



# 機械学習 / ディープラーニング 大全 – (1) 基礎編

畠山 大有

日本マイクロソフト株式会社  
プリンシパル ソフトウェア デベロップメント エンジニア



<https://www.facebook.com/dahatake/>



<https://twitter.com/dahatake/>



<https://github.com/dahatake/>



<https://daiyuhatakeyama.wordpress.com/>



<https://www.slideshare.net/dahatake/>

# 機械学習とは 何か?



ドアの後ろに人がいます

入ってくる人は、**男性 / 女性** のどちらで  
しょうか？

既知の情報は以下の通り

年齢: 35歳

年収: 600万円

有給残数: 12日

$$\begin{aligned} F(\text{性別}) &= (0.03 * \text{年齢}) + (0.07 * \text{年収}) - (0.04 * \text{有給残数}) + 0.05 \\ &= (0.03 * 3.5) + (0.07 * 6.0) - (0.04 * 1.2) + 0.05 \\ &= 0.41 \end{aligned}$$

0.5よりも小さいから

= **女性**

# 言葉の整理

- Data Science
  - 従来からある統計・数学的処理
- Machine Learning
  - データを元に、予測をさせる。言い換えれば、特徴の近いデータを探す
- Neural Network
  - 複雑な Machine Learning による予測の一つ
- Deep Neural Network (DNN)
  - 更に複雑な Neural Network
- Artificial Intelligence
  - 「... DNNのうち、人間の行動に近いもの」

# 機械学習の世界

## 機械学習

統計に基づいた手法での分析  
そのため、比較的少ないデータ量と  
計算量で分析を行うことができる

### 活用例

#### ラベル分類

不良品分析, 故障予測, チャーン分析

#### 数値予測

売上予測, 需要予測, 品質管理

#### データ分類

異常検知, 顧客グルーピング

## 深層学習

主に多層のニューラルネットワーク  
を用いた手法での分析  
分析のためには、莫大なデータ量、  
計算量、知識・スキルを要する

#### 画像解析

#### 音声解析

#### テキストや画像等の自動生成

#### 機械学習より強力な分析

## 深層強化学習

定義したあるべき姿に従い試行錯誤  
をして自ら学習を行うための分析手  
法である  
強化学習と、深層学習を組み合わせ  
た  
分析

#### 自律学習型ロボット

#### 自動運転車

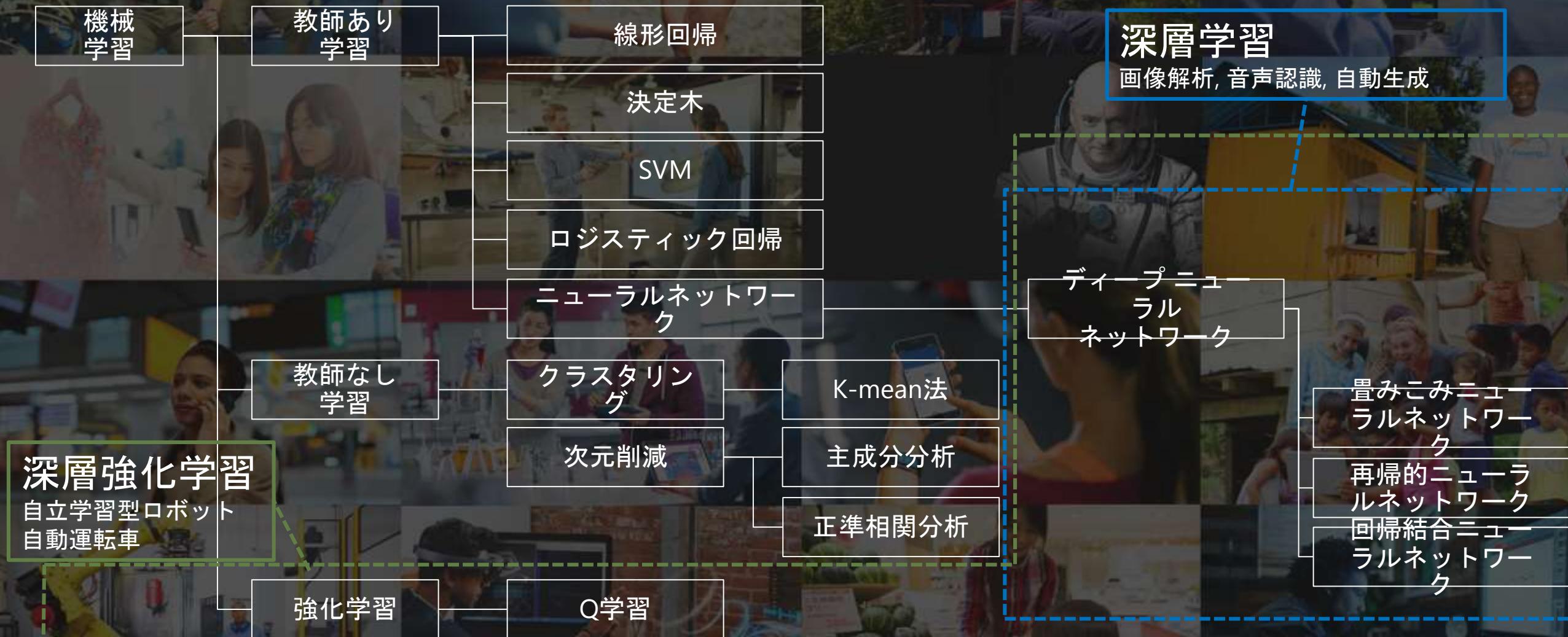


### マイクロソフトが提供する技術

Azure Machine Learning

Cognitive Toolkit / GPU Instance (N-Series)

# アルゴリズムと機械学習



# 機械学習とDeep Learningの違い

- ・ 計算量の違い
  - ・ 機械学習は、統計理論に基づいているので少ない計算量で分析が可能
  - ・ Deep Learningは(Neural Networkの場合)、人間の脳に模倣した分析方式で、理論に基づいているわけない。そのため計算量が膨大となる。  
しかし、爆発的に良い分析精度がでることが確認されている。
- ・ 計算量が大きいという事は...
  - ・ 大量のデータが必要
  - ・ 大量のコンピューティングリソースが必要
  - ・ 精密なパラメータチューニングが必要
  - ・ これらが実現できない場合、分析精度が「全く」出ない。  
そのため、計算量の大きい深層学習は敷居が非常に高い

# Model を理解する

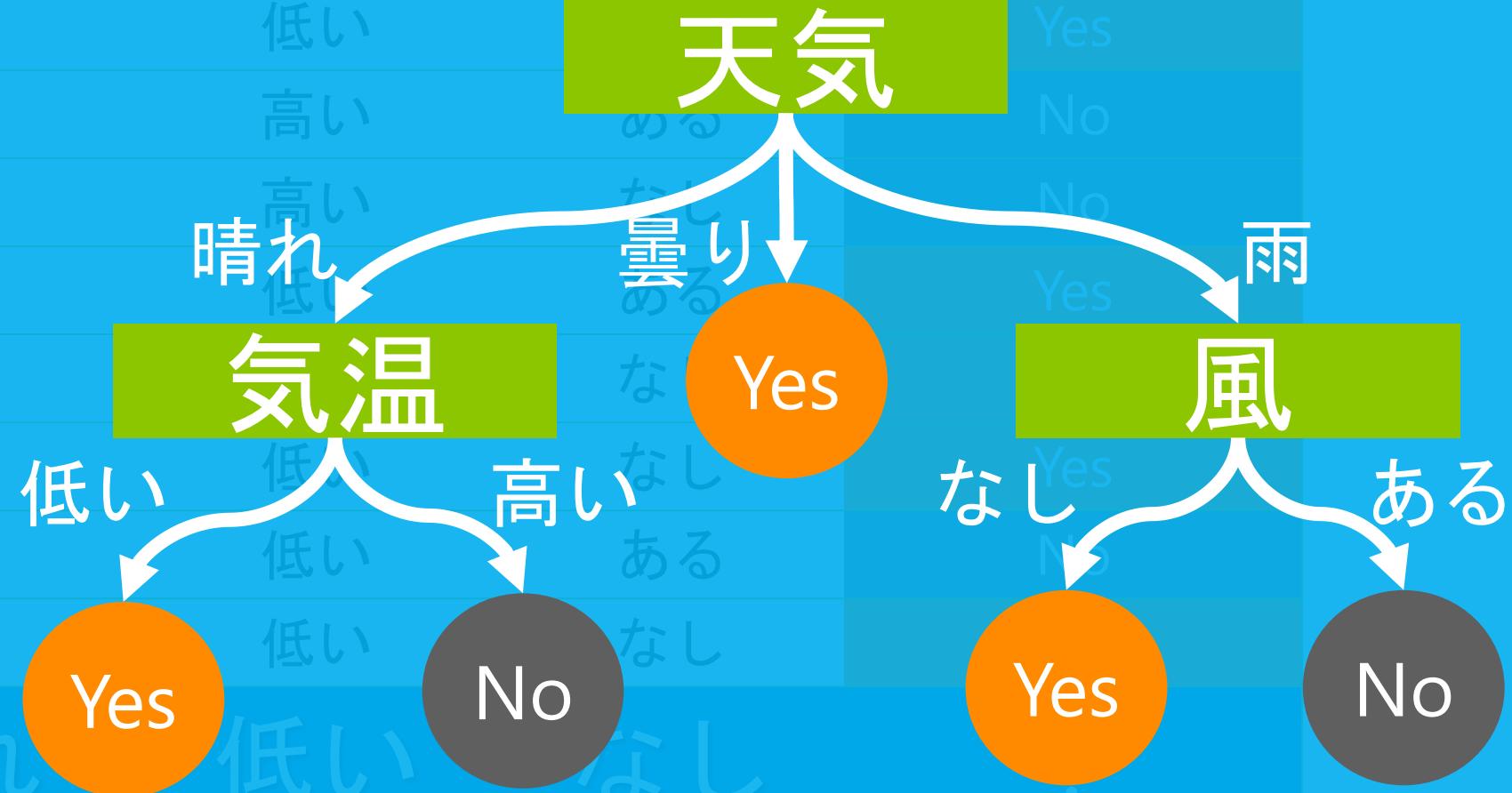


# 試合があると思いますか?

天気	気温	風	試合をしたか?
晴れ	低い	ある	Yes
晴れ	高い	ある	No
晴れ	高い	なし	No
曇り	低い	ある	Yes
曇り	高い	なし	Yes
曇り	低い	なし	Yes
雨	低い	ある	No
雨	低い	なし	Yes
晴れ	低い	なし	?

# どう“ロジック”を組み立てましたか？

## “Model”



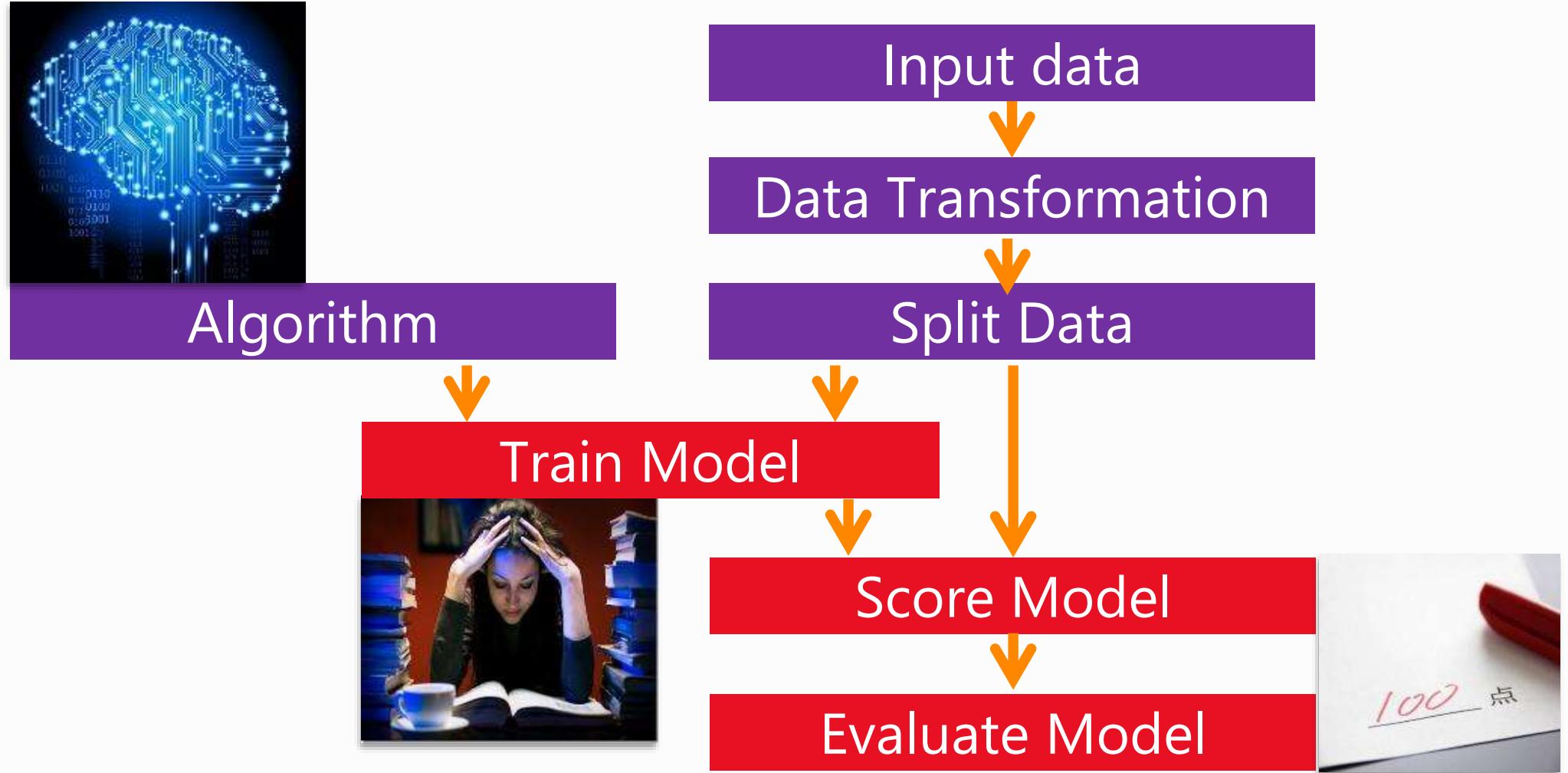
# どう“ロジック”を組み立てましたか?

