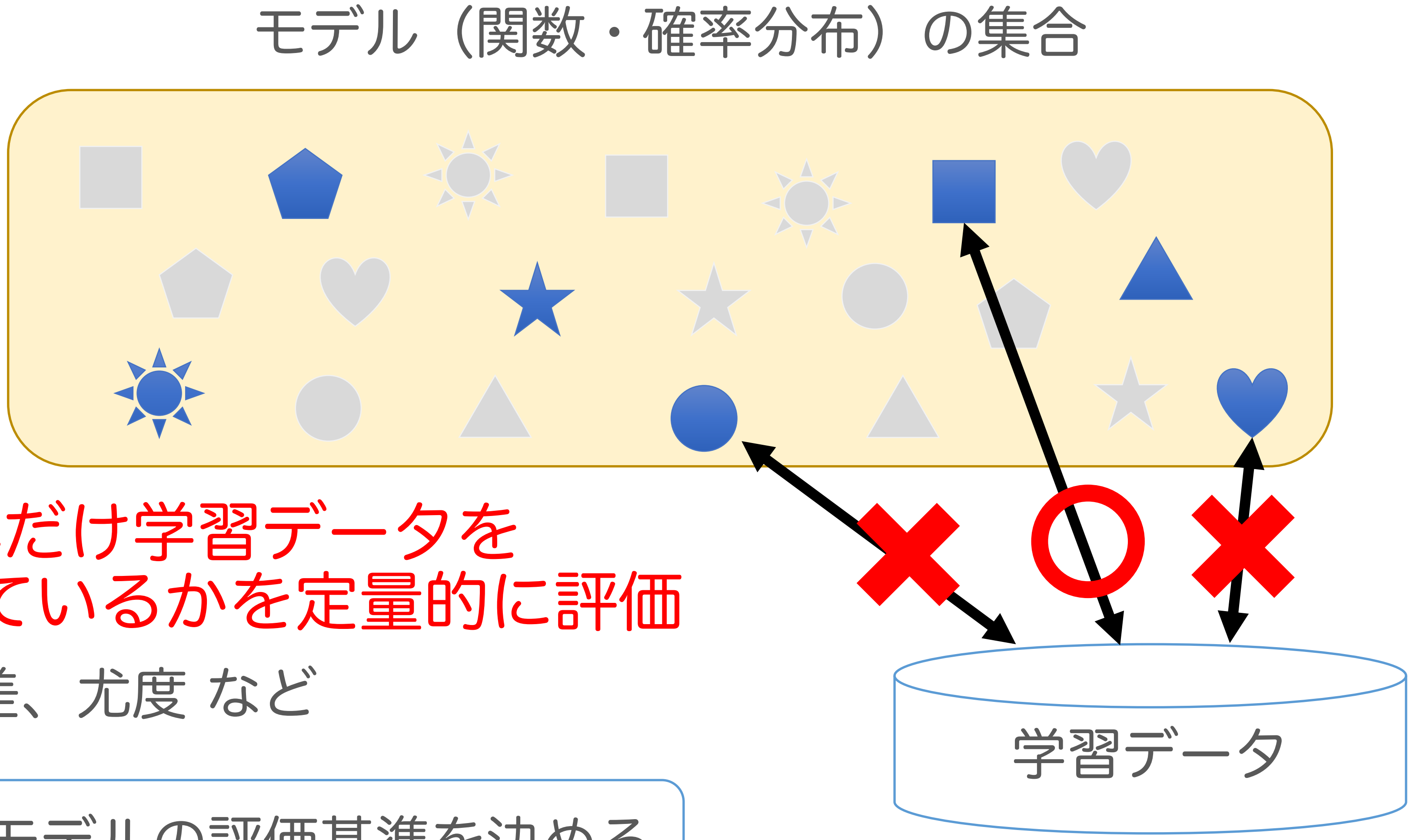


一般的にはデータの性質を考慮して、モデルの集合を制限

例）線形関数、正規分布 など

1. モデルを決める

機械学習アルゴリズムのポイント

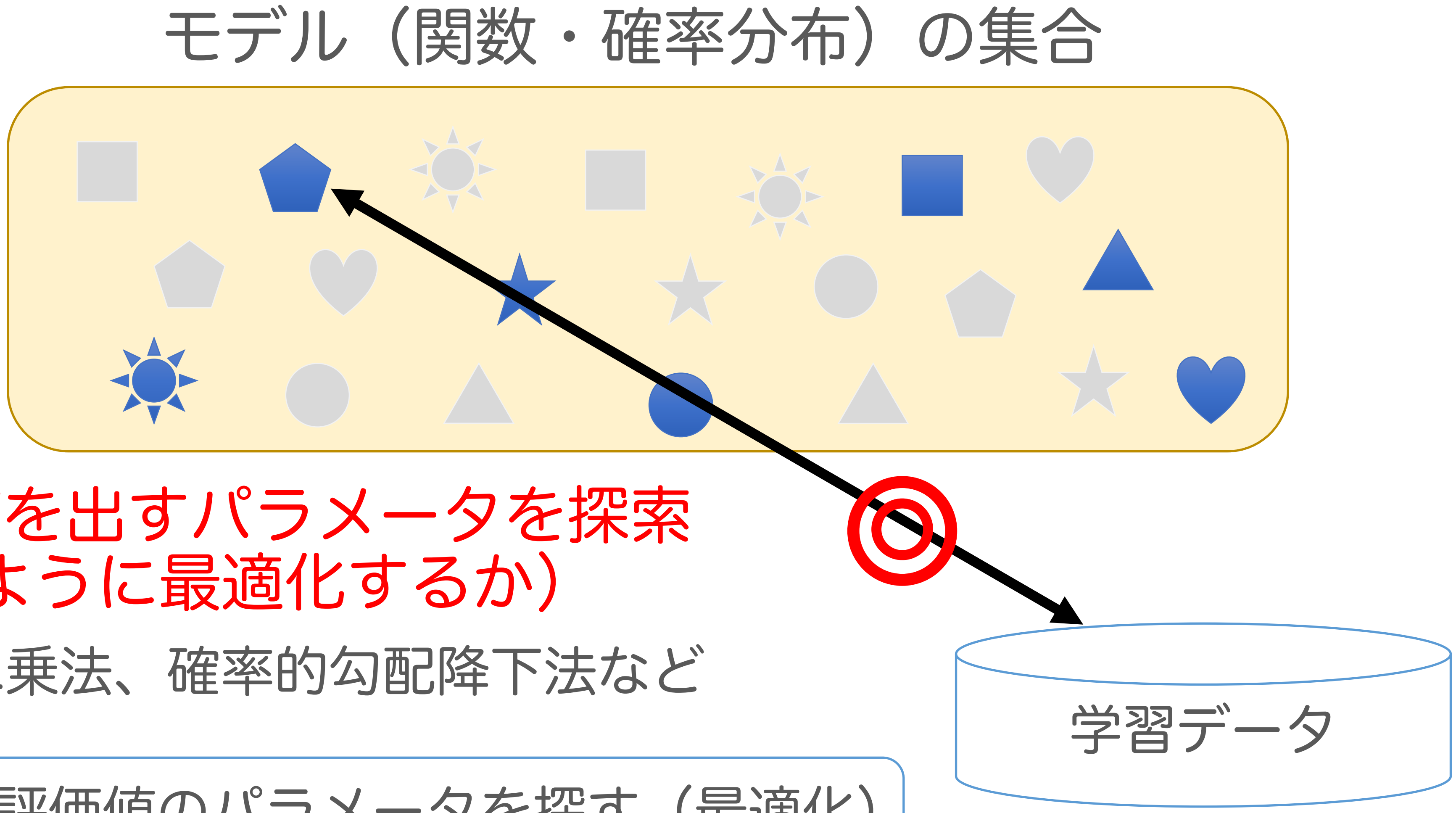


どれだけ学習データを
説明できているかを定量的に評価

例）二乗誤差、尤度 など

2. モデルの評価基準を決める

機械学習アルゴリズムのポイント



良い評価値を出すパラメータを探索
（どのように最適化するか）

例）最小二乗法、確率的勾配降下法など

3. 良い評価値のパラメータを探す（最適化）

機械学習アルゴリズムのポイント

1. データの性質に合わせてモデルを決める

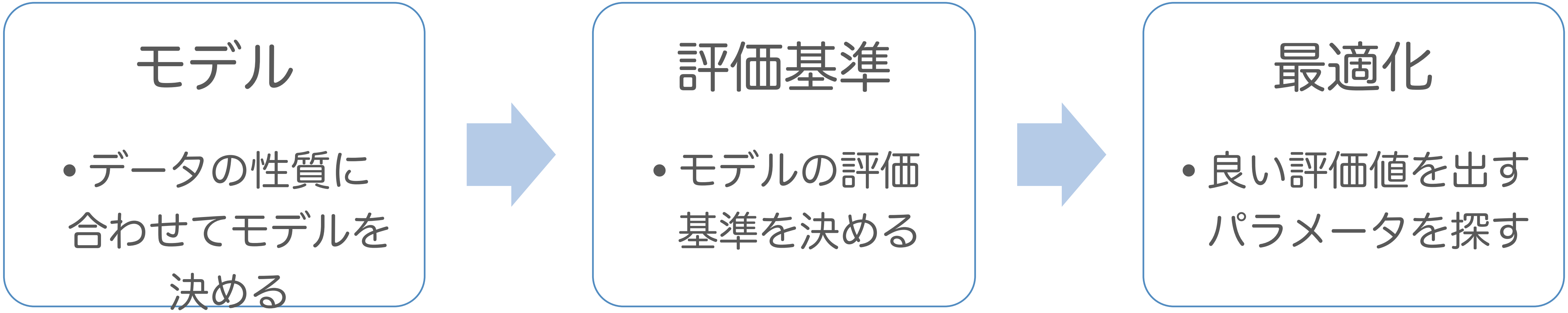
2. モデルの評価基準を決める

3. 良い評価値を出すパラメータを探す（最適化）

問題やデータに対する事前知識を用いて
1~3を設定することで機械学習アルゴリズムが完成する

機械学習アルゴリズムのポイント

- 問題やデータに対する事前知識を用いて、「モデル」「評価基準」「最適化（の方法）」を設定することで機械学習アルゴリズムが完成する

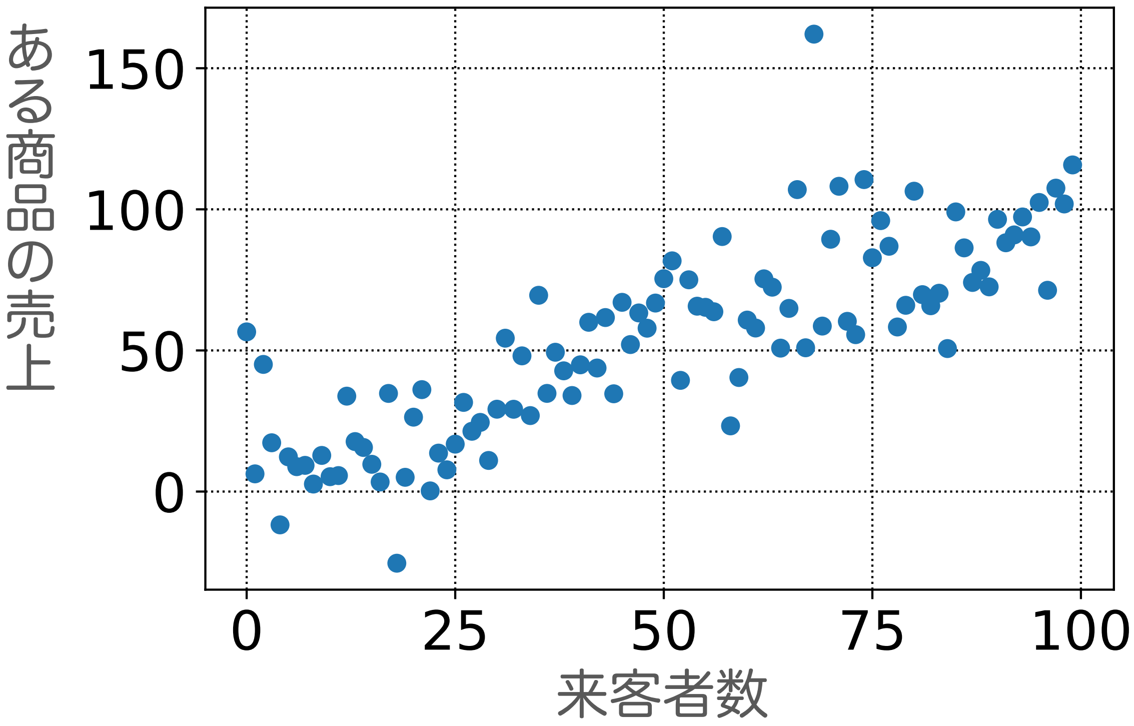


線形回帰

1. 回帰とは
2. 線形モデル（多項式モデル）
3. モデルの評価基準（二乗誤差）
4. モデルパラメータの最適化（最小二乗法）
5. 線形回帰モデルの解釈
6. 線形回帰まとめ（モデル、評価基準、最適化の観点から）

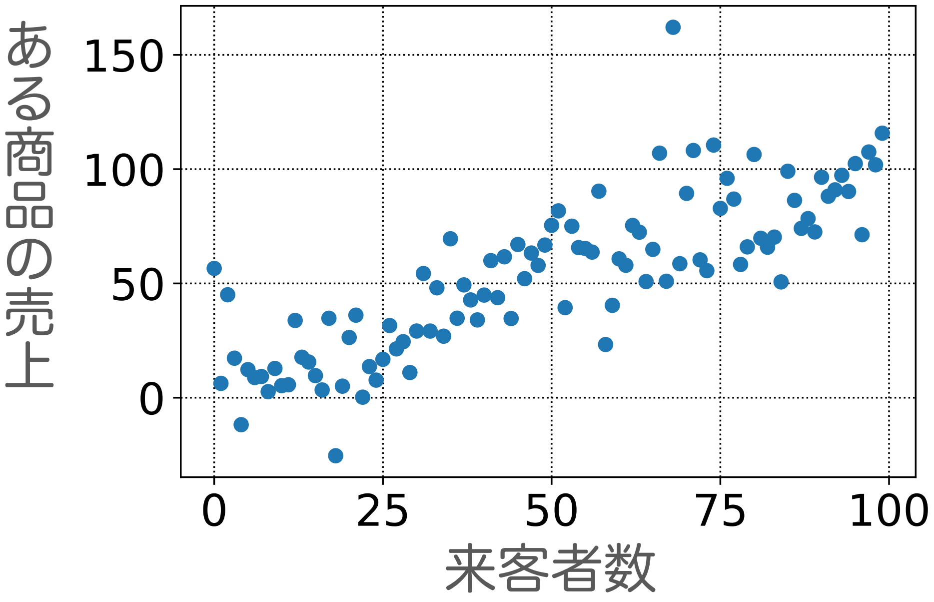
回帰（Regression）とは

- 回帰とは、数値を予測すること
  - 教師信号が連続値のケースは回帰問題に属する
  - 例）来客数からある商品の売上を予測



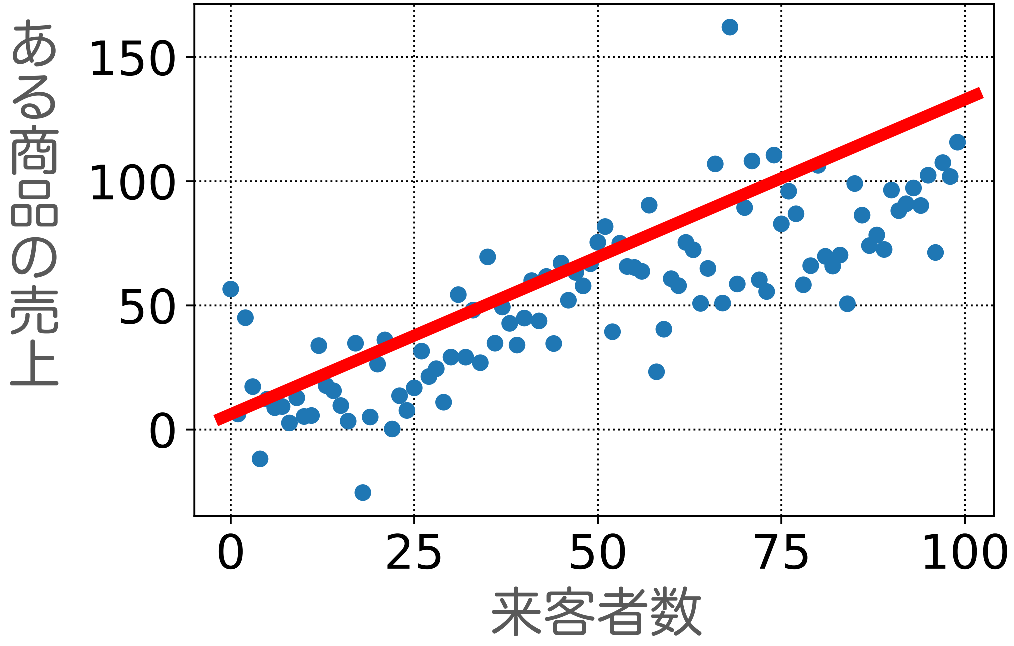
問題

- 小売店にて、ある商品の売り上げと来店客数の関係を調査したところ下図の結果がえられた
- このデータから、ある来店客数のときにおける商品の売上を予測するモデルをつくりたい
- どのようなモデルをつくればいいか？



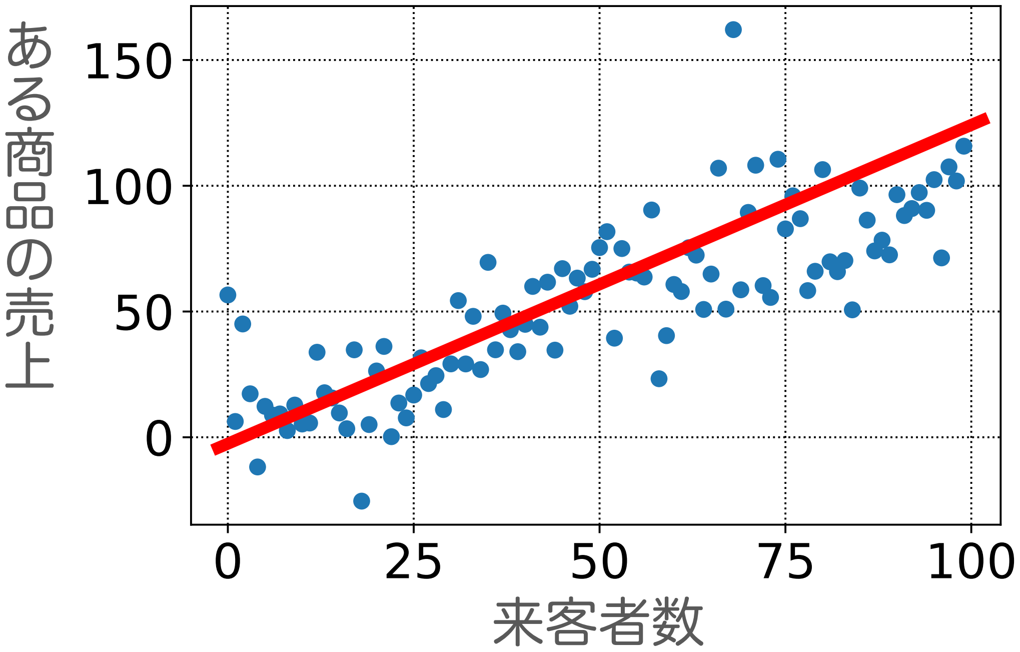
問題に応えるための1つのアイデア

- 直線を引くのがよさそう
- 直線といえば、 $y = ax + b$を思いつくが、傾きaと切片bはどうやって決めればいいのか?



線形回帰モデル（Linear Regression Model）｜説明変数が1次元のとき

- （2次元の）直線：切片と傾きが学習可能なモデルパラメータ
- 切片と傾きのベクトル表現を$\mathbf{w}=(w_0,\ w_1)^{\mathrm{T}}$、モデルへの入力（説明変数）を$\mathbf{x}=(1,\ x_1)^{\mathrm{T}}$としたとき、モデルの出力$\hat{Y}(\mathbf{x};\mathbf{w})$は次式で定義する



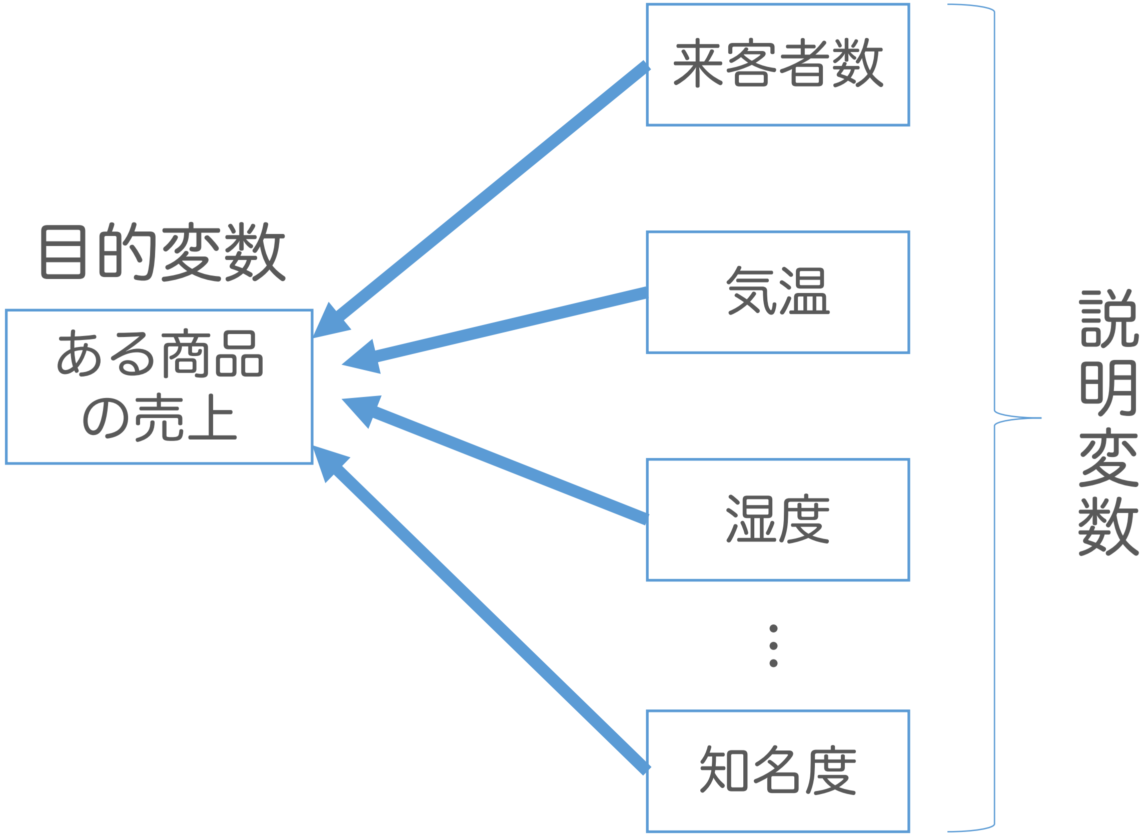
$$\underline{\hat{Y}(\mathbf{x};\mathbf{w})} = \underline{\mathbf{w}^{\mathrm{T}}\mathbf{x}} = w_0 + w_1 x_1$$

ある重み $\mathbf{w}$ のときに、入力(説明変数)$\mathbf{x}$を入れた時の出力。
推定値なので、$\mathbf{Y}$の上に^がついている。

ベクトル$\mathbf{w}$を転置したものと入力(説明変数)のベクトル$\mathbf{x}$の積。

売上と来客者数の関係さえわかれば十分？

- 売上に関係がありそうな要因は、来客者数以外にも色々ありそう
- 線形回帰モデルの考え方で、他の要因も考慮して予測できないだろうか？

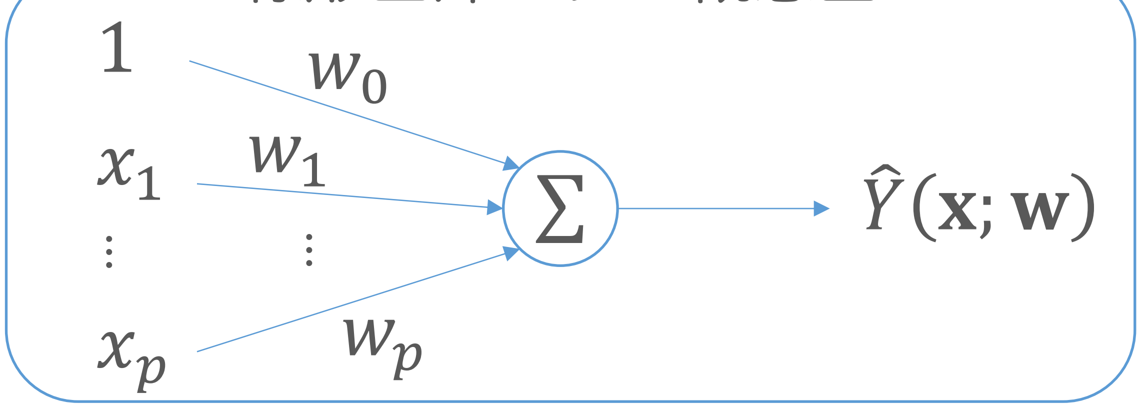


線形回帰モデル（Linear Regression Model）| 説明変数が多次元の場合

- 学習可能なパラメータを$\mathbf{w} = (w_0, w_1, \dots, w_p)^{\mathrm{T}} \in \mathbb{R}^{p+1}$、モデルへの入力$\mathbf{x} = (1,\ x_1, \dots, x_p)^{\mathrm{T}}$としたとき、線形回帰モデルの出力$\hat{Y}(\mathbf{x}; \mathbf{w})$を次式で定義する

$$\hat{Y}(\mathbf{x}; \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_{i=0}^{p} w_i x_i = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p$$