天気	気温	風	試合をしたか?
晴れ	低い	ある	Yes
晴れ	高い	ある	No
晴れ	高い	なし	No
曇り	低い	ある	Yes
曇り	高い	なし	Yes
曇り	低い	なし	Yes
雨	ある	なし	なし

**“Feature”**  
列の事。データの特徴を表す

**“Label”**  
データ(答え)がある!

# 典型的な Model 開発のフロー





人が  
データ(答え)を教えてあ  
げる

データの中のパターン  
を、  
コンピューター自身に  
探させる (=学習させ  
これを“ロジック”とし  
て利用

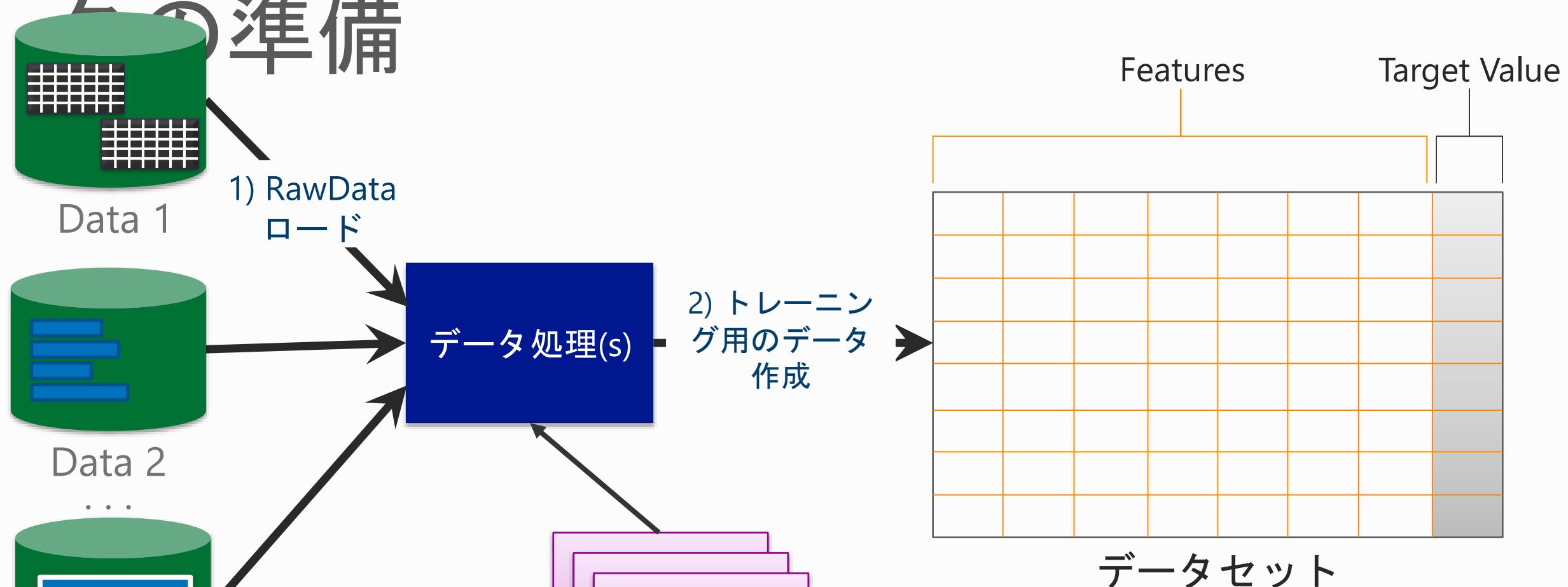
どんなデータでも  
答えを  
出してくれる  
の?



# ちなみに、こういうデータもある....

天気	気温	風	場所	試合をしたか?
晴れ	25	ある	さいたま	Yes
晴れ	27	ある	さいたま	Yes
晴れ	高い	10	東京	No
曇り	5	ある	千葉	No
雨	低い	なし	神奈川	No