線形回帰モデル概念図

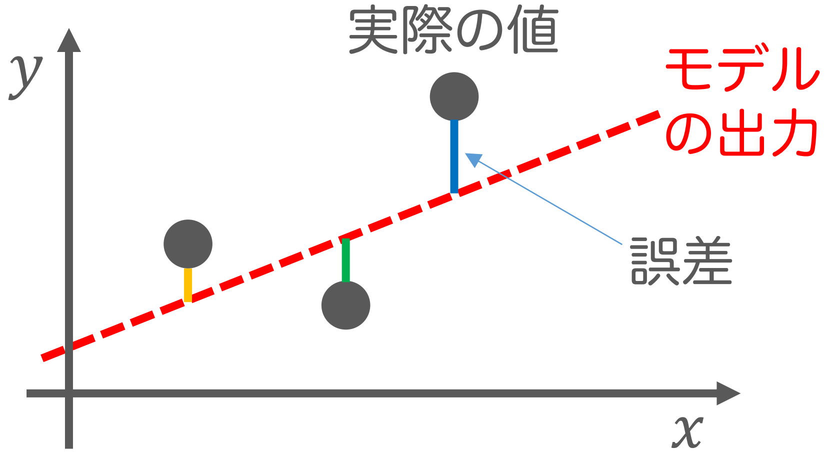


各説明変数に重みづけして
足し合わせるシンプルなモデル

学習データをよく説明できる
ように重み$\mathbf{w}$を学習

モデルの評価基準（二乗誤差）

- モデルの良さは出力と教師データの二乗誤差で評価



二乗誤差の考え方

1辺の長さが誤差の正方形の面積の総和

$$E_D = \frac{1}{2}(\square + \square + \square)$$

を最小化すれば、誤差も小さくなる！

モデルの評価基準（二乗誤差）

- N個の学習データのもとで二乗誤差の総和E_Dは次式で定義される

$$E_D = \frac{1}{2}\sum_{n=1}^{N}\Big(\underbrace{\hat{Y}\big(\mathbf{x}^{(n)};\mathbf{w}\big)}_{y\text{の推定値}} - \underbrace{y^{(n)}}_{y\text{の正解値}}\Big)^2$$

$\mathbf{x}^{(n)}$：n番目のデータの説明変数

$y^{(n)}$：n番目のデータの目的変数
＝教師データ

モデルパラメータの最適化（最小二乗法）

- 二乗誤差を最小化する方法 ＝ 最小二乗法
- 線形回帰モデルは、二乗誤差を最小化する重みは解析的に（＝数式変形によって）求めることが可能
- 二乗誤差が最小になる条件式 $\frac{\partial E_D}{\partial \mathbf{w}} = \mathbf{0}$ を$\mathbf{w}$について解くと…

最適な重み

$$\mathbf{w}^* = \left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}\mathbf{y}$$

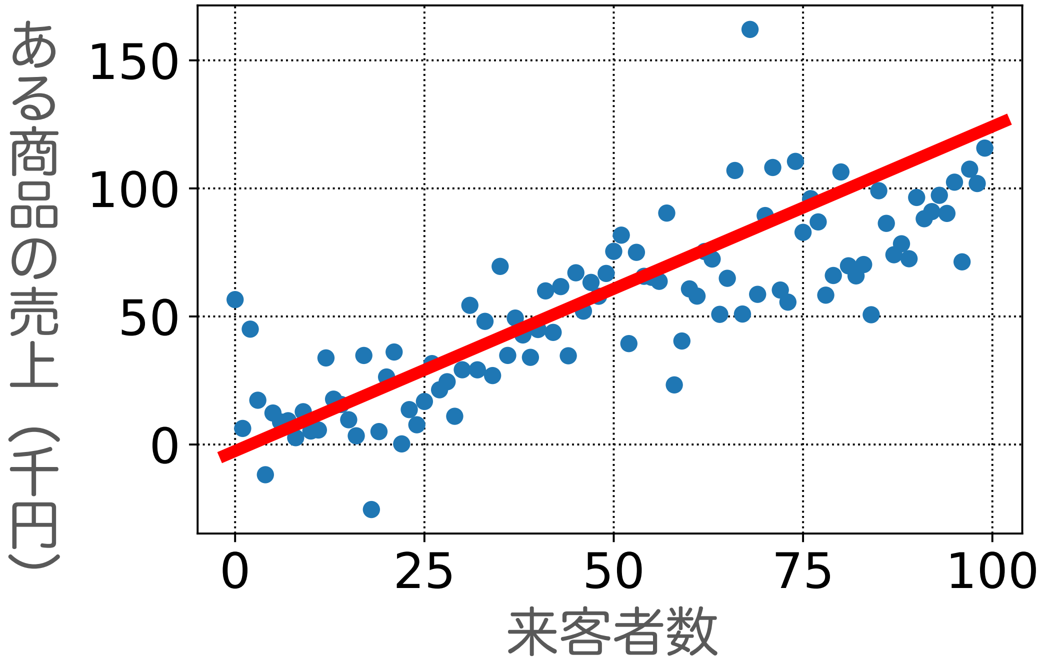
導出の詳細は
『ITエンジニアのための機械学習理論入門』
p.64~p.66を参照

$$\Phi = \begin{pmatrix} \mathbf{x}^{(1)\mathrm{T}} \\ \vdots \\ \mathbf{x}^{(N)\mathrm{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_p^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \cdots & x_p^{(N)} \end{pmatrix}$$

$$\mathbf{y} = \left(y^{(1)}、\ y^{(2)}、\ \ldots、\ y^{(N)}\right)^{\mathrm{T}}$$

線形回帰モデルの解釈 | 直線の意味とは？ 係数の意味とは？

- この直線は来客者数50人なら売上は約6万円になるという関係を示唆
- モデルを得たら直観に反していないか必ずチェックしよう
  - 可視化してデータに重ねてみる、係数をチェックしてみるなど



線形回帰モデルの解釈 | 偏回帰係数

- 線形回帰モデルの係数は、偏回帰係数とも呼ばれる
- 偏回帰係数は、他の変数を固定してその変数だけを動かしたときに得られる目的変数Yの変化量を意味する
- 係数プロットという一覧化の手法で検討するのが一般的
  - DAY3の「特徴選択」で詳細を説明します

線形回帰まとめ（モデル、評価基準、最適化の観点から）

- 線形回帰（回帰分析）：ある変数を他の変数で説明するための直線を求めること
  - 前者の変数のことを目的変数、後者の変数を説明変数と呼ぶ
- 例）1日の売上を目的変数、気温を説明変数にし、その関係性を表す直線を求める
  - この直線を求まれば、明日の天気予報を用いて売上を予測できる

モデル
- 線形回帰モデル

→

評価基準
- 二乗誤差

→

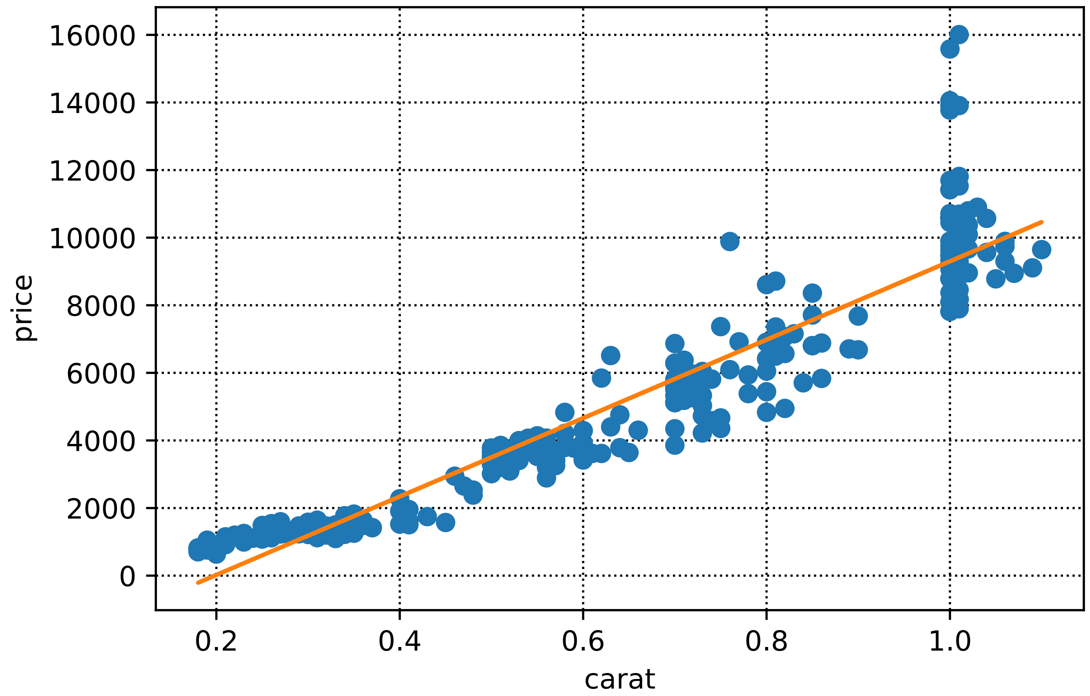
最適化
- 最小二乗法

[演習] 線形回帰（1_linear_regression_psedo_data.ipynb）

- まずは簡単なデータで線形回帰を試してみましょう
- ランダムに直線のパラメータを決めたものよりも、最小二乗法によって決めたパラメータの方が二乗誤差が小さくなることを確認しましょう

[演習] 線形回帰（2_linear_regression_real_data_trainee.ipynb）

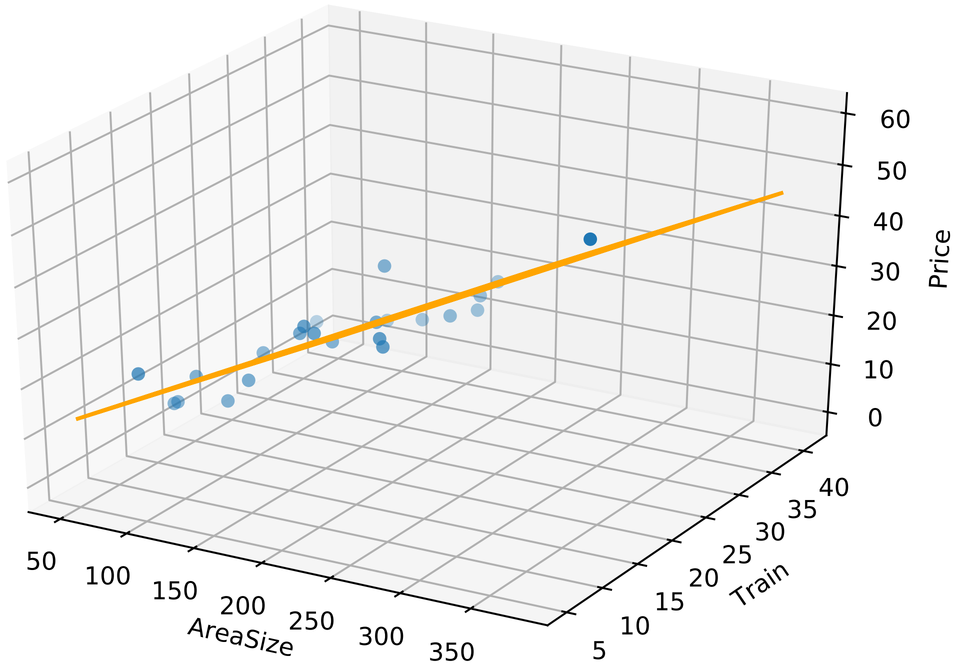
- 実際のデータで線形回帰を実行してみましょう
- ダイヤモンドのカラット数からその価格を予測するモデルを作りましょう



[演習] 線形回帰（3_linear_regression_multi_psedo_data.ipynb）

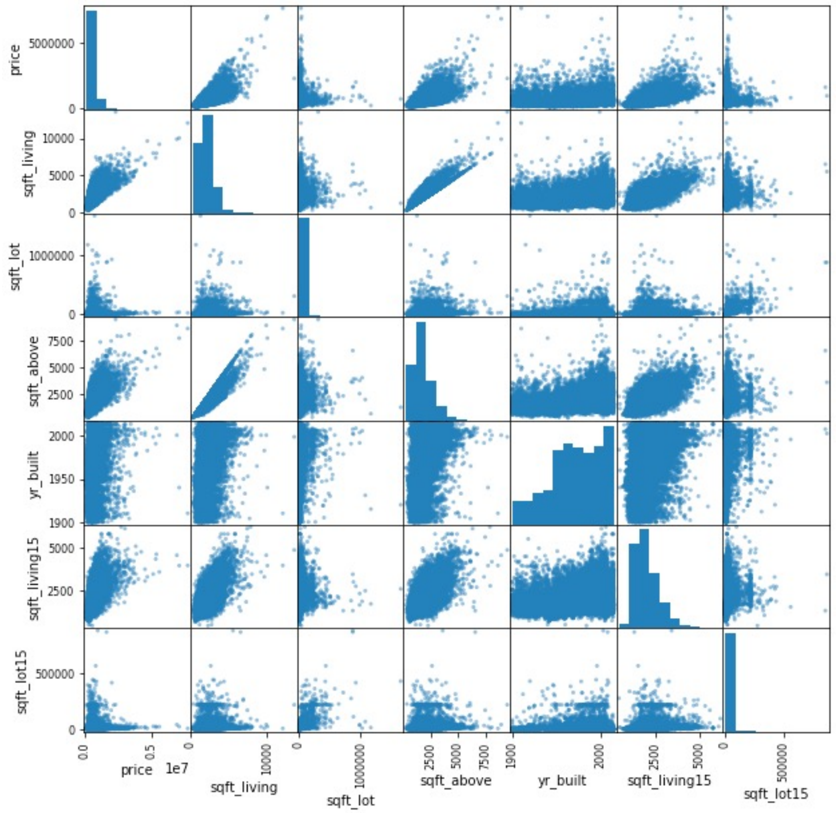
- 説明変数が多い住宅価格データセットで線形回帰をしてみましょう
- さらに説明変数ごとに基本的な統計量を確認してみましょう

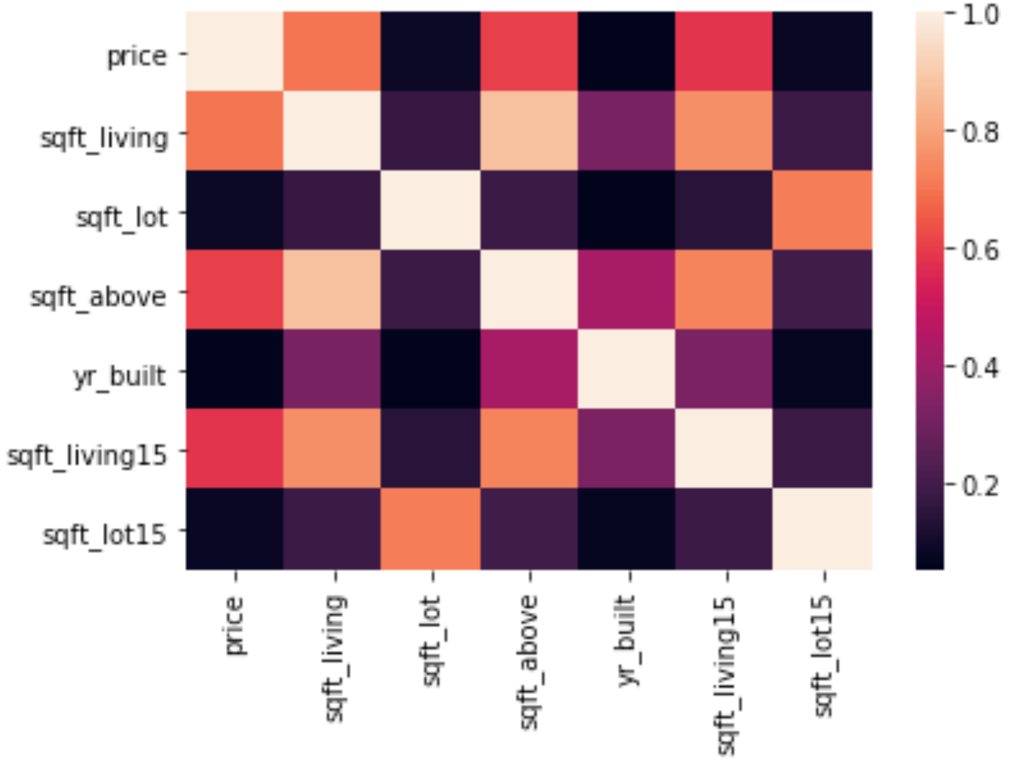
	AreaSize	HouseSize	PassedYear	Price	Train	Walk
count	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000
mean	144.139130	81.356522	9.834783	19.395652	23.260870	6.217391
std	70.086095	26.436955	4.023071	11.605151	10.247915	5.062846
min	50.000000	48.700000	3.100000	5.500000	5.000000	1.000000
25%	99.000000	66.300000	5.700000	11.600000	16.000000	2.500000
50%	139.600000	77.900000	10.500000	17.600000	19.000000	5.000000
75%	173.300000	86.750000	13.150000	25.900000	34.500000	7.500000
max	379.800000	163.700000	14.700000	59.500000	41.000000	20.000000



[演習] 線形回帰（4_linear_regression_multi_real_data_trainee.ipynb）

- データ数が大きな住宅価格データセットで線形回帰をしてみましょう
- 統計量や散布図行列、相関係数なども確認してみましょう





[グループワーク] 線形回帰モデルの使い方

- あるコンビニチェーン店はすべての店舗が駅前に立地している
- あなたは売上予測モデルを構築し、最も売上が高くなりそうな場所に新しい店舗を出店することを思いついた
- 3人程度のグループを作り、以下のディスカッションを進めましょう

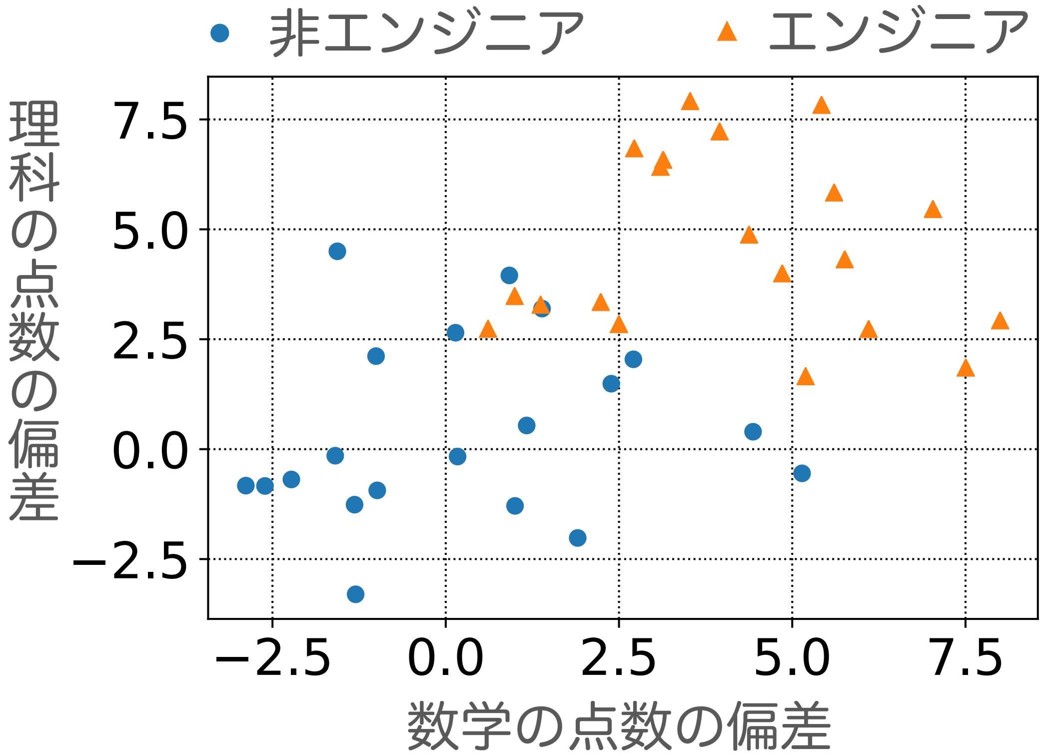
1. 説明変数として用いるべき要因をできるだけ多く挙げてください(5分)
2. 挙げた要因から重要そうなものを5つ程度選択してください
   その要因が、なぜ重要なのかを説明できるようにしてください(5分)
3. その要因に関するデータはどのように取得するのか、
   そして、どの程度データが集まる見込みがあるか検討しましょう(5分)
4. 自分のグループで最終的に使うことを決めた要因、その理由、
   データの取得方法、データ量の見込みを全体に向けて発表しましょう(各2分)

ロジスティック回帰

1. 分類とは
2. データ生成過程の導入
3. シグモイド関数
4. 最尤推定
5. 確率的勾配降下法
6. ロジスティック回帰まとめ（モデル、評価基準、最適化の観点から）

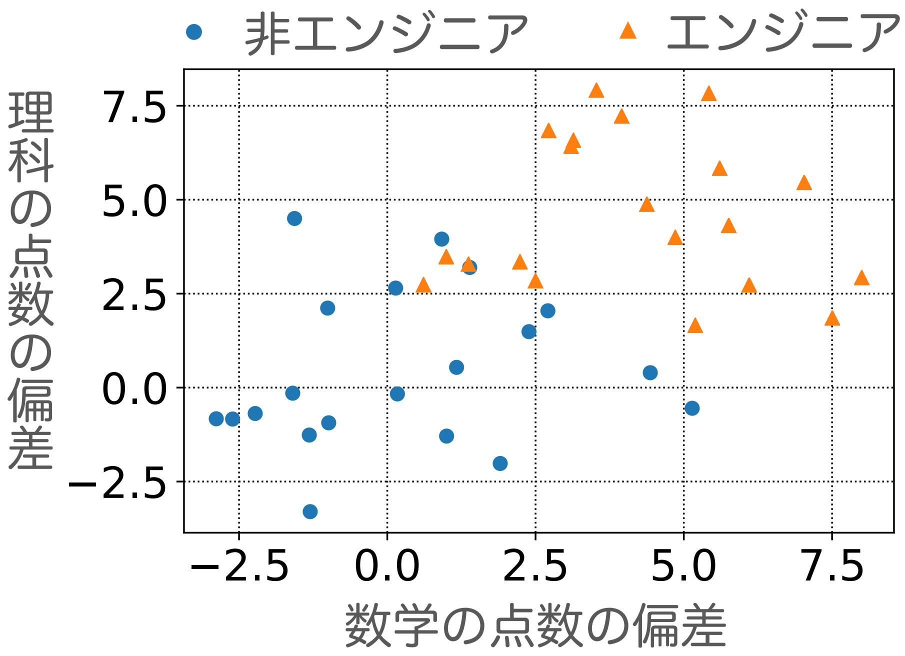
分類（Classification）とは

- 分類とは、カテゴリを予測すること
  - 教師信号が離散値（カテゴリ変数）のケースは分類問題に属する
  - 例）理科と数学の偏差から将来エンジニアになるか予測（2カテゴリ、二値分類）



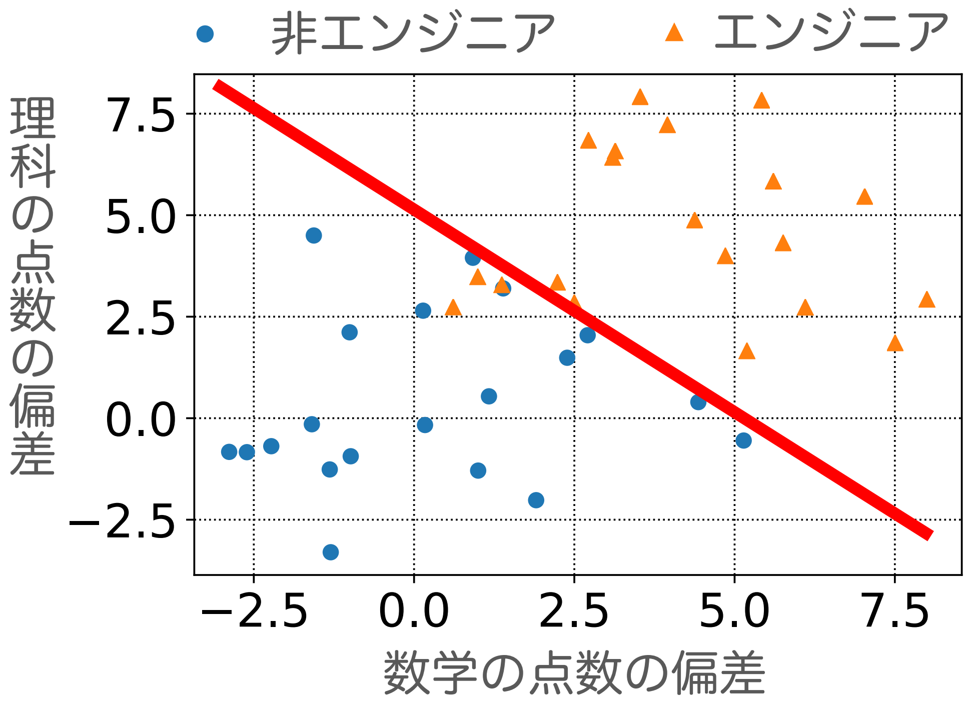
問題

- あるエンジニアと非エンジニアを対象として、学生時代の理科および数学の偏差を調査したところ下図の結果が得られた
- このデータから、ある学生が将来エンジニアになるかを予測するモデルをつくりたい
- どのようなモデルをつくればいいか？