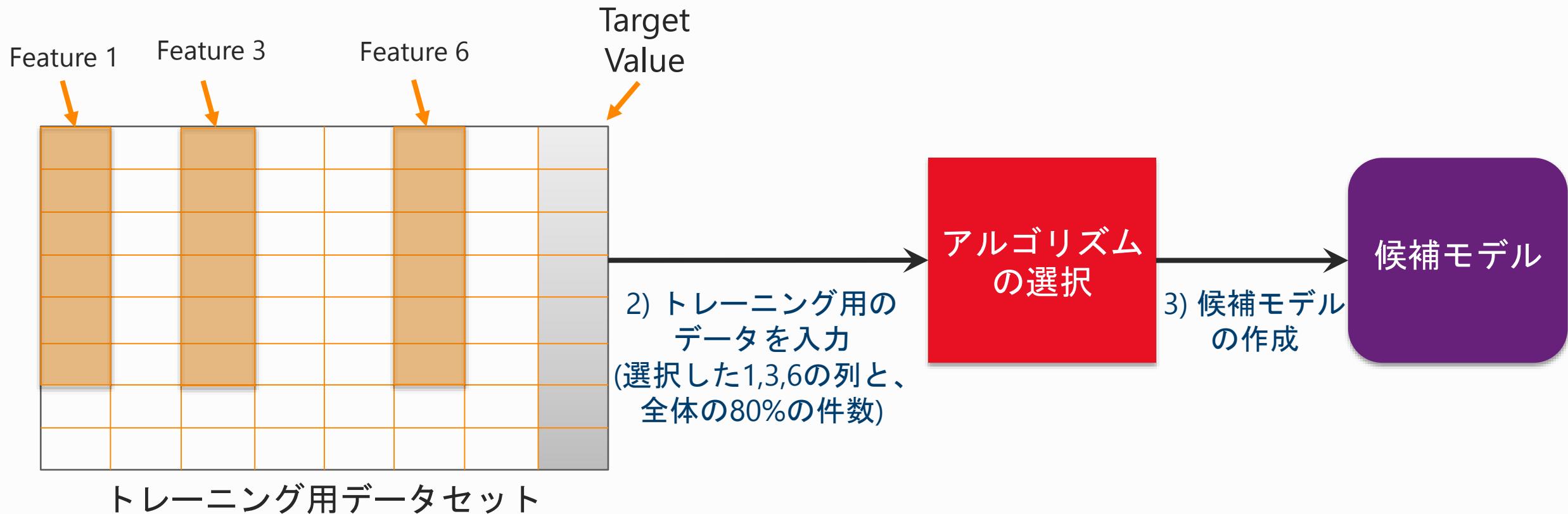
# コンピューターが処理しやすいデータの準備



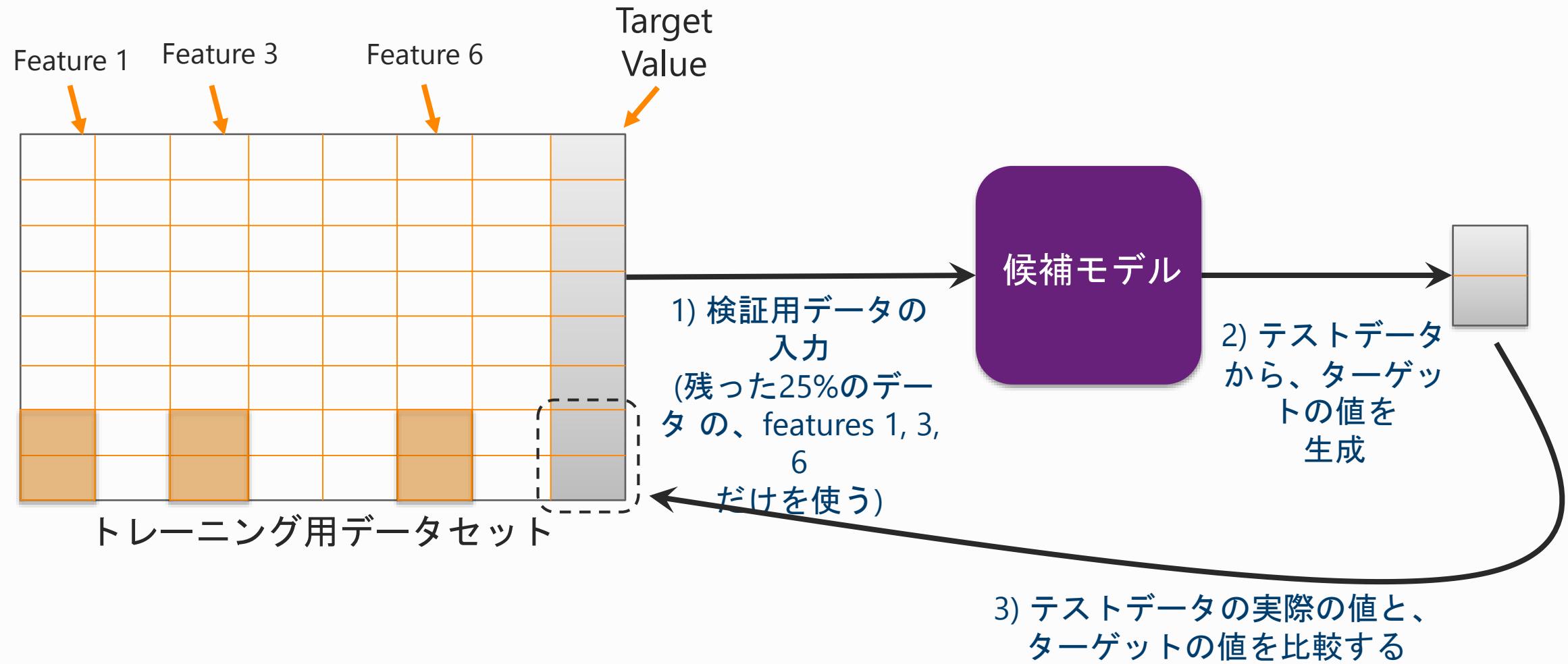
データサイエンスプロジェクトにおいて、最も重要な部分

# データにアルゴリズムを適用して Model 作成

## 1) features の選択

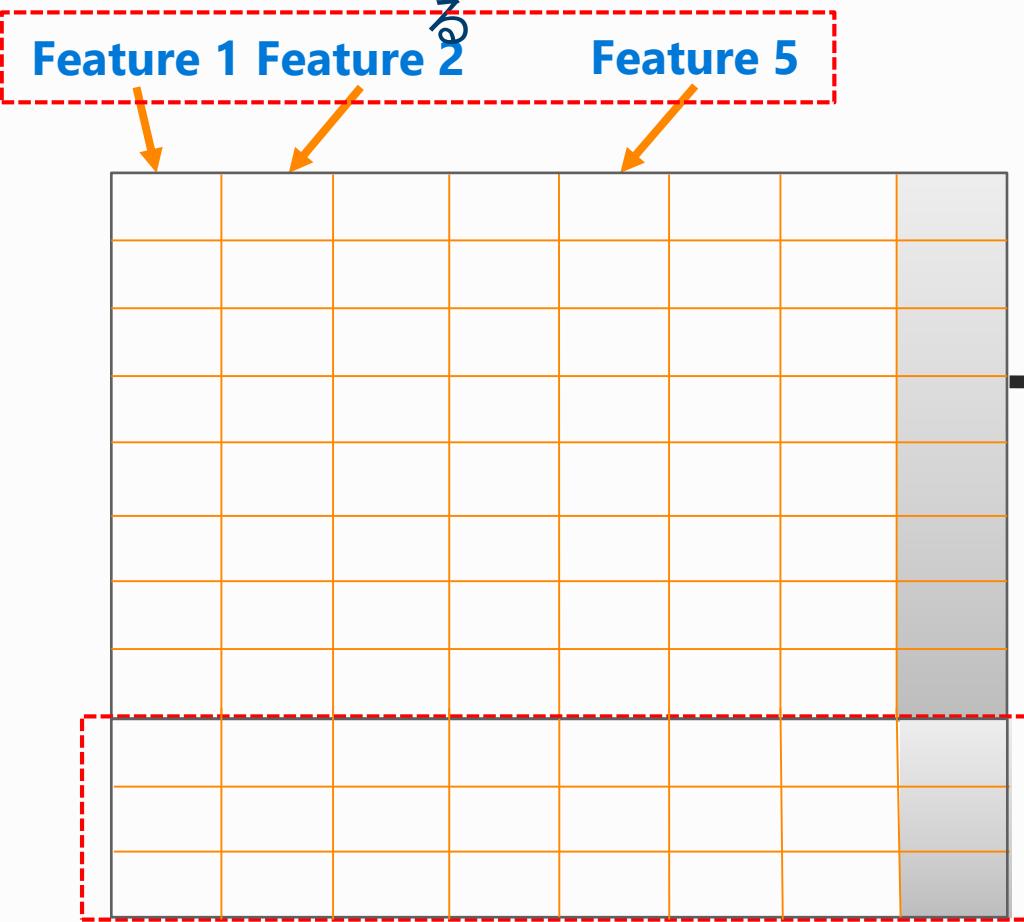


# Model の検証



# Model の改善ポイント

1) 別の features を選択する



2) サンプルデータを追加する

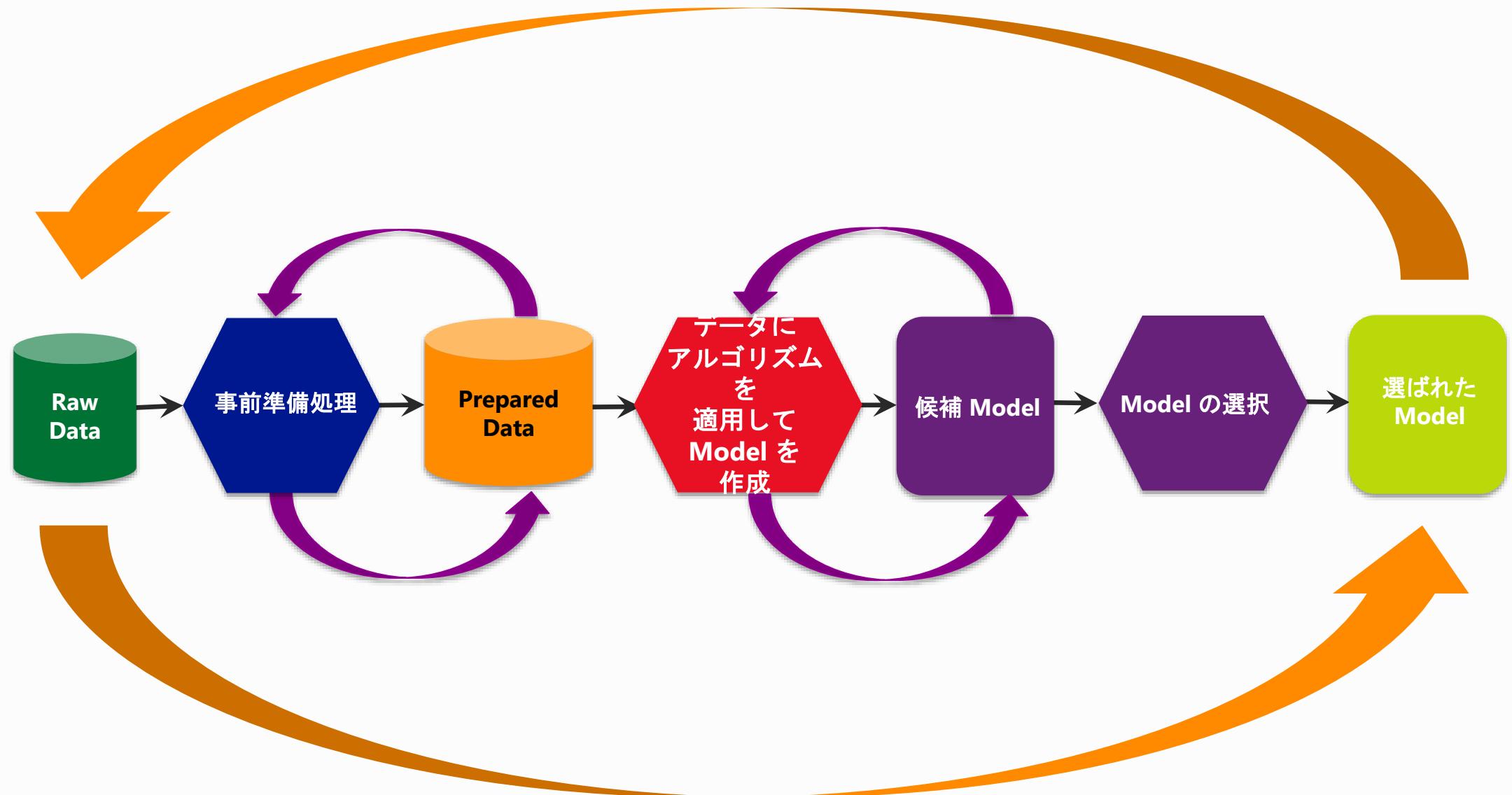
3) アルゴリズムのパラメータの  
変更。もしくは、別のアルゴリ



学習用アルゴリズム

候補  
モデル

# ビジネスの変化 = Model の更新!



どんな種類の  
処理が出来る  
の?

何でも出来るわ  
けじゃないで  
しょ?



# 機械学習の主なタスク

数値予測

ラベル予測

回帰分析  
Regression

分類  
Classification

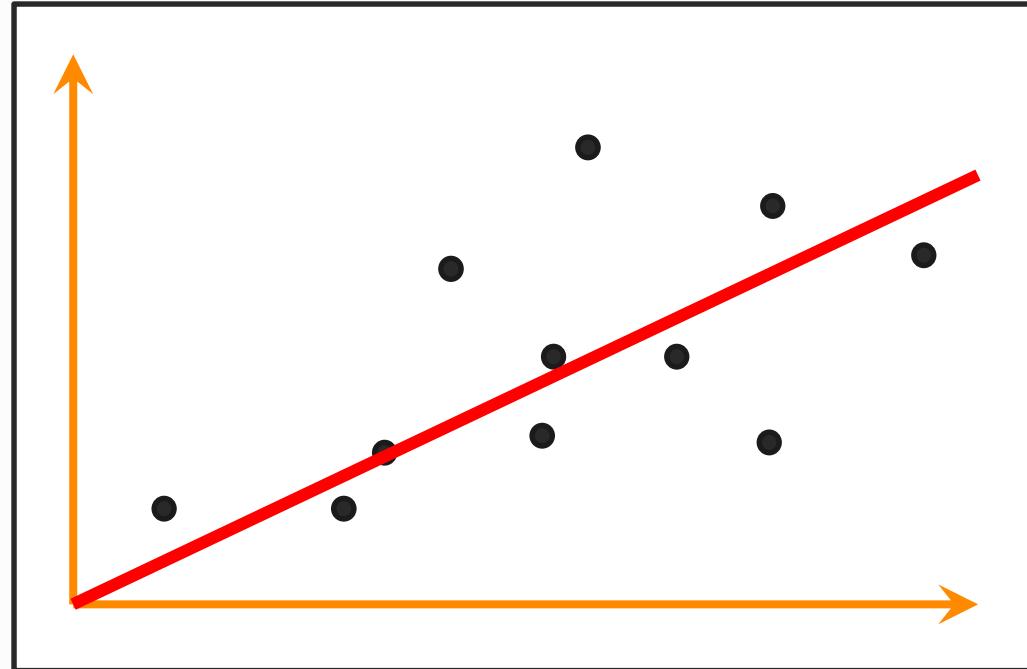
クラスタリング  
Clustering

- ・売上予測
- ・需要予測
- ・品質管理

- ・不良品分析
- ・故障予測
- ・チャーン分析
- ・販促効果測定
- ・与信分析

- ・セグメンテーション
- ・顧客グルーピング
- ・メール キャンペーン

# 回帰 “Regression”



**Goal:** 値を予測する

教師あり学習

ゴールの例:

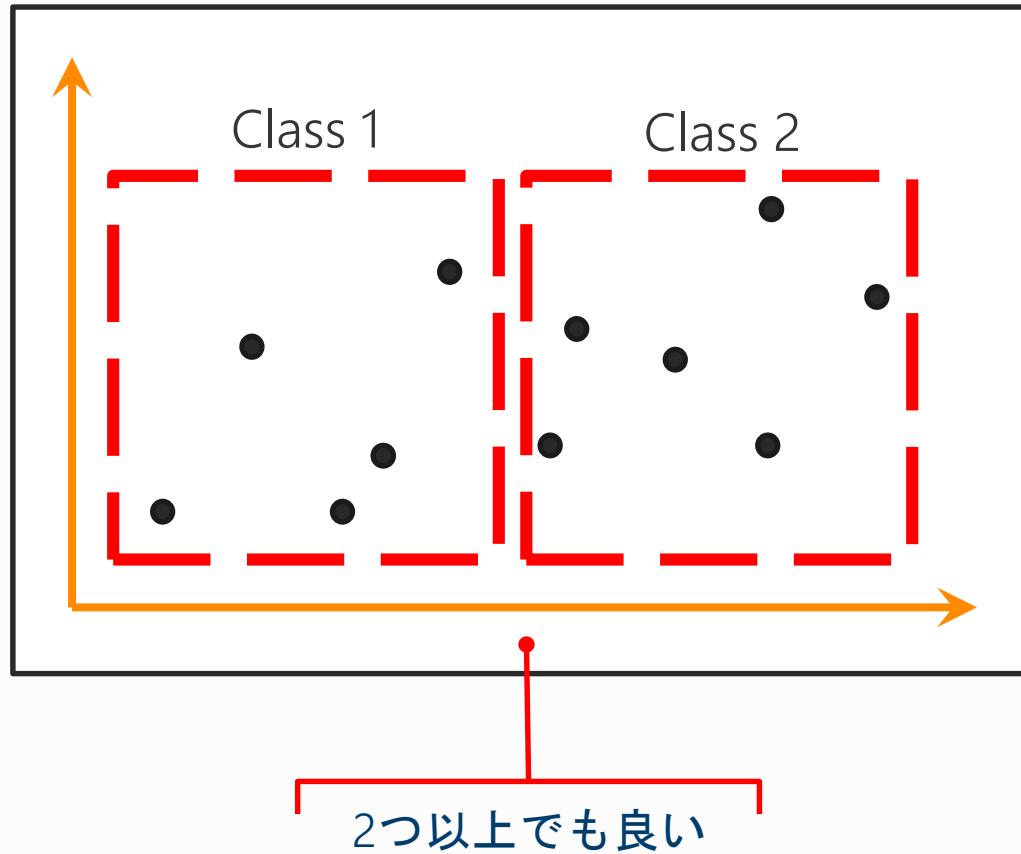
来月この製品は何個売れるか?

# 車の価格予想

- 車の各種情報から車の価格を予測する
  - ブランド
  - 燃料タイプ(ガソリン, ディーゼル)
  - 吸気タイプ(ターボ, NA)
  - ドアの数(2ドア, 4ドア)
  - 車体(セダン, ワゴン, ハードトップ)
  - 駆動方式(FWD, RWD, 4WD)
  - エンジン位置(フロント, リア)
  - ホイールの大きさ
  - シリンダーの数
  - 車体の長さ, 幅, 高さ
  - など



# 分類 “Classification”



Goal: 分類を予測する

教師あり学習

ゴールの例:

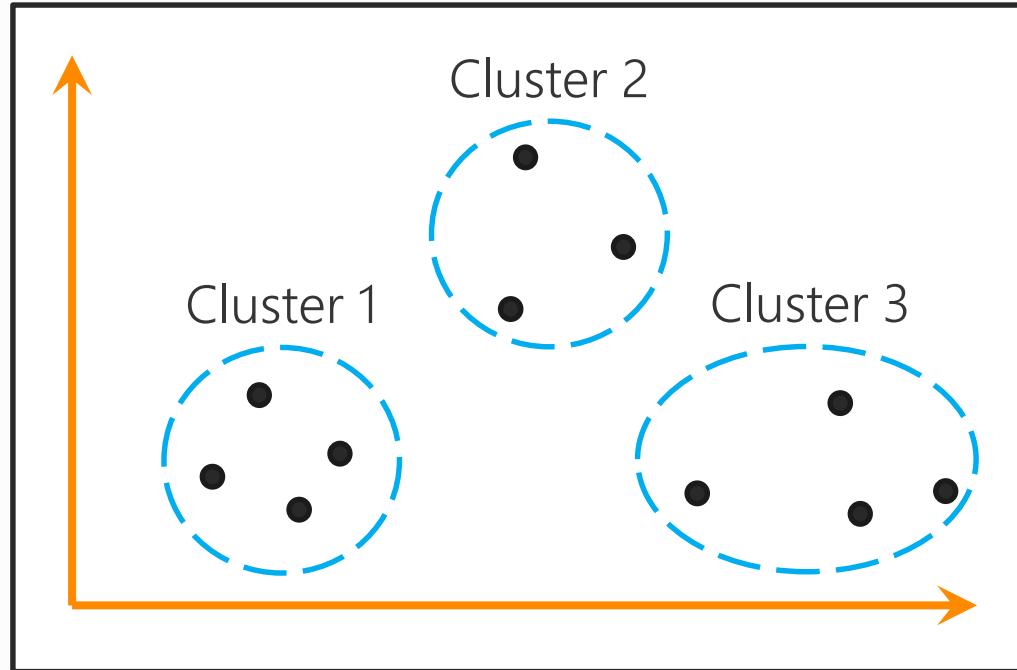
このクレジットカードは不正か?

# DCモータの製造不良原因予測

「不良種類(成功含む)」を  
予測する分析器を作成

工程	① プレス	② 研磨	③ 捲線	④ 溶接	⑤ 組立	検査
装置	プレス機械	研磨装置	自動コイル捲線機	溶接機	人手作業	検査装置
治工具	金型 (電機子軸用)					
部品	電磁鋼板		銅線 (巻線)	整流子		
PLC と 検査装置 から 取得する データ	<ul style="list-style-type: none"> <li>- シリアルNo</li> <li>- プレス装置コード</li> <li>- 金型コード</li> <li>- 鋼板コード</li> <li>- 鋼板ロット</li> <li>- 加圧力</li> <li>- 上下面平行度</li> <li>- 上下運動直角度</li> <li>- 着手・完了時刻</li> <li>- プレス時間</li> <li>- 装置動作状態</li> </ul>	<ul style="list-style-type: none"> <li>- シリアルNo</li> <li>- 研磨装置コード</li> <li>- 砥石回転数</li> <li>- 試料版回転数</li> <li>- 研磨圧力</li> <li>- センター位置ずれ</li> <li>- ワーク不平衡率</li> <li>- 研削液pH</li> <li>- 着手・完了時刻</li> <li>- 研磨時間</li> <li>- 装置動作状態</li> </ul>	<ul style="list-style-type: none"> <li>- シリアルNo</li> <li>- 捲線装置コード</li> <li>- 銅線コード</li> <li>- 銅線ロット</li> <li>- モータトルク</li> <li>- 張力</li> <li>- コイル巻付け速度</li> <li>- 着手・完了時刻</li> <li>- 捲線時間</li> <li>- 装置動作状態</li> </ul>	<ul style="list-style-type: none"> <li>- シリアルNo</li> <li>- 溶接装置コード</li> <li>- 整流子コード</li> <li>- 溶接電流</li> <li>- 押下圧力</li> <li>- 溶接部温度</li> <li>- 着手・完了時刻</li> <li>- 通電時間</li> <li>- 装置動作状態</li> </ul>	<ul style="list-style-type: none"> <li>(MES から取得)</li> <li>- シリアルNo</li> <li>- 作業者コード</li> <li>- 着手・完了時刻</li> <li>- 組立時間</li> </ul>	<ul style="list-style-type: none"> <li>- シリアルID</li> <li>- 検査結果</li> <li>- 不良種類</li> <li>- 検査測定値</li> <li>- 検査終了時刻</li> </ul>

# クラスタリング “Clustering”



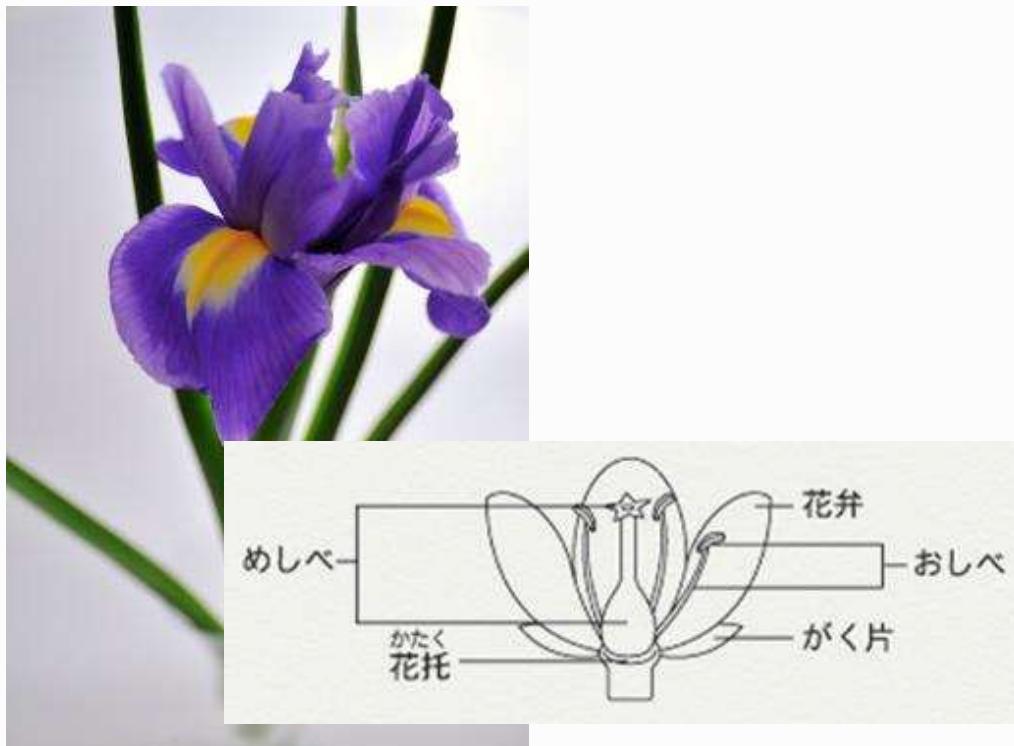
**Goal:** 構造を理解する  
教師無し学習

ゴールの例:

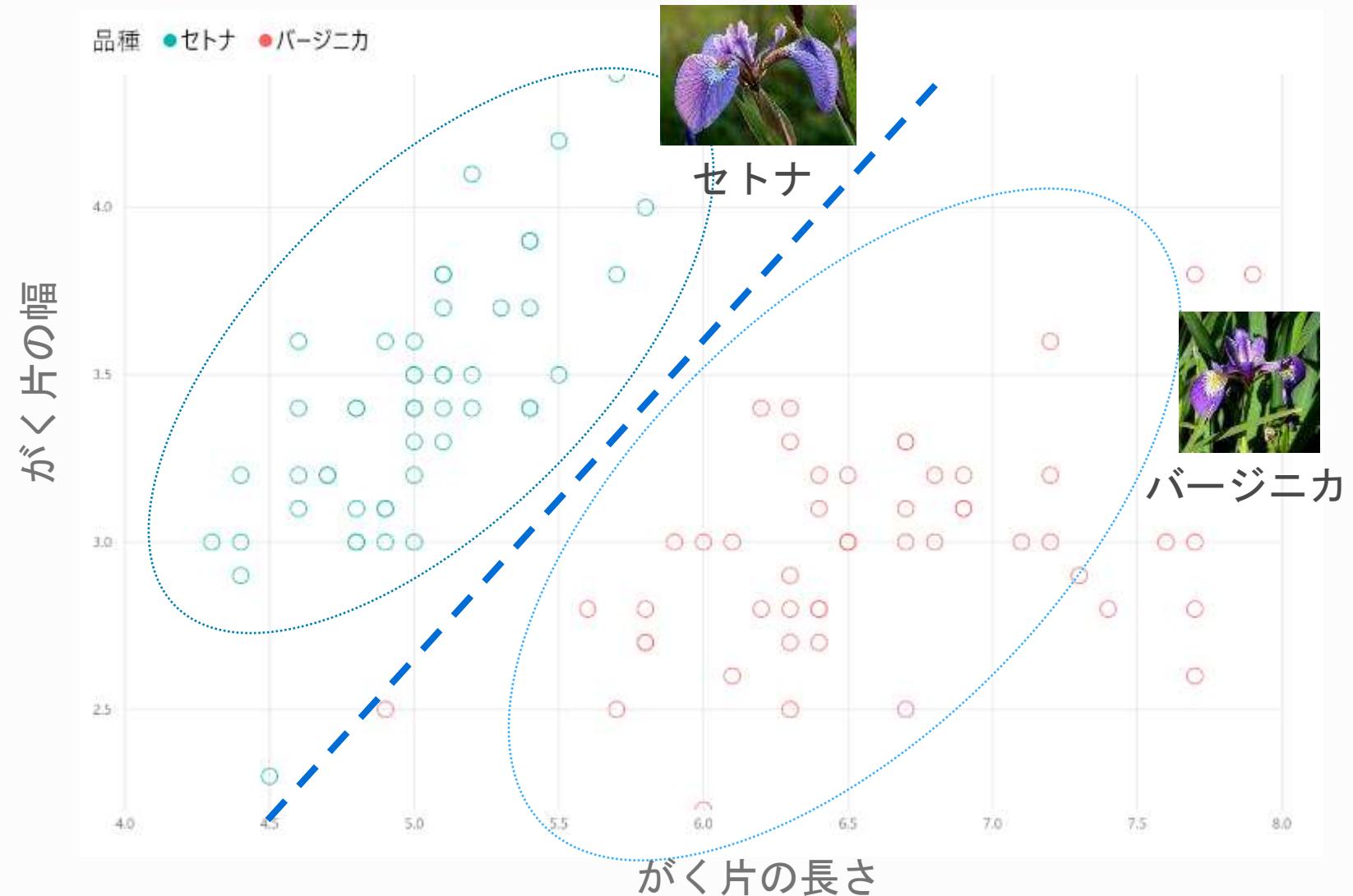
私たちの顧客セグメントは  
どうなっているのか?

# あやめの品種分類

- あやめの「がく片の長さ」「がく片の幅」のみの情報でデータの類似性から100個のあやめを2種類に分類する
- 訓練用の答えデータを必要としない教師なし学習



# あやめ品種分類



# Model

は、関数と一緒に。  
-> 二つの事しかできない。



# Computer Vision Task

## Image Classification

Is there a deer in the image?



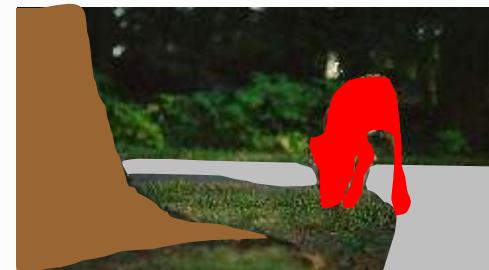
## Object detection

Where is the deer in the image?



## Image segmentation

Where exactly is the deer? What pixels?



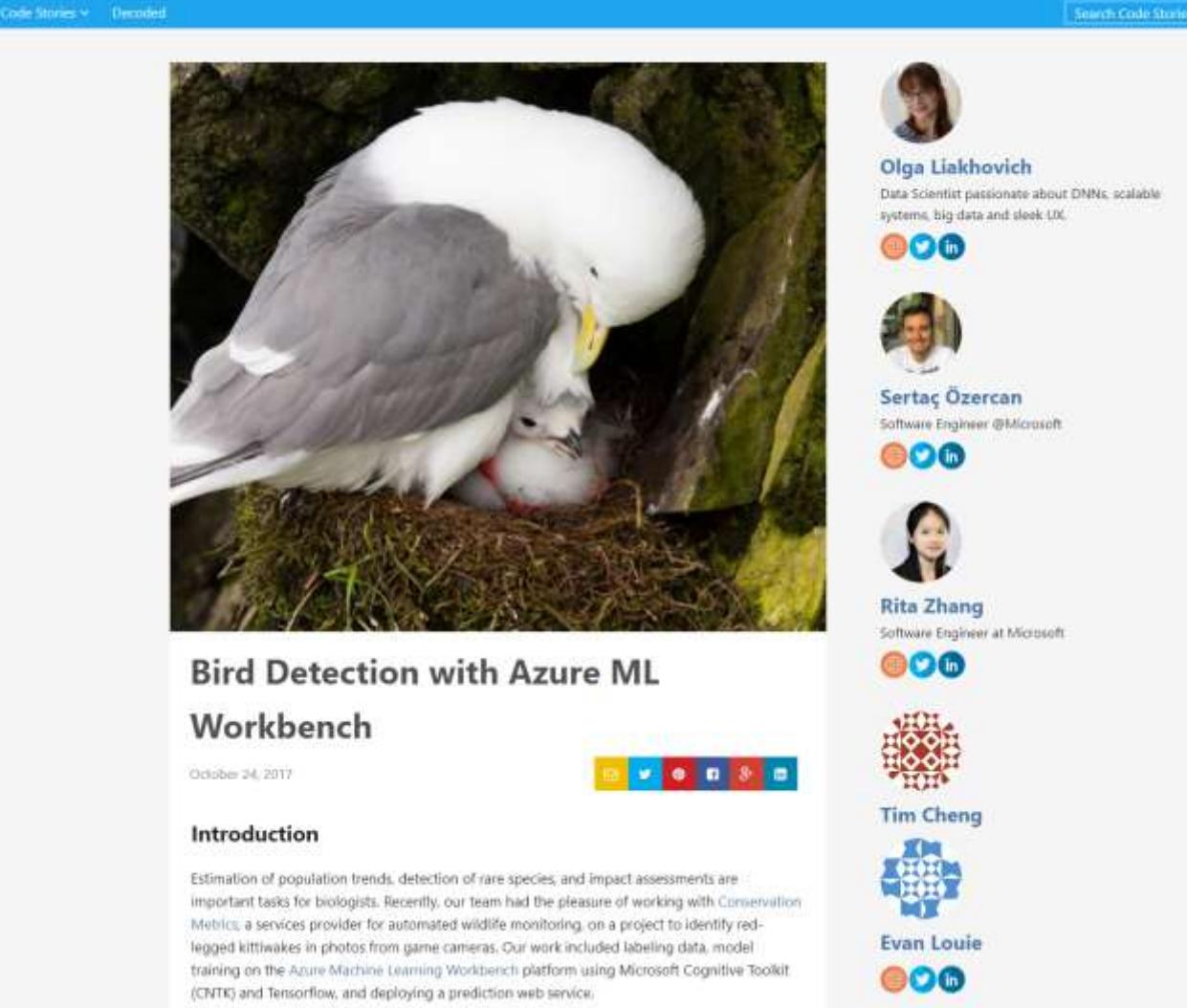
## Image Similarity

Which images are similar to the query image?