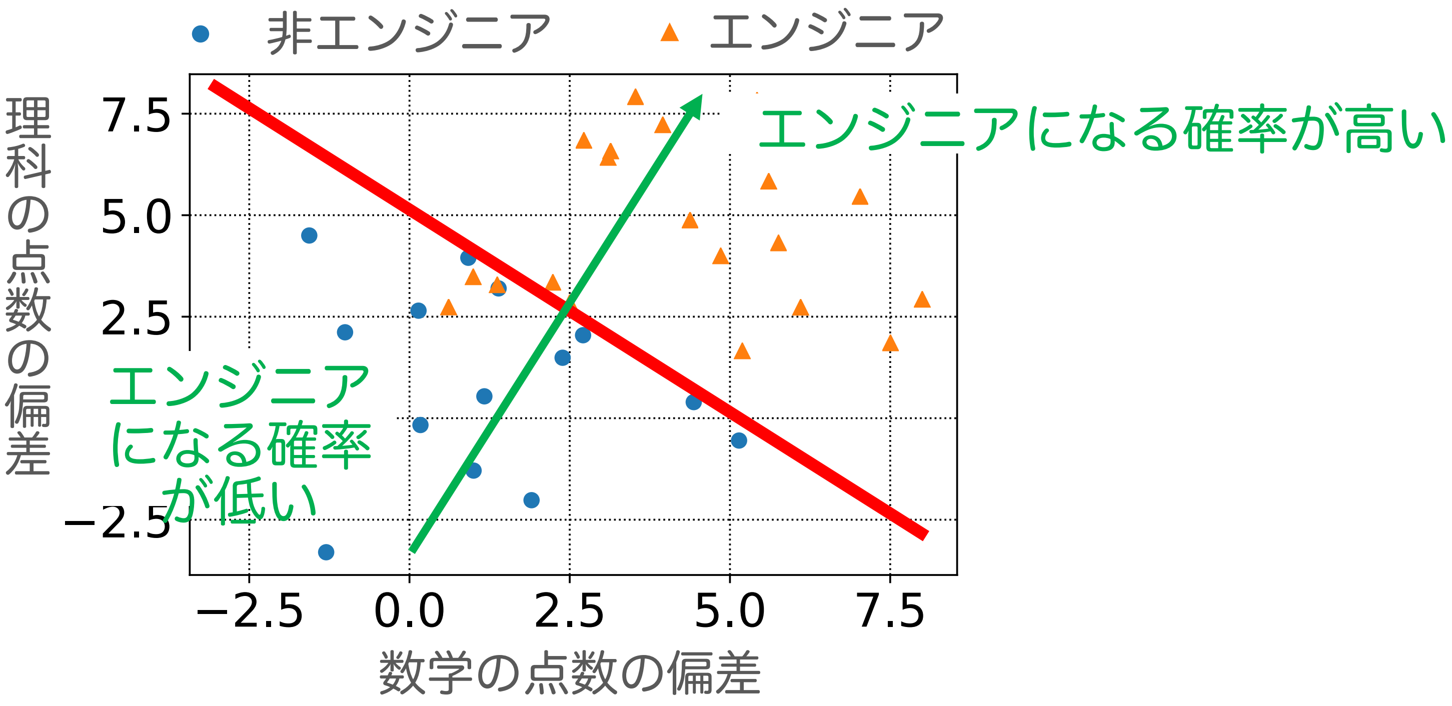
問題に応えるための1つのアイデア

- 直線の境界線を引くのがよさそう
  - これには、線形回帰モデルの考え方が使えるか？



問題に応えるための1つのアイデア

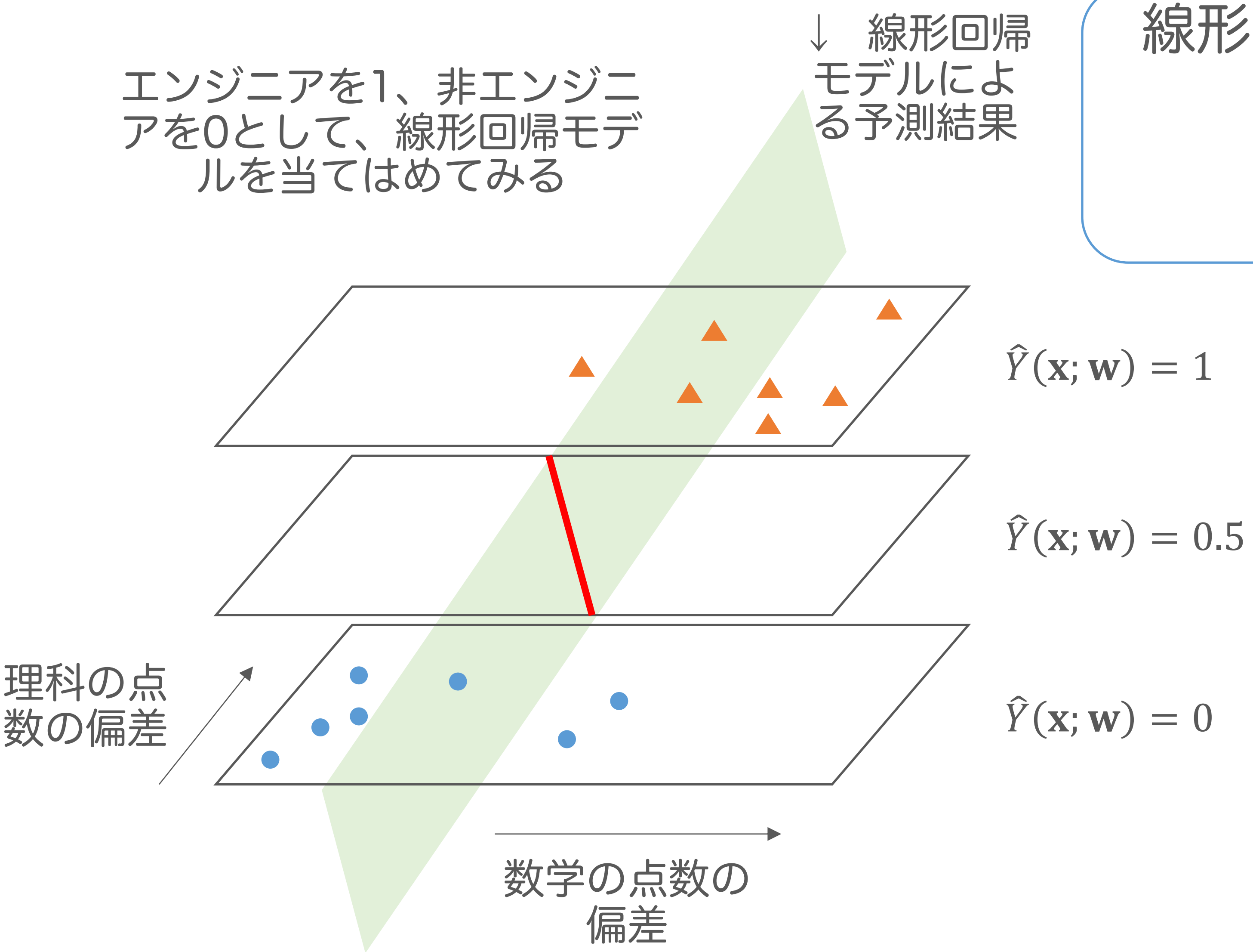
- さらに、エンジニアになる確率（＝確信度）を表現できるとうれしい



線形回帰モデルをどうやって使うのか？

線形回帰モデルの出力が以下を満たす直線を境界線とする

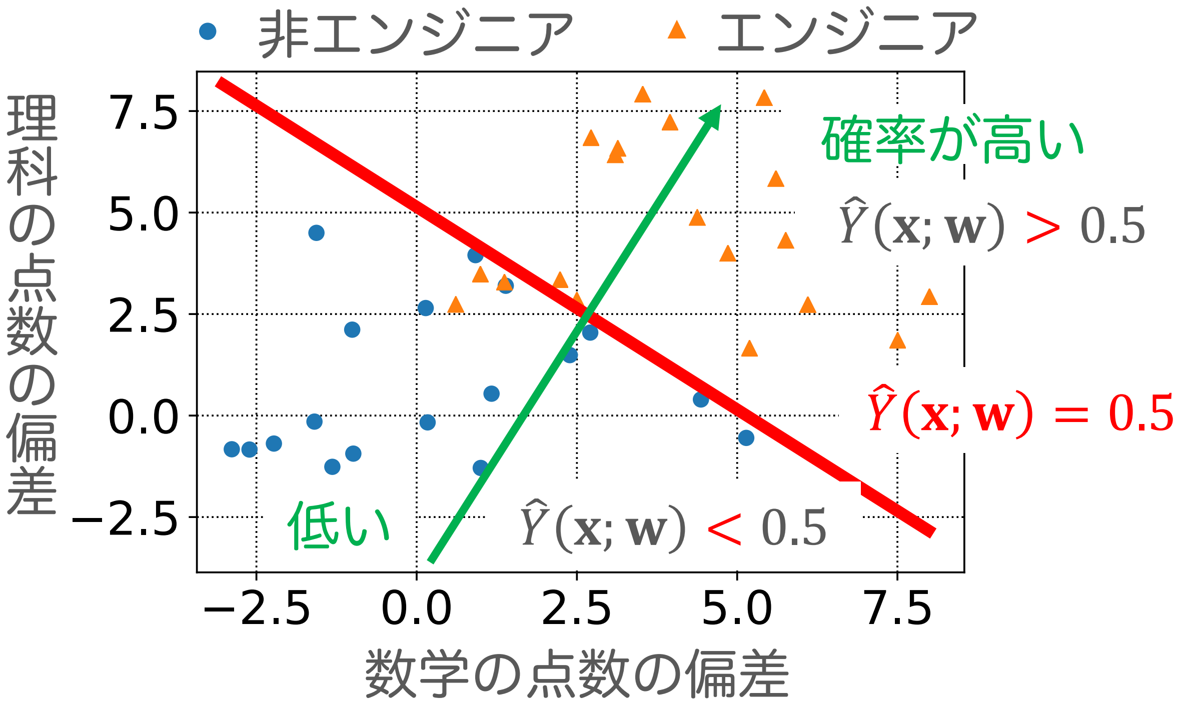
$$\hat{Y}(\mathbf{x};\mathbf{w}) = 0.5$$



緑平面の中の$\hat{Y}(\mathbf{x};\mathbf{w}) = 0.5$ となる直線を境界線として採用する。

得られた赤線を境界線として採用すれば、数学の点数の偏差と理科の点数の偏差で2値を分類できる

データ発生確率の導入



線形回帰モデルの出力が以下を満たす直線を境界線とする

$$\hat{Y}(\mathbf{x};\mathbf{w}) = 0.5$$

出力$\hat{Y}(\mathbf{x};\mathbf{w})$の値が大きい
＝エンジニアである確率が高い
（出力が小さければ確率は低い）

出力$\hat{Y}(\mathbf{x};\mathbf{w})$を確率値に変換

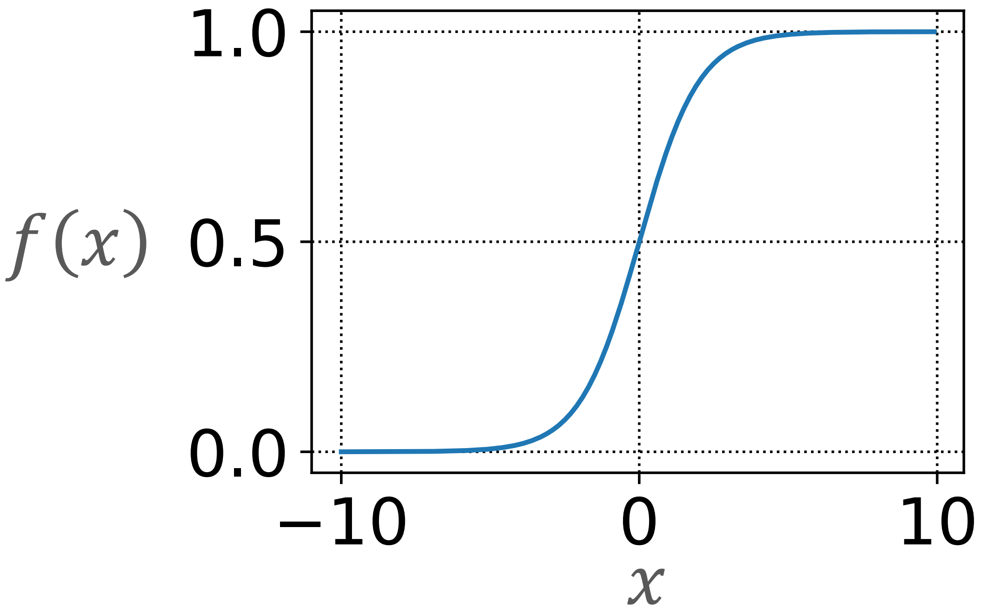
シグモイド関数

- 出力値をシグモイド関数によって確率値に変換
  - シグモイド関数：入力が$-\infty$から∞に大きくなるにつれて、出力が0から1までなめらかに変化する関数

シグモイド関数

定義： $$f(x) = \frac{1}{1+\exp(-x)}$$

性質： $$\frac{d}{dx}f(x) = f(x)(1-f(x))$$

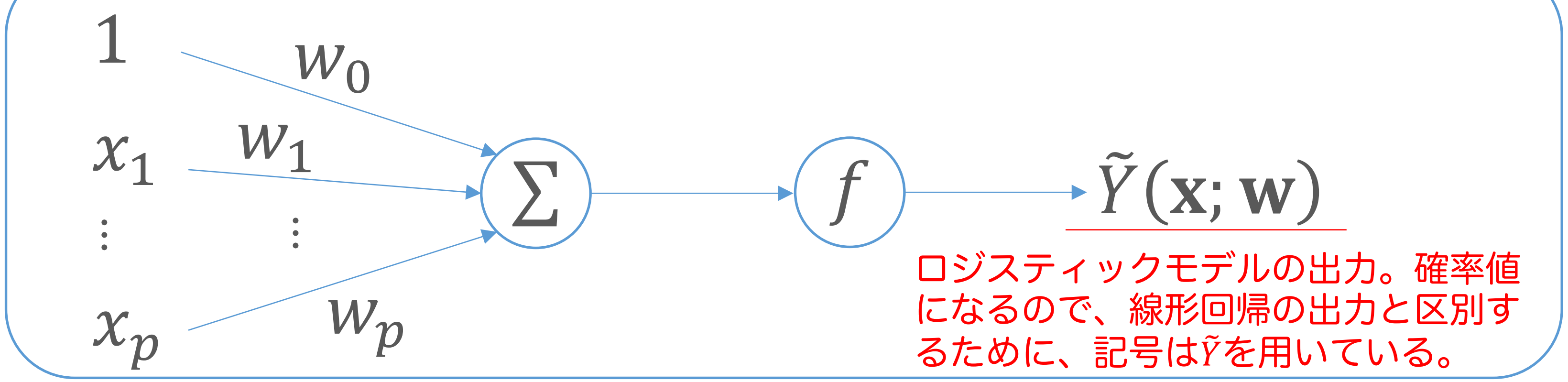


- 学習可能なパラメータを$\mathbf{w} = \left(w_0, w_1, \dots, w_p\right)^{\mathrm{T}} \in \mathbb{R}^{p+1}$、モデルへの入力を$\mathbf{x} = \left(1,\ x_1, \dots, x_p\right)^{\mathrm{T}}$、シグモイド関数を$f(x)$としたとき、ロジスティック回帰モデルの出力$\tilde{Y}(\mathbf{x};\mathbf{w})$を次式で定義する

$$\tilde{Y}(\mathbf{x};\mathbf{w}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right) = \frac{1}{1 + \exp\left(-\underline{\sum_{i=0}^{p} w_i x_i}\right)}$$