# Bird Detection Project

- 離島での野鳥観察作業
  - Object Detection
- Blog:
  - <https://www.microsoft.com/developerblog/2017/10/24/bird-detection-with-azure-ml-workbench/>
- GitHub:
  - <https://github.com/olgaliak/detection-amlworkbench/>



The screenshot shows a Microsoft developer blog post titled "Bird Detection with Azure ML Workbench". The main image is a close-up of a white and grey bird, likely a red-legged kittiwake, sitting on a nest. Below the image, the title "Bird Detection with Azure ML Workbench" is displayed, along with the date "October 24, 2017" and social sharing icons. To the right of the main content, there are profiles for four Microsoft employees: Olga Liakhovich, Sertaç Özercan, Rita Zhang, and Tim Cheng, each with a small profile picture, their names, titles, and social media links.

Code Stories Decoded Search Code Stories



Bird Detection with Azure ML Workbench

October 24, 2017

Olga Liakhovich  
Data Scientist passionate about DNNs, scalable systems, big data and sleek UX.  
[GitHub](#) [Twitter](#) [LinkedIn](#)

Sertaç Özercan  
Software Engineer @Microsoft  
[GitHub](#) [Twitter](#) [LinkedIn](#)

Rita Zhang  
Software Engineer at Microsoft  
[GitHub](#) [Twitter](#) [LinkedIn](#)

Tim Cheng



Estimation of population trends, detection of rare species, and impact assessments are important tasks for biologists. Recently, our team had the pleasure of working with Conservation Metrics, a services provider for automated wildlife monitoring, on a project to identify red-legged kittiwakes in photos from game cameras. Our work included labeling data, model training on the Azure Machine Learning Workbench platform using Microsoft Cognitive Toolkit (CNTK) and Tensorflow, and deploying a prediction web service.

Evan Louie



# Model

は、DLLと一緒に。  
-> 単なるファイル



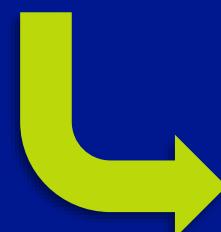
# Model は、ファイルにして扱う



## 学習 = Model 作成

<Python> Modelの保存

```
293  
294  
295  
296  
297  
    # save model to outputs folder  
    z.save('outputs/mnist-cntk.model')  
    z.save('outputs/mnist.onnx', format=C.ModelFormat.ONNX)
```



## 推論 = Model 利用

<C#> Model の利用 (実行)

```
//Evaluate the model  
ModelOutput = await ModelGen.EvaluateAsync(ModelInput);
```



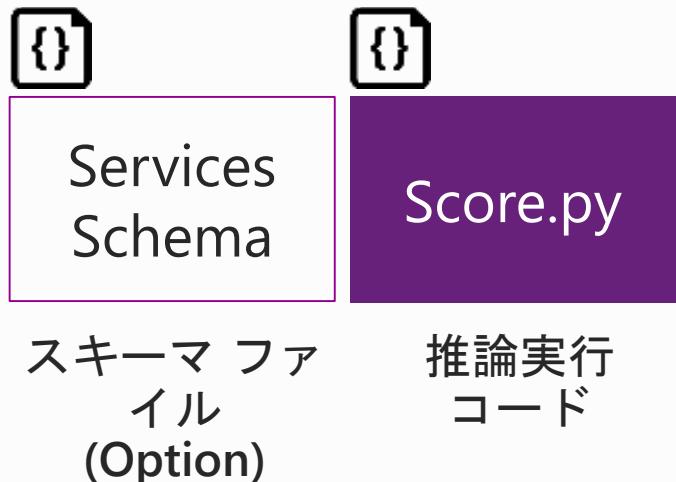
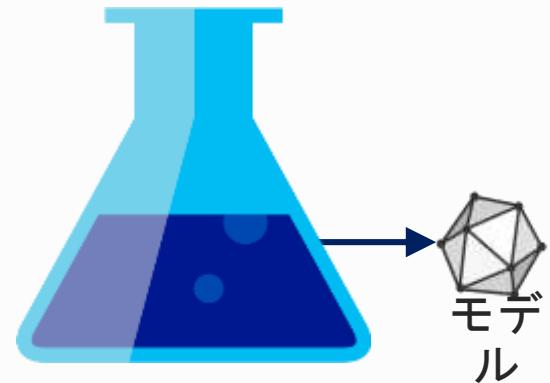
<C#> Model の読み込み

```
1 個の参照  
private async void LoadModel()  
{  
    //Load a machine learning model  
    StorageFile modelFile = await StorageFile.  
        GetFileFromApplicationUriAsync(  
            new Uri($"ms-appx:///Assets/MNIST.onnx"));  
    ModelGen = await MNISTModel.CreateMNISTModel(modelFile);  
}
```

CNTK の例 – Python, C# などのAPIがある

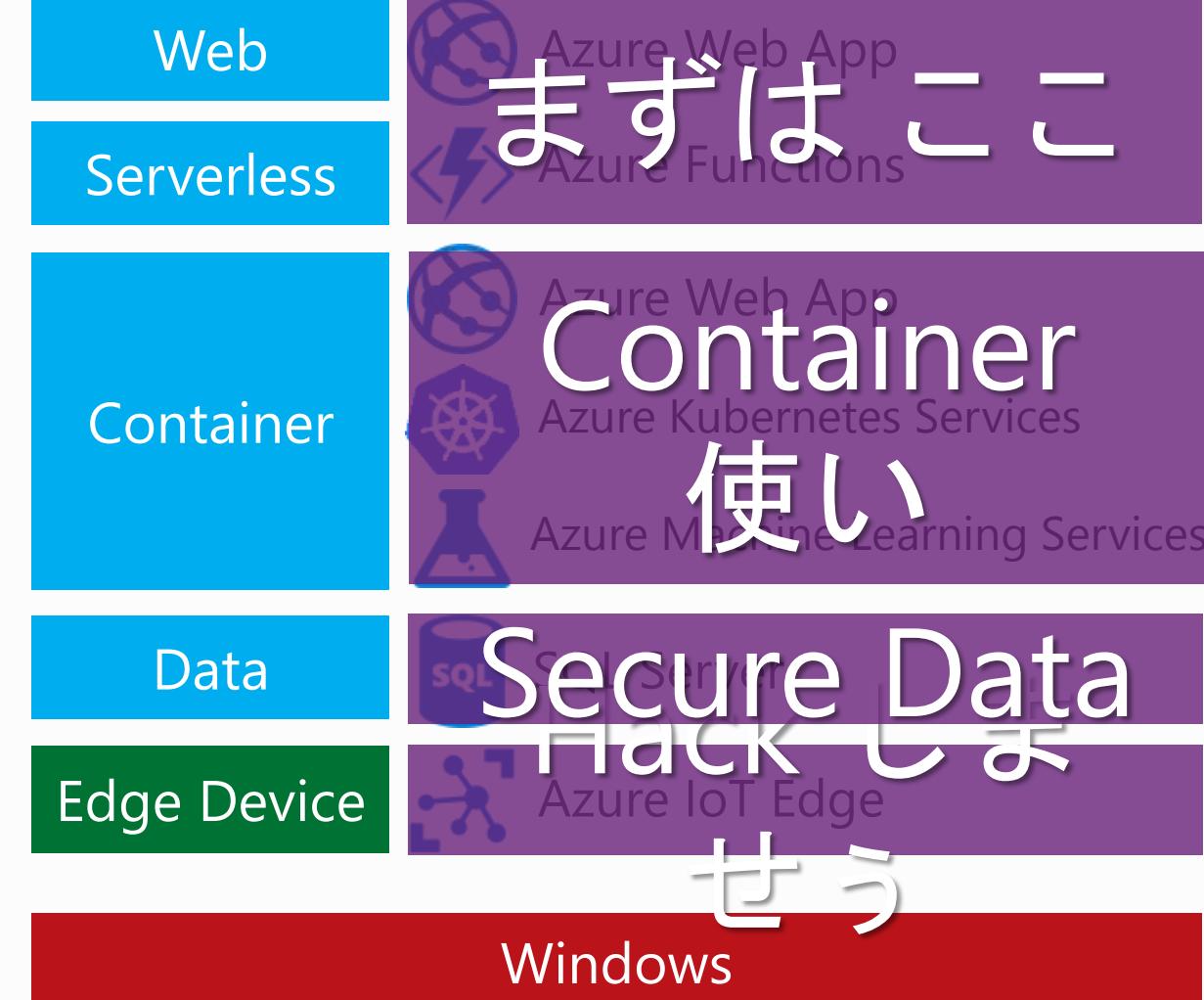
# 推論環境の全体像

## 学習



Machine Learning モデルを Web サービスとしてデプロイする  
<https://docs.microsoft.com/ja-jp/azure/machine-learning/desktop-workbench/model-management-service-deploy>

## 推論



従来の  
システム開発と  
の  
違い



# これまでのシステム開発と違う点

汎用品

データが全て

精度の考え方

# これまでのシステム開発と違う点

汎用品

データが全て

精度の考え方

# プログラミングと機械学習

## Traditional Programming



$$2 + 3 = 5$$

# プログラミングと機械学習

## Traditional Programming



簡単

$$2 + 3 = 5$$

大変...



# プログラミングと機械学習

## Traditional Programming



簡単

$$2 + 3 = 5$$

大変...



# プログラミングと機械学習

## Traditional Programming



## Machine Learning



それぞれの写真: 猫? Yes/No

# プログラミングと機械学習

Program = Algorithm

## Traditional Programming



人が書く  
タスクの仕様の定義  
アルゴリズムは固定  
アルゴリズムは容易に説明できる

ソフトウェアが書く

目的: **汎化**

アルゴリズムはデータに依存

アルゴリズムは時間とともに変わる

## Machine Learning



# プログラミングと機械学習

人は経験 (=Data) から、未知の事象への対応を考え  
られる = 経験値を汎化できる

Tradit

Data -

Program -

人は...

未知の事象に応できる場合がある

ソフトウェアが書く

目的: **汎化**

アルゴリズムはデータに依存

アルゴリズムは時間とともに変

実世界の全てを想定して、  
プログラミングするのは、難しい...



万能なものは無い

# 情報ドメインと評価方法

- エンジン毎に ドメインの得意・不得意がある



# これまでのシステム開発と違う点

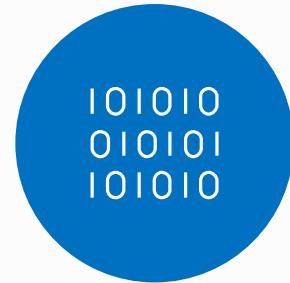
汎用品

データが全て

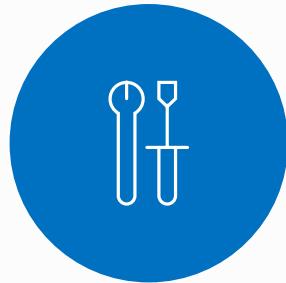
精度の考え方

# データサイエンス プロジェクトのライフサイクル

#azurejp



データの準備



モデル構築・学習

推論  
デプロイメント

んな データ を整備するか?  
競争力のための  
自社にしかないデータが  
活用できるか?

世界中の研究者が  
論文として公表。  
多くの実証 コードも  
公開される。  
最新の技術を利活用

ビジネスフロー全体の中の  
どこで モデルを  
利用すべきか?

# データが定義するプログラム



Windows

Alpha Go Zero

100万

2,000

# 情報と権威



国際学会と情報の**独占**



- 時間的に制限される最新研究成果、知見へのアクセス
- 出版、手紙、人的なネットワーク（人伝）

情報の独占による**権威**の獲得  
(優位な立場の獲得)

# 共有、活用される最新研究

- ・オープンに**共有**される深層学習の研究
  - ・論文、サンプルコード、データセット

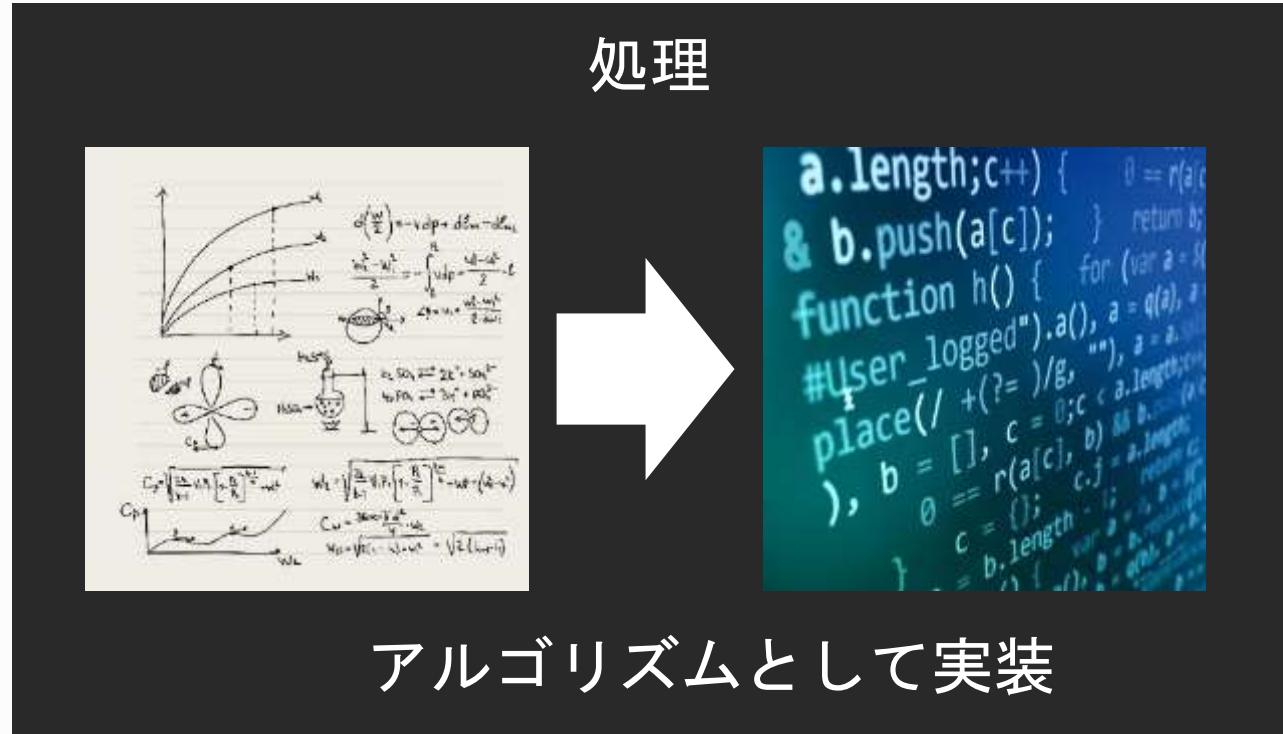
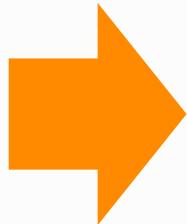


大きく変わる  
**変革の速度**と  
**競争領域**

# 変化への対応を、プログラミングで行うのか？

- ・ 職人による新規性、進歩性を競える職人(?)の世界

# 输入



# 处理



# 出力

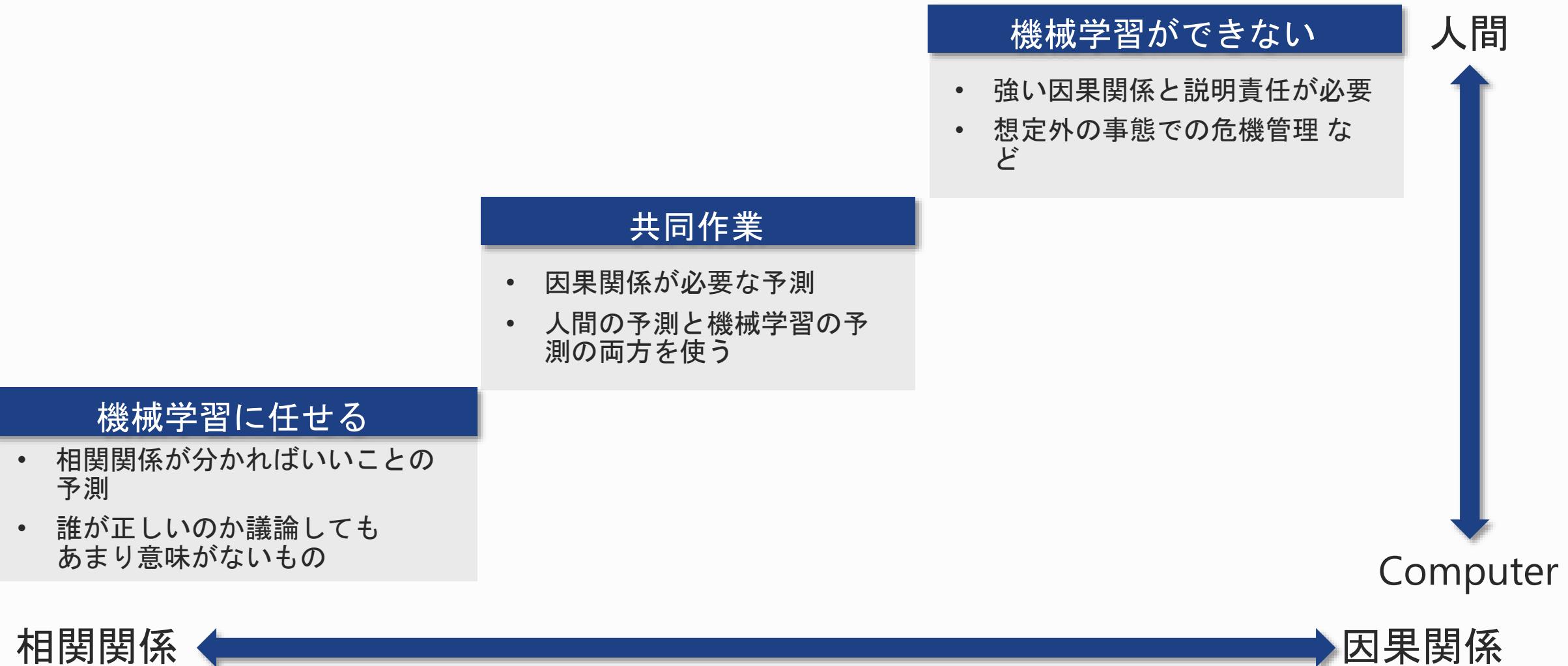
# 変化への対応を、データを元に行うのか？

- ・入力と教師データとしての出力のデータ  
モデルの構築はデータ任せ



機械学習の最大限の可能性は、  
データソース (IoT) との紐づけと デプロイの自動化 (AutoML)

# 相関関係と因果関係



# データの関連性というけれど...

[tylervigen.com](http://tylervigen.com)

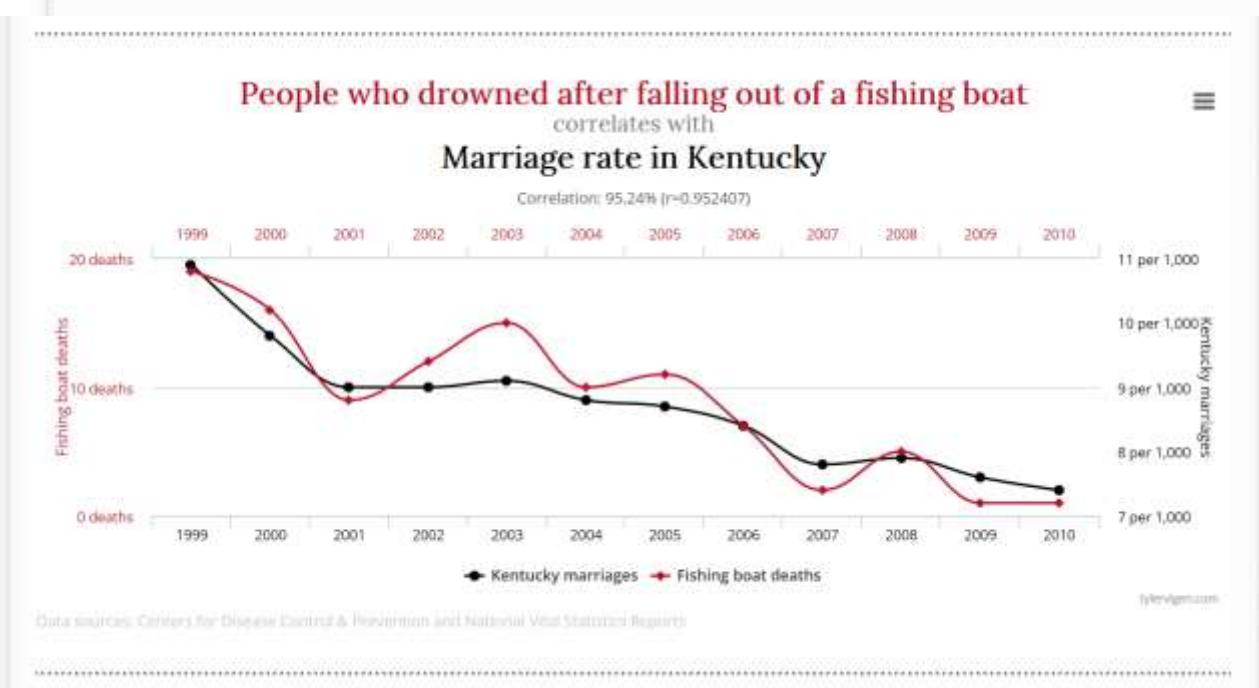
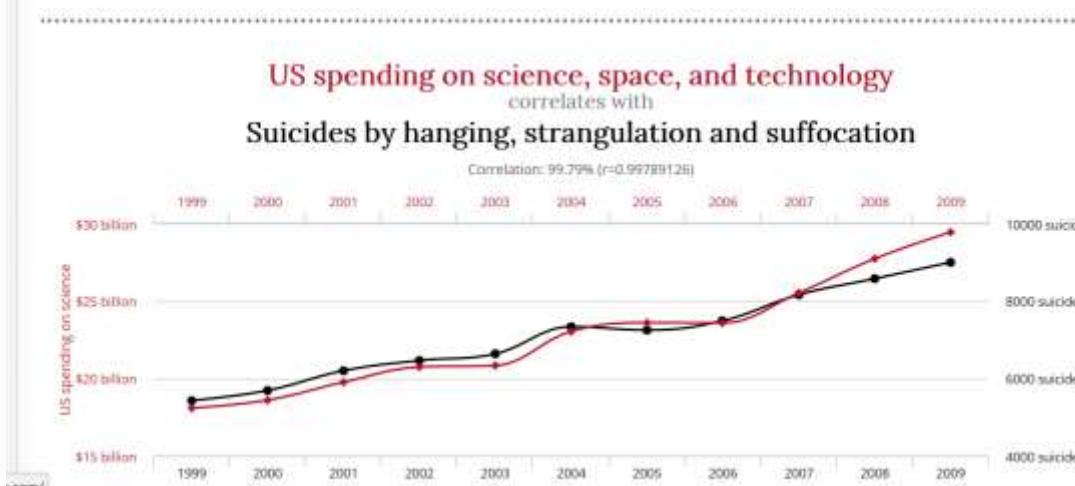
about | twitter | email | subscribe

## Spurious correlations

Now a ridiculous book!