線形回帰の出力をシグモイド関数に入力している

ロジスティック回帰モデル概念図



最尤推定（Maximum Likelihood Estimation）

- ロジスティック回帰モデルの出力$\tilde{Y}(\mathbf{x};\mathbf{w})$は、データ$\mathbf{x}$がエンジニアである確率を意味する
- どのデータに対してもエンジニアである/ではないという確率（＝確信度）ができるだけ大きくなるように境界線を引きたい
- できるだけ尤（もっと）もらしい境界線のパラメータ$\mathbf{w}$を学習することを**最尤（さいゆう）推定**という

最尤推定（Maximum Likelihood Estimation）

- モデルの評価指標として、全データに対する確率の対数値（対数尤度；たいすうゆうど） $\ln P$ を使用
- 対数尤度を最大化するようにパラメータ$\mathbf{w}$を学習

$$\ln P=\sum_{n=1}^{N}\left\{y^{(n)}\ln\tilde{Y}\left(\mathbf{x}^{(n)};\mathbf{w}\right)+\left(1-y^{(n)}\right)\ln\left(1-\tilde{Y}\left(\mathbf{x}^{(n)};\mathbf{w}\right)\right)\right\}$$

正解ラベルがエンジニアであるデータは
そのデータがエンジニアであるという確信度
が高いほど評価値が良い

正解ラベルがエンジニアではないデータは
そのデータがエンジニアではないという確信度が
高いほど評価値が良い

確率的勾配法

- 対数尤度が最大となる条件式 $\frac{\partial \ln P}{\partial \mathbf{w}} = \mathbf{0}$ を解いて、最適な重みを求めたいところだが解析解が求まらない…
- そこで繰り返し計算によって最適化な値を探索する
- 勾配$\frac{\partial \ln P}{\partial \mathbf{w}}$は解析的に求まるので、確率的勾配法で最適化

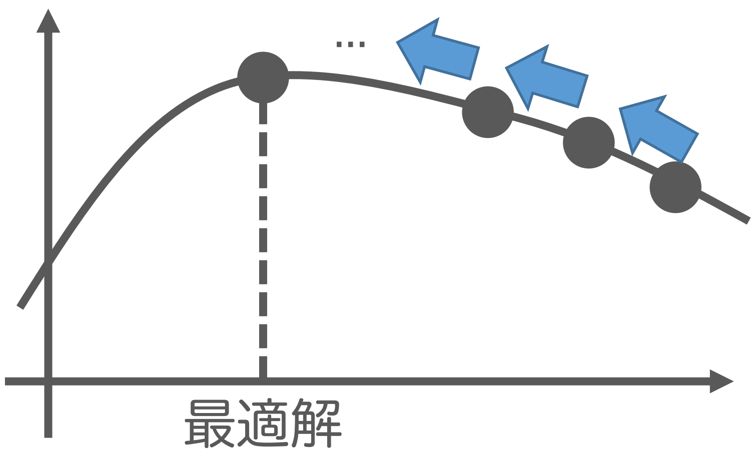
重みの更新式

$$\mathbf{w}^{\text{new}} \leftarrow \mathbf{w} + \eta \frac{\partial \ln P}{\partial \mathbf{w}}$$

η:学習率

データの一部を
使って勾配$\frac{\partial \ln P}{\partial \mathbf{w}}$を計算

回数を繰り返すことで
最適解が数値的に求められる



勾配の計算の詳細は
『ITエンジニアのための機械学習理論入門』p.164を参照

ロジスティック回帰まとめ（モデル、評価基準、最適化の観点から）

- ロジスティック回帰：回帰によってそのカテゴリである確率を予測する手法
- 例）メールに含まれる単語を説明変数として、新たに着信したメールが迷惑メールかどうかを判別する

モデル

- ロジスティック回帰モデル

評価基準

- 対数尤度

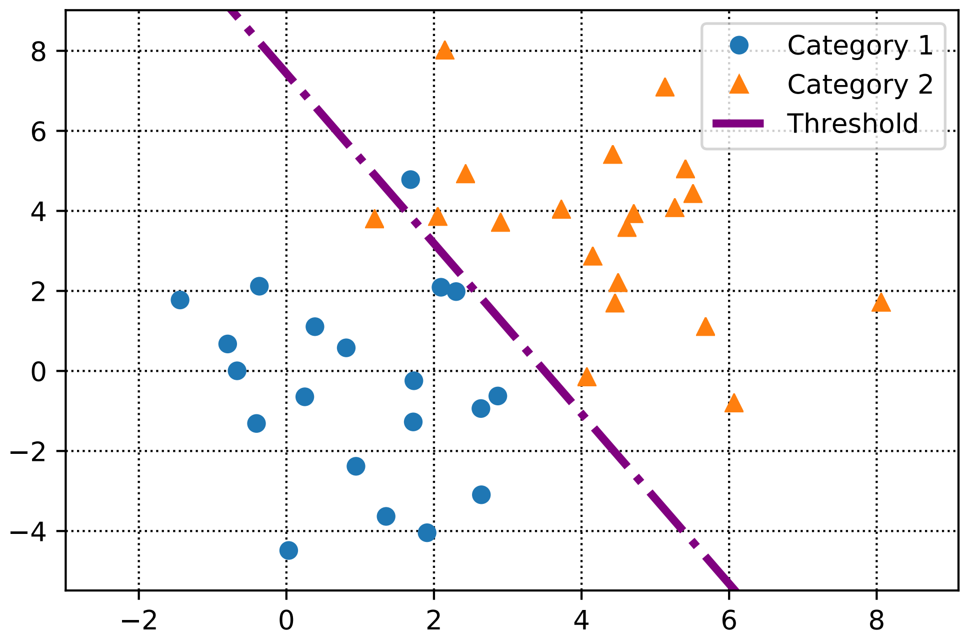
最適化

- 確率的勾配法

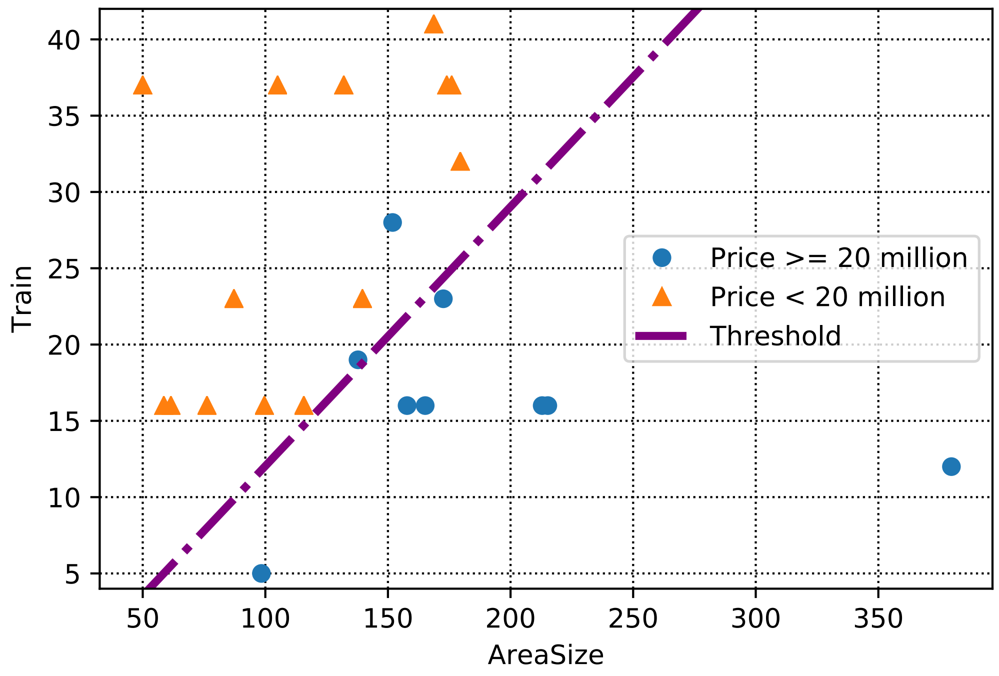
[演習] ロジスティック回帰（5_logistic_regression_psedo_data.ipynb）

- まずは簡単なデータでロジスティック回帰を試してみましょう
- エンジニア予測問題と住宅価格のカテゴリ予測問題でそれぞれモデルを作って確認してみましょう

エンジニア予測



住宅価格のカテゴリ予測



[演習] ロジスティック回帰（6_logistic_regression_real_data_trainee.ipynb）

- 説明変数が多いデータセットでロジスティック回帰を実装してみましょう
- 住宅がリノベーションされているかどうかを予測するモデルを作ってみましょう

多変量モデルへの拡張

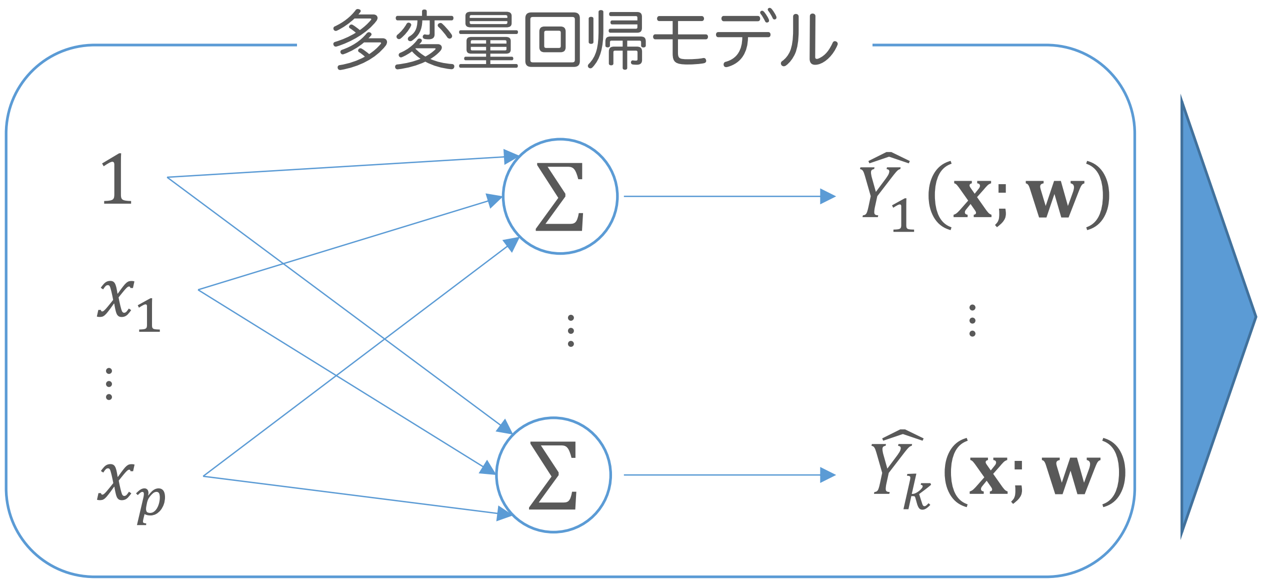
1. 多変量回帰
2. 多クラス分類（ソフトマックス関数と交差エントロピー）

問題

- エンジニア、医者、弁護士、デザイナーを対象として、学生時代の理科および数学の偏差を調査した
- このデータから、ある学生が将来これら4つの職業になる確率をそれぞれ求めたい
- どのようなモデルをつくればいいか？

多変量回帰

- 今までは目的変数が1次元のケースを取り扱ってきた
- しかし、実際には目的変数が多次元であることもありうる
- 線形回帰において出力が多次元のとき→**多変量回帰**

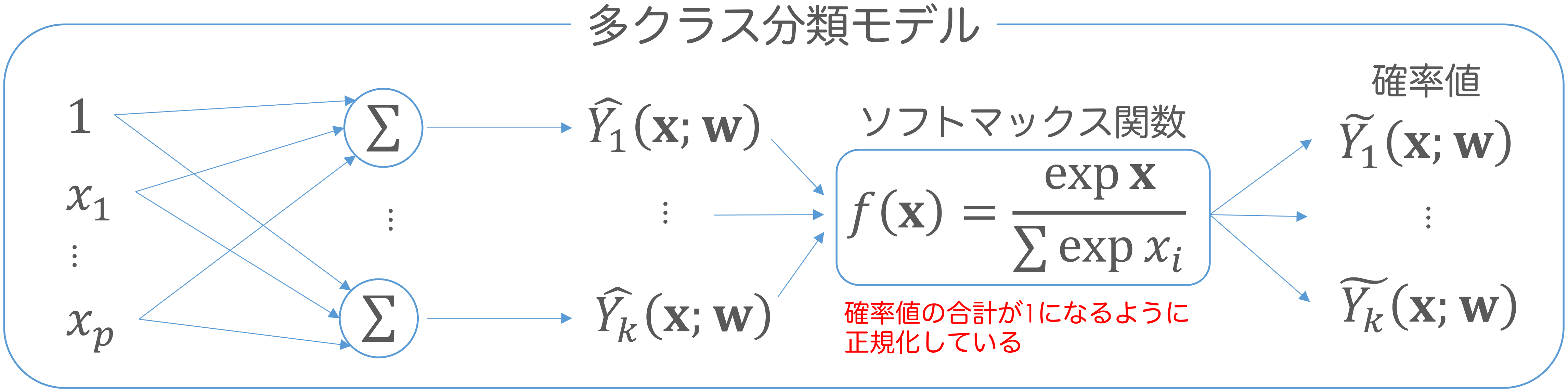


目的変数の各次元ごとに
線形回帰モデルを置く

評価基準もそれぞれの次元に
対する二乗誤差の和でOK

多クラス分類

- 分類問題のときは、各出力は「各カテゴリである確率」をそれぞれ意味する
  - 1つ目は犬である確率、2つ目は猫である確率、3つ目はパンダである確率…
- このような多クラス分類モデルのときは、シグモイド関数の代わりに**ソフトマックス関数**を用いて出力を確率値に変換する



多クラス分類

- 多クラス分類モデルにおける評価指標には交差エントロピーを用いることが多い
  - 深層学習でも一般的に用いられる
- 交差エントロピー＝負の対数尤度

$$\underbrace{-\ln P}_{\text{負の対数尤度 =当てはまりの悪さを示す}} = -\sum_{n=1}^{N}\sum_{i=1}^{k} \underbrace{y_i^{(n)}}_{y\text{の正解値}} \underbrace{\ln \tilde{Y}_i\left(\mathbf{x}^{(n)};\mathbf{w}\right)}_{y\text{の推定値の対数}}$$

モデルの評価指標

1. 回帰問題の評価指標
2. 分類問題の評価指標

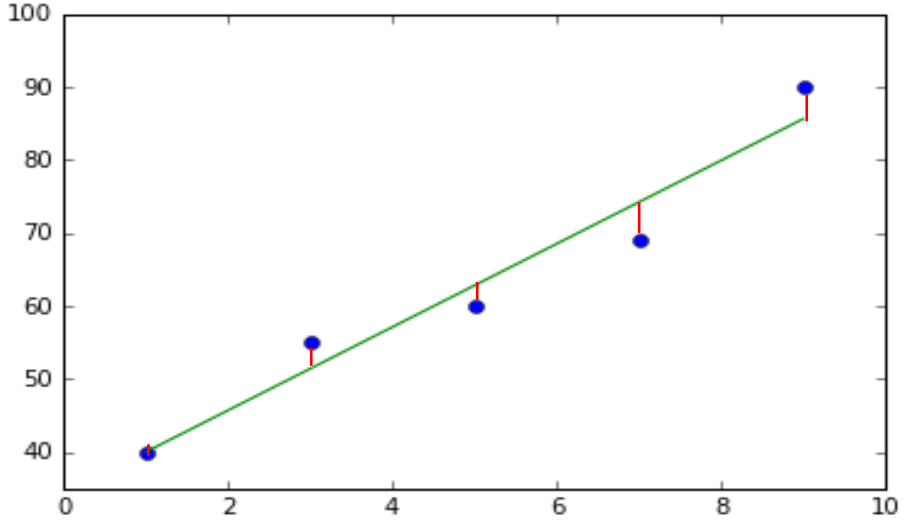
回帰問題の評価指標（MAE・RMSE・MSE）

- RMSE…予測と実際の差の二乗の平均の平方根
  - 平方根平均二乗和誤差とも呼ばれる
- MAE…予測と実際の差の絶対値の平均
  - 平均絶対値誤差とも呼ばれる
- MSE…予測と実際の差の二乗の平均
  - 平均二乗和誤差とも呼ばれる
- MAEは誤差の平均という点で解釈性がよい
  - 右図ならばMAEは「赤線の平均」

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2$$



分類問題の評価（混同行列/Accuracy/Recall/Precision/F1）

- 予測結果と真の結果でクロス集計をしたものを混同行列という
- Accuracy…正解率 (TP+TN) / (TP+TN+FN+FP)
- Recall…実際に正しいもののうち、
  正であると予測された割合 (TP)/(TP+FN)
- Precision…正と予測したもののうち、
  どれくらい正しかったか (TP)/(TP+FP)

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">真の結果</td></tr>
<tr><td>正
(Positive)</td><td>負
(Negative)</td></tr>
<tr><td rowspan="2">予測結果</td><td>正 (Positive)</td><td>TP個</td><td>FP個</td></tr>
<tr><td>負 (Negative)</td><td>FN個</td><td>TN個</td></tr>
</table>

分類問題の評価（混同行列/Accuracy/Recall/Precision/F1）

- F1…RecallとPrecisionの調和平均
  両者のバランスと精度を評価した指標
- 目的によってどれで評価するかは変えるべき
  例）がんの診断→患者視点ならPrecision優先，医師目線ならRecall優先

		真の結果	
		正 (Positive)	負 (Negative)
予測結果	正 (Positive)	TP個	FP個
	負 (Negative)	FN個	TN個

[演習] モデルの評価指標（7_model_evaluation.ipynb）

- 回帰問題、分類問題それぞれにおける評価指標の実装方法を確認してみましょう

```
# 値を予測
y_pred = regr.predict(X)

# MSEを計算
mse = mean_squared_error(y, y_pred)
print("MSE = %s"%round(mse,3) )

# MAEを計算
mae = mean_absolute_error(y, y_pred)
print("MAE = %s"%round(mae,3) )

# RMSEを計算
rmse = np.sqrt(mse)
print("RMSE = %s"%round(rmse, 3) )
```

```
MSE = 14.885
MAE = 3.057
RMSE = 3.858
```

```
# ラベルを予測
y_pred = clf.predict(X)

# 正答率を計算
accuracy =  accuracy_score(y, y_pred)
print('正答率（Accuracy） = {:.3f}%'.format(100 * accuracy))

# Precision, Recall, F1-scoreを計算
precision, recall, f1_score, _ = precision_recall_fscore_support(y, y_pred)

# カテゴリ「2000万以上」に関するPrecision, Recall, F1-scoreを表示
print('適合率（Precision） = {:.3f}%'.format(100 * precision[0]))
print('再現率（Recall） = {:.3f}%'.format(100 * recall[0]))
print('F1値（F1-score） = {:.3f}%'.format(100 * f1_score[0]))
```

```
正答率（Accuracy） = 91.304%
適合率（Precision） = 87.500%
再現率（Recall） = 100.000%
F1値（F1-score） = 93.333%
```

[グループワーク] ロジスティック回帰モデルの使い方

- ある食品Aについて、ECサイトの登録情報をもとにその顧客が買うか買わないかを予測するロジスティック回帰モデルを作成した
- モデル作成に用いた顧客の登録情報：

年齢	性別	現住所の県名	類似商品の購入回数	同じブランド商品の購入回数

[グループワーク] ロジスティック回帰モデルの使い方

- 1. このモデルはどのように使えば利益を挙げられるでしょうか？
  - 利益が挙げられないという場合は、その問題点と改善方法を議論しましょう
- 2. 1の使い方に応じた、モデルの評価指標を議論してみましょう
- 3. 2で決めた指標について実製品 or サービスとして提供可能な基準(KPI)を設定してみましょう
- 4. 1~3の理由を説明できるように整理し、全体に向けて発表しましょう

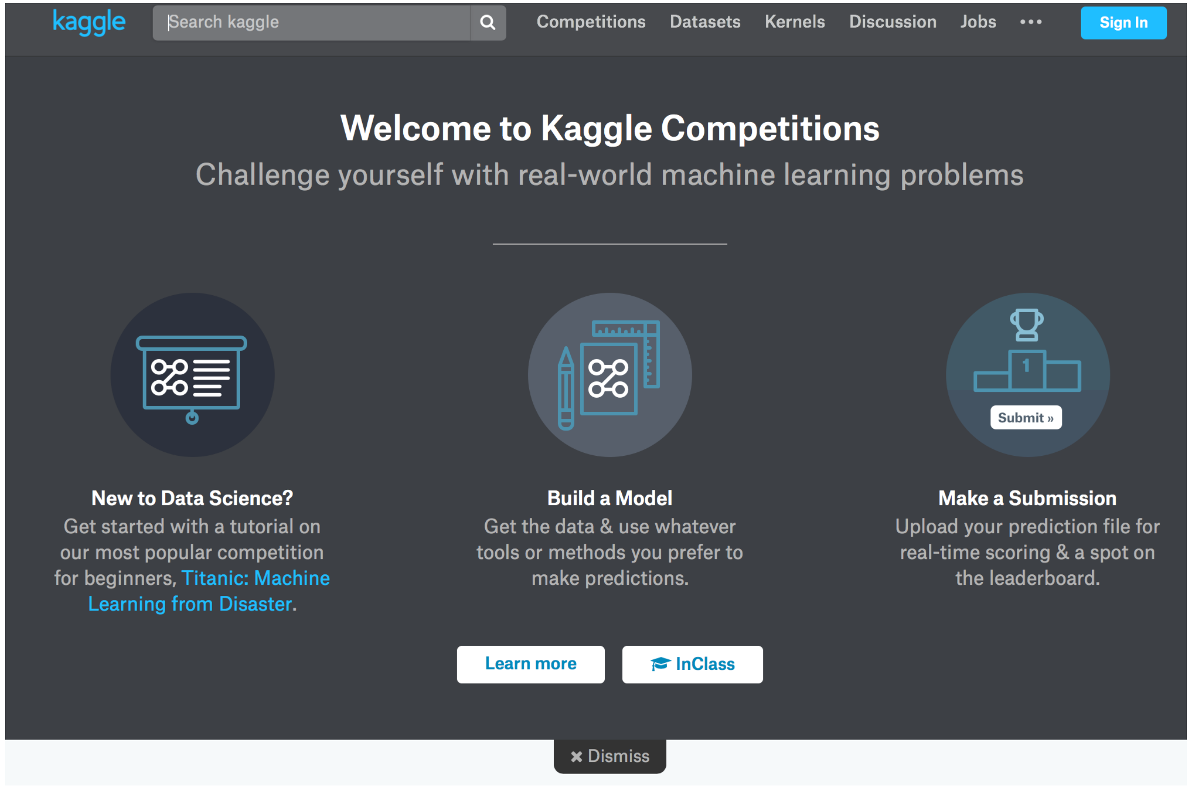
『現場で使える機械学習・データ分析基礎講座』を通しての課題

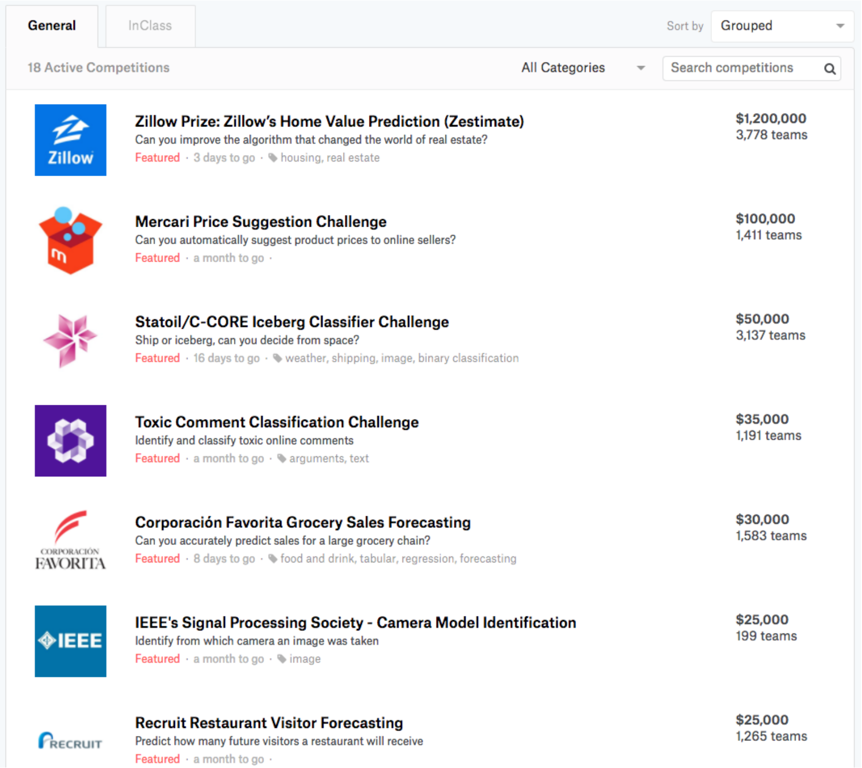
通し課題

- 課題
  - あるクラウドファンディングプロジェクトが成功するか失敗するかをプロジェクト開始前に予測せよ
- データセット
  - kaggle の Kickstarter Projects