- Spurious charts
- Fascinating factoids
- Commentary in the footnotes

[Amazon](#) | [Barnes & Noble](#) | [Indie Bound](#)



<http://tylervigen.com/spurious-correlations>

# これまでのシステム開発と違う点

汎用品

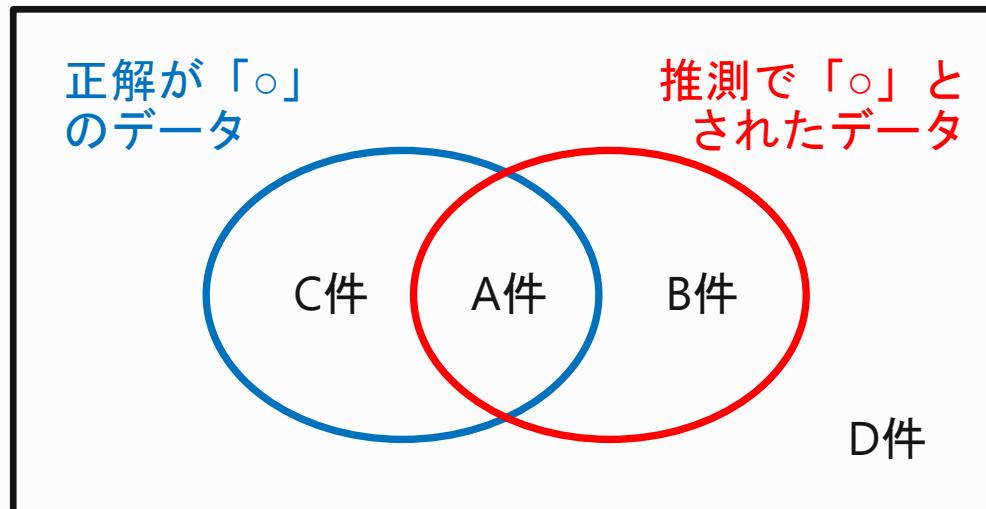
データが全て

精度の考え方

# 分類モデルの評価 = Confusion Matrix

予測結果例

検証用データ		予測で得たクラス	
		○	×
正解の クラス	○	A件	C件
	×	B件	D件



主な評価指標

- ① True Positive (真陽性) : 100%に近いほど良好  
 $\Rightarrow A/(A+C)$
- ② False Positive (偽陽性) : 0%に近いほど良好  
 $\Rightarrow B/(B+D)$
- ③ True Negative : 100%に近いほど良好  
 $\Rightarrow D/(B+D)$
- ④ False Negative : 0%に近いほど良好  
 $\Rightarrow C/(C+D)$
- ⑤ Accuracy (正解率) : 100%に近いほど良好  
 $\Rightarrow 「○」「×」を正しく予測できた割合$   
 $\Rightarrow (A+D)/(A+B+C+D)$
- ⑥ Precision (適合率) : 100%に近いほど良好  
 $\Rightarrow A/(A+B)$
- ⑦ Recall (再現率) : 100%に近いほど良好  
 $\Rightarrow ①$ に同じ
- ⑧ F1 Score : 1.0に近いほど良好  
 $\Rightarrow ⑥、⑦の複合指標$   
 $\Rightarrow 2 \times (⑥ \times ⑦) / (⑥ + ⑦)$

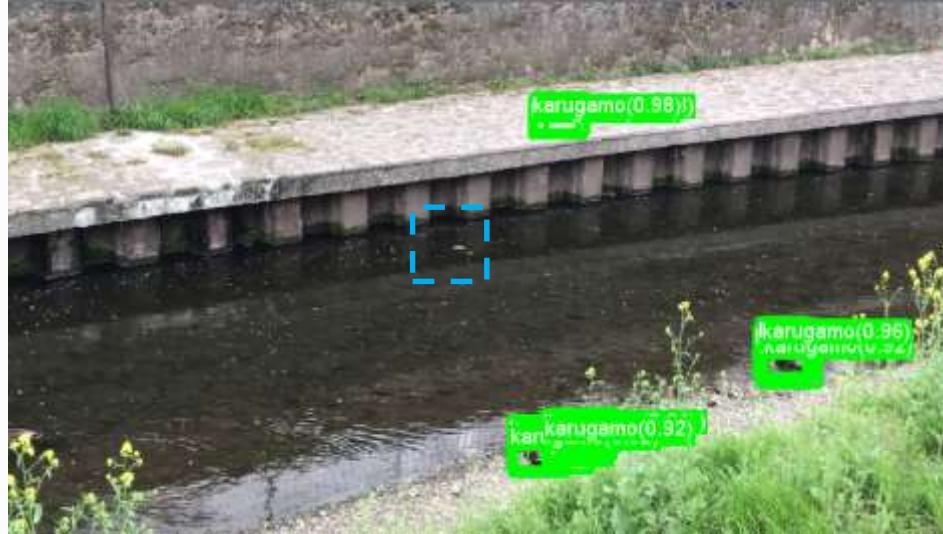
# Confusion Matrix for karugamo



karugamoが写っているのに、  
モデルは推定できなかつ  
た  
→ モデルの見逃し

	あり[予測]	なし[予測]
あり[正解]	XX	XX
なし[正解]	XX	XX

# Confusion Matrix for karugamo



次フレー  
ム



Karugamo でないもの  
に、  
Karugamo と推定  
► モデルの過検知？

	あり[予測]	なし[予測]
あり[正解]	XX	XX
なし[正解]	XX	XX

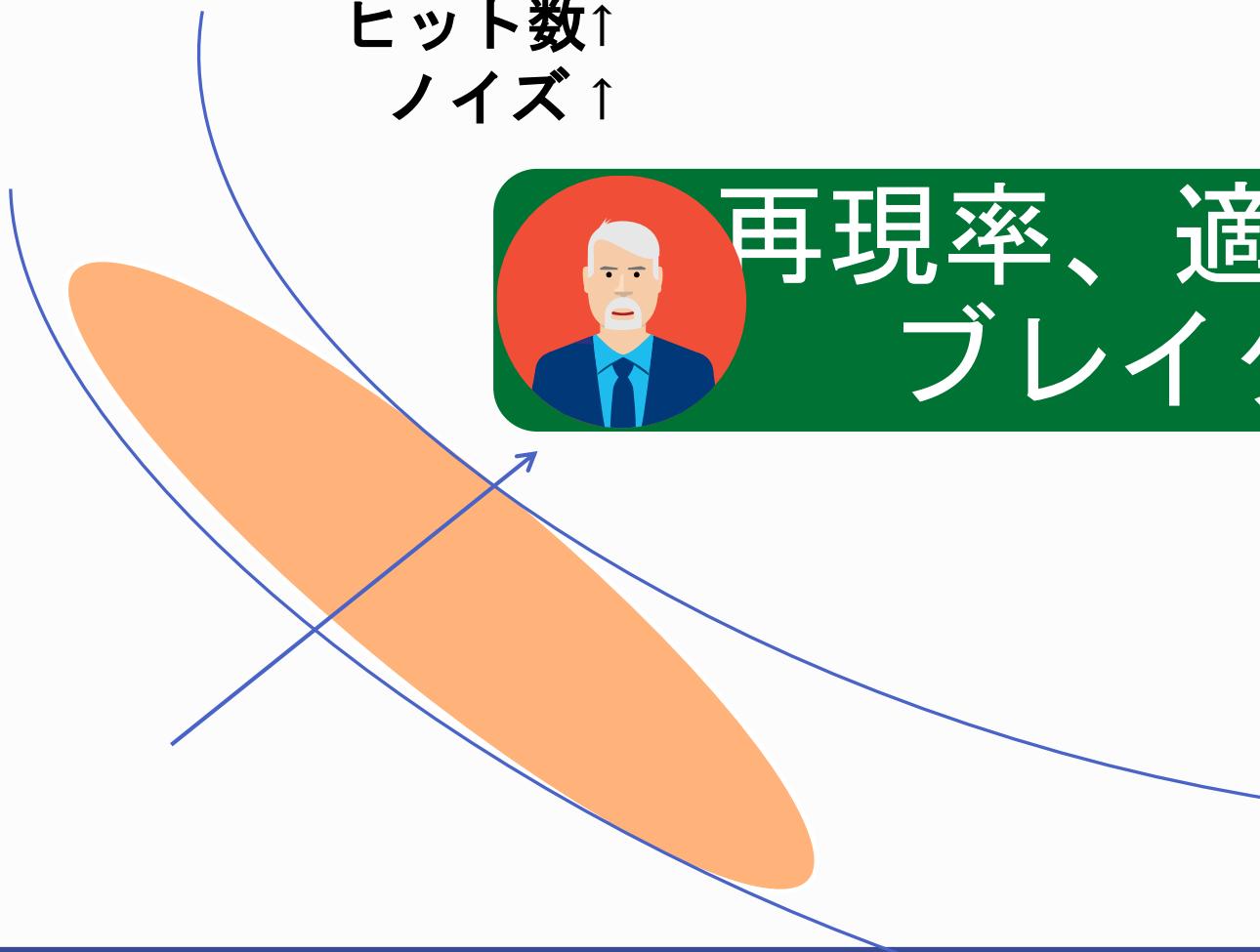
# 何を優先するかは、一概に決められない!

## 再現率(Recall)

ヒット数↑  
ノイズ↑



再現率、適合率の最適な  
ブレイクポイント



## 適合率 (Precision)

精度↑  
漏れ↑

# 同じソフトウェアでも、こんなに違う

	プログラミング	機械学習
アプローチ	演繹的	帰納的。つまりブラックボックスは残る
機能保証 (≒ 精度): Function Test	可能	訓練データ次第。ただ、統計の域を出ない
性能保証: Performance Test	可能	可能
妥当性確認試験: Validation Test	可能	やってみないと、わからぬ

# アルゴリズム





どのアルゴリズムを使って、  
モデルを作ってみればいいか  
見当もつかない!

# 一応説明はあるのですが...

## Microsoft Azure Machine Learning のアルゴリズムの選択方法

2016年8月9日・3分(所要時間)・共同作成者  すべて

「どのような機械学習アルゴリズムを使用すべきか」という質問への答えは、常に「場合による」です。データのサイズ、品質、および性質によって異なります。回答で何を行うかによって異なります。アルゴリズムの数値演算が使用しているコンピューターの命令にどのように変換されるかによって異なります。そして、どれだけ時間があるかによって異なります。最も経験豊富なデータ科学者であっても、試してみる前にどのアルゴリズムが最適か判断することはできません。

### 機械学習アルゴリズム チートシート

Microsoft Azure Machine Learning のアルゴリズム チートシート を使用すると、Microsoft Azure ルゴリズム ライブラリから、予測分析ソリューションに適した機械学習アルゴリズムを選択できます。その使用方法について説明します。

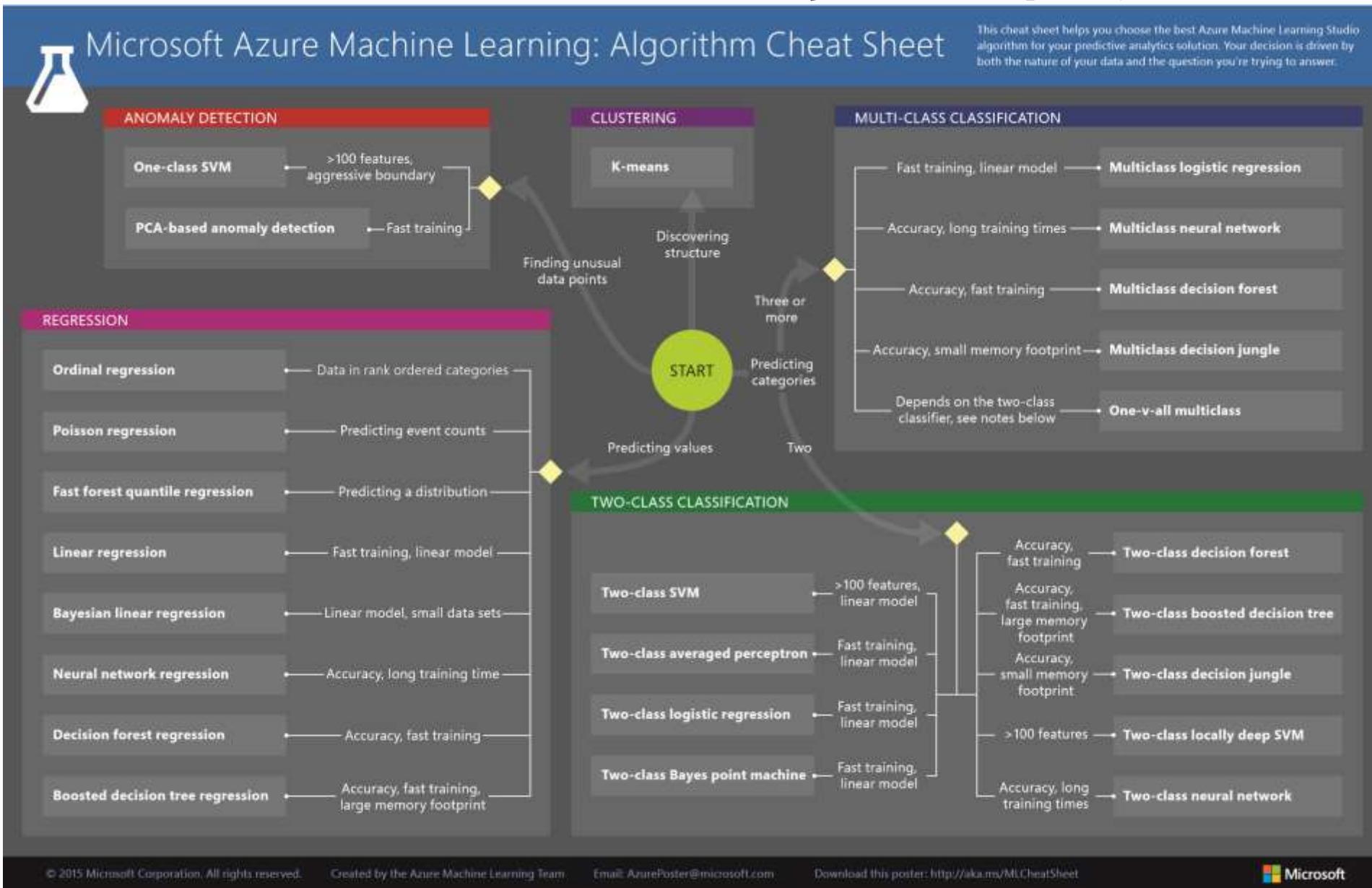
#### メモ

チートシートをダウンロードし、それを見ながらこの記事を読むには、「[Microsoft Azure Machine Learning アルゴリズム チートシート](#)」にアクセスしてください。