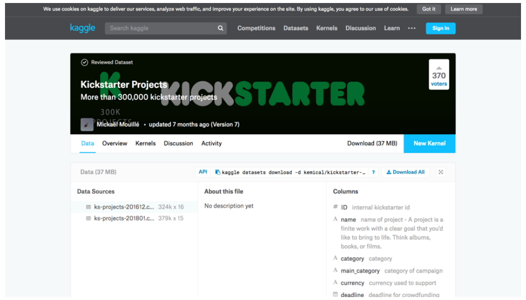
kaggle

- 世界で最も大きいデータ分析コンペプラットフォーム
  - https://www.kaggle.com/





Kickstarter Projects



- Kickstarter は、クラウドファンディングを主催する企業
- Kickstarter が所有するデータセットが Kaggle にて公開されている
    - https://www.kaggle.com/kemical/kickstarter-projects
    - KaggleにアカウントをするとCSVデータをダウンロードできるようになる
    - データは、ks-projects-201801.csvの方だけを使うこと
- 今回は、このデータセットを用いて、あるクラウドファンディングプロジェクトが成功するか失敗するかを**プロジェクト開始前に**予測するモデルを構築する
- 課題の目的は**プロジェクト開始前に**予測することであり、**プロジェクト開始前に**知り得ない情報を予測に用いないようにすること

通し課題 | DAY1後にやること

- Kaggleアカウントを作成し、データをダウンロードする
  （ここからはノートブック上の作業）
- 目的変数と説明変数の関係を確認するためのグラフを作成する
- 欠損値と異常値を確認し、適切に処理する
- 目的変数を説明するのに有効そうな説明変数を見つける
- DAY1で学んだ手法を用いてモデルを作る
  - 今回は2値分類なので、ロジスティック回帰が使える
  - 質的変数はダミー変数に置き換える（pandasのget_dummiesなどを用いる）
- 予測精度を確認する
  - 今回は分類なので、混同行列を作成し、Accuracy、 Recall、 Precisionを求める
- できたところまでをノートブックファイルにまとめ提出する

通し課題 | DAY2後にやること

- DAY2で学んだ手法を用いてモデルを作る
  - 正則化、正規化、標準化、SVMなど
- 交差検証法またはホールドアウト法を用いて、汎化性能を確認する
- 交点探索を用いてハイパーパラメータチューニングを行う
- DAY1後提出時の精度と比較する
- できたところまでをノートブックファイルにまとめ提出する

通し課題｜DAY3後にやること

- DAY3で学んだ手法を用いてモデルを作る
  - 特徴選択、決定木、ランダムフォレスト、アダブースト、ニューラルネットワークなど
- 交点探索を用いてハイパーパラメータチューニングを行う
- DAY2後提出時の精度と比較する
- できたところまでをノートブックファイルにまとめ提出する

通し課題 | DAY4後にやること

- DAY4で学んだ手法を用いてモデルを作る
  - k最近傍法、主成分分析など
- ランダムサーチを用いてハイパーパラメータチューニングを行う
- ベイズ最適化を用いてハイパーパラメータチューニングを行う
  - ベイズ最適化を用いて探索できるライブラリ: scikit-optimize
- DAY3後提出時の精度と比較する
- できたところまでをノートブックファイルにまとめ提出する

課題の提出について

- 各DAY学習後、通し課題に取り組む
- 取り組んだ結果をまとめたノートブックファイルをGフォームにて提出
  - GフォームのURLは別途共有する
- 提出した方には、課題取り組み例をお渡しする
- 提出する際のファイル名は、DayX_work_お名前. ipynbとすること
  - DAY1後提出の場合、例えば『 Day1_work_鈴木一郎. ipynb』

機械学習モデルを作る前に、グラフを描きましょう

- より良い機械学習モデルを作るためには、グラフを描くことが重要
- グラフを描くと、説明変数と目的変数の関係性がみえてくる
  - 例）家の価格は新築ほど高い
  - 例）色が鮮やかならば毒キノコである確率が高い
- グラフを描くと、欠損値や異常値に気づける
- グラフを描くと、機械学習モデルの動作を検証することができる
  - 機械学習モデルがグラフと異なる傾向を出力したとき、すぐにおかしいと気付ける

Any Questions?

Appendix

機械学習の文献を読む際に最低限覚えておきたい数学記号

機械学習の文献を読む際に最低限覚えておきたい数学記号

ギリシャ文字

ギリシャ文字の一覧を以下に示す。頻出度は作者の独自の判断で定めたものである。

頻出度	大文字	小文字	読み方	使用例
高	A	α	アルファ	係数
高	B	β	ベータ	係数
高	Γ	γ	ガンマ	係数,Γ関数
中	Δ	δ	デルタ	微小区間
高	E	ϵ	イプシロン	係数,微小なもの,誤差項
低	Z	ζ	ゼータ	
中	H	η	イータ	係数
高	Θ	θ	シータ	角度,推定すべきパラメータ
低	I	ι	イオタ	
低	K	κ	カッパ	係数
高	Λ	λ	ラムダ	係数
高	M	μ	ミュー	平均

頻出度	大文字	小文字	読み方	使用例
低	N	ν	ニュー	
低	Ξ	ξ	グザイ	
低	O	o	オミクロン	
高	Π	π	パイ	確率,Πは総乗記号
高	P	ρ	ロー	係数
高	Σ	σ	シグマ	標準偏差,Σは総和記号や共分散行列など
中	T	τ	タウ	時間
低	Υ	υ	ユプシロン	
高	Φ	ϕ	ファイ	係数
中	X	χ	カイ	χ^2分布
高	Ψ	ψ	プサイ	係数
低	Ω	ω	オメガ	角速度

機械学習の文献を読む際に最低限覚えておきたい数学記号

記号

記号	意味	用例
^	推定値	$\hat{y}, \hat{x}$
-	平均値	$\bar{y}, \bar{x}$

不等号

記号	意味
≦	小なりイコール
≤	小なりイコール, ≦と同じ
≧	大なりイコール
≥	大なりイコール, ≧と同じ
≃	近似イコール

集合記号

記号	意味	用例
{ }	集合	$x_i = \{x_1, x_2, x_3, \ldots x_{100}\}$
∈	属する	$x \in \{1,2,3,\ldots,100\}$
R	実数	$x \in \mathbf{R}$
Z	整数	$x \in \mathbf{Z}$
≡	定義	$A \equiv B$, AをBと定義する
∪	または	$A \cup B$
∩	かつ	$A \cap B$
---	---	--

機械学習の文献を読む際に最低限覚えておきたい数学記号

確率

記号	意味	用例
$N(\,,\,)$	正規分布	$N(\mu, \sigma^2)$
$\sim$	ある確率変数がある確率分布に従う	$\epsilon \sim N(\mu, \sigma^2)$
$\propto$	比例	$p \propto q$, pとqは比例している
$p(\,,\,)$	同時確率	$p(A, B)$,AとBが同時に生じる確率
$p(\,\mid\,)$	条件付き確率	$p(A \mid B)$, Bが生じたという条件つきのもとでのAが生じる確率

機械学習の文献を読む際に最低限覚えておきたい数学記号

表記に関する慣例

- ×は省略されることが多い
  例、$a \times x = ax$
- カッコは省略されることがある
  $log(x) = logx$
- 底が省略された対数は、自然対数
  $log(x) = log_e(x) = ln(x)$
  ln:読み方はエルエヌ,ロンなど
- 総和記号Σは、記号が省略されることがある

$$\sum_{i=0}^{k} P_i = \sum_{i} P_i$$

閉区間、開区間

[a,b]：aとｂを含む区間、閉区間
(a,b)：a〜ｂの区間だが、aとｂを含まない、開区間
[a,b):a〜ｂの区間だが、aは含み、ｂは含まない

絶対値

|x|

ノルム

1ノルム
$\|x\|_1 = |x_1| + |x_2| + |x_3| + ... + |x_n|$

2ノルム(ユークリッドノルム。単にノルムという場合はこれ。)
$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + ... + x_n^2}$
$\|x\|_2$を$\|x\|$と略して表現されることもある

2ノルムの2乗
$\|x\|_2^2$

大文字シグマの使われ方

Σは、総和記号として以外にも共分散行列を意味する記号としても使われる

最尤推定法

最尤推定法

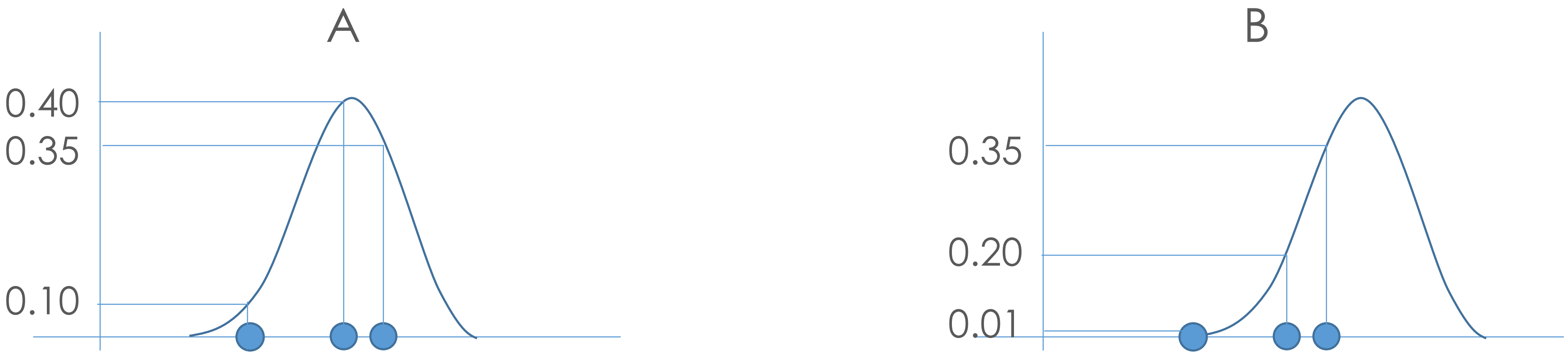
- 以下のデータが観測されたとする。
- このデータに最もよく当てはまる正規分布を求めたい。

最尤推定法

- 以下のどちらの正規分布の方がデータによく当てはまっているか？

最尤推定法

- 正規分布A、正規分布Bのそれぞれについて、尤度を算出する。
- 尤度とは、確率(密度)を掛け算した値。



尤度=0.40×0.35×0.10=0.014 ＞ 尤度= 0.35×0.20×0.01=0.0007

Aの正規分布の方が尤度が大きいので、Aの方がよく当てはまっていると判断する。
これが最尤法。実際には、解析的に最もよく当てはまる正規分布を求める。

情報量、エントロピー、交差エントロピー

情報量

- 情報量とは、事象の驚き度合いを数値化した指標
- 到底起きそうもない事象が起きたことを知った→情報量が大きい
  - 例）砂漠に雨が降った→到底起きそうにない出来事なので情報量 大
- いつでも起きそうな事象が起きたことを知った→情報量は小さい
  - 例）東京に雨が降った→いつでも起きそうな出来事なので情報量 小

$$I(x) = -\log_2 p(x)$$

$I(x)$：ある事象xの情報量（単位はbits）

$p(x)$：ある事象xが起きる確率

[例]

例えば、東京に雨が降ったという情報よりも、砂漠に雨が降った(確率が低い事象)という情報の方が驚くはずである。

ある日に、東京に雨が降る確率$p(x)$を0.3とすれば、その情報量は、

$I(x) = -\log_2 p(0.3) = 1.737$

となり、ある日に、砂漠に雨が降る確率$p(x)$を0.01とすれば、その情報量は、

$I(x) = -\log_2 p(0.01) = 6.644$

となる。

エントロピー（平均情報量）

- エントロピーとは、情報量を平均化したもので**事象のばらつき具合**を表す
- 大半は同じ事象しか起きない→エントロピーが小さい
  - 例）砂漠の天気→ほとんどが晴れなのでエントロピー 小
- 観測するたびに事象がばらついている→エントロピーが大きい
  - 例）東京の天気→晴れだったり雨だったりとバラバラなのでエントロピー 大

$$H=\sum p(x)I(x)=-\sum p(x)\log_2 p(x)$$

H：エントロピー

$I(x)$：ある事象xに対する情報量

[例]

例えば、ある日の東京の天気の確率が、p(晴れ)$= 0.5$、p(雨)$= 0.3$、p(曇り)$= 0.2$とすると、そのエントロピーは、

$H=-\Sigma p(x)\log_2 p(x)=1.485$

となり、ある日の砂漠の天気の確率が、p(晴れ)$= 0.9$、p(雨)$= 0.01$、p(曇り)$= 0.09$とすると、そのエントロピーは、

$H=-\Sigma p(x)\log_2 p(x)=0.516$

となる。

東京の天気は、確率変数xがばらついており、砂漠の天気に比べ予測が難しいと言える。

交差エントロピー

- 交差エントロピーは2つの事柄について、その中で起こりうる事象のばらつきの違いを数値化したもの
- 事象のばらつき方が類似している→交差エントロピーは小さい
  - 例）横浜の天気と東京の天気 → 天気のばらつき方は似ているため
    交差エントロピー 小
- 事象のばらつき方が全く異なる→交差エントロピーは大きい
  - 例）砂漠の天気と東京の天気 → 天気のばらつき方が大きく異なるため
    交差エントロピー 大
- 機械学習では予測値の分布と正解ラベルの分布の違いを測るために用いる

ソフトマックス関数

ソフトマックス関数

- ソフトマックス関数は、複数の入力を正規化し、合計値が1になるようにする関数
- 正規化する前に、指数関数(exp)を計算している。

$$y(k) = \frac{\exp(a_k)}{\Sigma_{i=1}^{n}(\exp(a_i))}$$

k:出力ノードの番号