アルゴリズム	精度	トレーニング時間	線形性	パラメータ	メモ
<b>2クラス分類</b>					
ロジスティック回帰	●	●	●	5	
デシジョン フォレスト	●	○	●	6	
デシジョン ジャングル	●	○	●	6	低メモリ フットプリント
ブースト デシジョンツリー	●	○	●	6	大メモリ フットプリント
<b>多クラス分類</b>					
ロジスティック回帰	●	●	●	5	
デシジョン フォレスト	●	○	●	6	
ニューラル ネットワーク	●	●	●	6	低メモリ フットプリント
	●	●	●	9	追加カスタマイズ可能
<b>平均化</b>					
サポートベクター マシン	●	●	●	6	選択した 2 クラス法のプロパティを参照してください
	●	●	●	9	
<b>サポートベクター マシン</b>					
ロジスティック回帰	●	●	●	5	
デシジョン フォレスト	●	○	●	6	
ブースト デシジョンツリー	●	○	●	6	
高遅延 フォレスト 分位	●	●	●	6	
ニューラル ネットワーク	●	●	●	9	
ポワソン回帰	●	●	●	6	
序数	●	●	●	9	
<b>異常検出</b>					
サポートベクター マシン	○	○	●	2	大きい特徴セットに特に好適
PCA ベースの異常検出	○	●	●	3	
K-Means	○	●	●	4	クラスタリング アルゴリズム

<https://docs.microsoft.com/ja-jp/azure/machine-learning/machine-learning-algorithm-choice>

# チートシートがある！目安...





論理的背景を…

# 予測精度向上の考え方

- 特徴選択
  - ・ 予測のための変数(特徴)として、より優れたデータ(列)を選択。  
データに関する知識、業務に関する知識およびデータの分析のもとにデータ(列)を選択する
- アルゴリズムのパラメータチューニング
  - ・ 選択したアルゴリズムのパラメータをチューニングする
- アルゴリズムの選択
  - ・ 適切なアルゴリズムを選択
- 再学習
  - ・ データ量を増やしたり最新データを使用して再学習させる

# 特徴選択 (1)

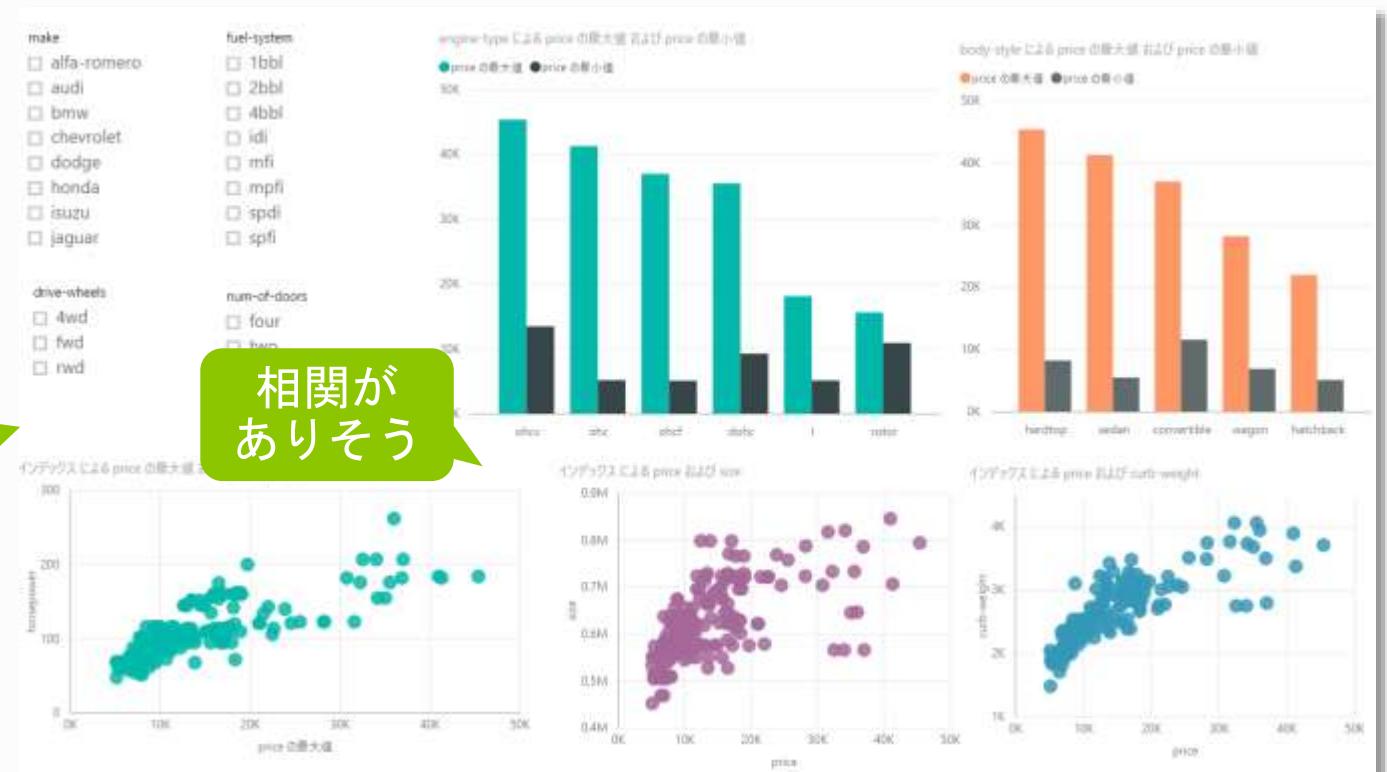
基礎集計によってデータを理解して特徴を発見することが重要。  
予測対象との相関を見つけ、  
説明変数として採用することで精度を上げることができる可能性がある

自動車の価格データ



可視化

データを可視化することで予測対象との相関を探す



相関が  
ありそう



予測対象の説明変数として採用

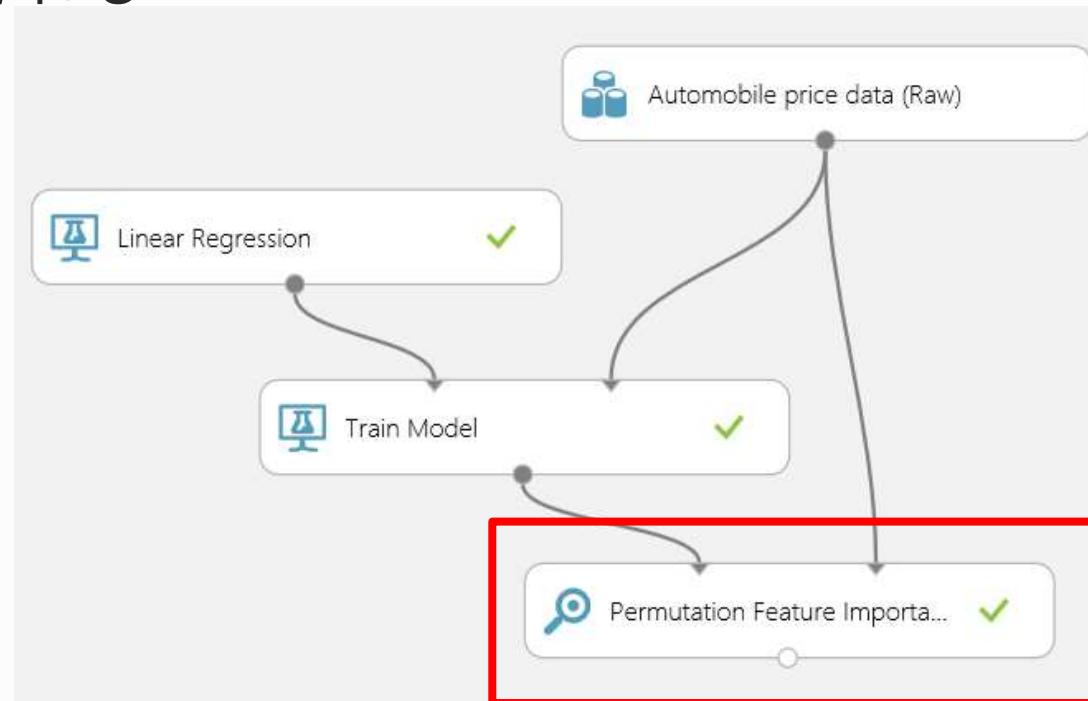


Microsoft Power BI での可視化例

# 特徴選択 (2)

## Permutation Feature Importance (PFI)

それぞれの説明変数の重要度をスコアリングするアルゴリズム。重要度は説明変数の値を変化させたときに、どの程度目的変数に影響したかを持って計算される



rows	columns	
25	2	
view as	Feature	Score
	compression-ratio	3074.528083
	make	3037.526148
	curb-weight	3003.680839
	normalized-losses	2434.317677
	wheel-base	1660.249452
	fuel-type	1389.741648
	fuel-system	1044.002293
	city-mpg	825.540236
	drive-wheels	717.599109
	length	681.660363
	aspiration	616.228936
	num-of-cylinders	568.684636
	body-style	545.300549

# 特徴選択 (3)

## Filter Based Feature Selection

多くの場合、無関係な特徴、重複した特徴、関連性の高い特徴を排除することで、分類の精度を向上させる

スコアリングメソッドに基づいて目的変数との関連性が高い N 個の説明変数のみに

自動車の価格予測 2017-03-15 > Filter Based Feature Selection > Features

rows	columns						
1	25						
<hr/>							
price	engine-size	curb-weight	horsepower	width	highway-mpg	city-mpg	length
1	0.888778	0.835368	0.812453	0.754649	0.719178	0.706618	0.695928

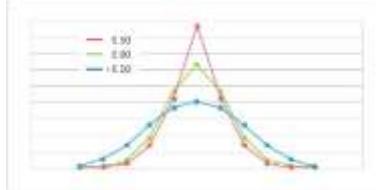
### スコアリングメソッド

- ピアソンの相関関係
- 相互情報量
- ケンドールの相関関係
- スピアマンの相関関係
- カイニ乗
- フィッシャースコア
- カウントベース

# 統計的手法による予測

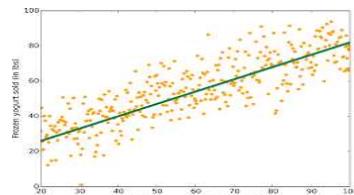
基本統計量の算出

データの傾向を知る



↓  
相関分析

2変数間の因果関係を理解する



↓  
回帰分析

過去のデータから、数値を予測する

$$y = f(x)$$

# 基本統計量 – 都度見返しましょう

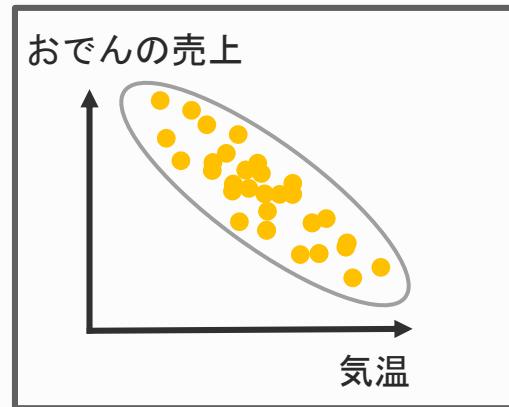
次数	種類	代表値	内 容	
1 次	位置	最大値	もっとも大きな値	
		最小値	もっとも小さな値	
		最頻値（モード）	出現する回数が最も多い値	
		中央値（メジアン）	値を大小の順に並べた場合に中央にくる値（個数が偶数の場合は中心の2値の平均）	
		相加平均	値を合計して個数で割った数	
	平均	相乗平均	値の積の $n$ 乗根（n:値の個数）	
		調和平均	値の逆数の平均の逆数	
		中間項平均	特異値を除いた値の平均	
		加重平均	重みをかけて合計した値を重みの合計で割ったもの	
	バラツキ	平均偏差	値と平均の差の絶対値の平均	
2 次		標準偏差	値と平均の差の平方の平均の平方根	
		分散	値と平均の差の平方の平均	
3 次	偏り	歪 度	値が正負どちらの方向に外れているかを表す値	
4 次	集中度	尖 度	データの平均値への集中の程度を表す値	

# 相関分析

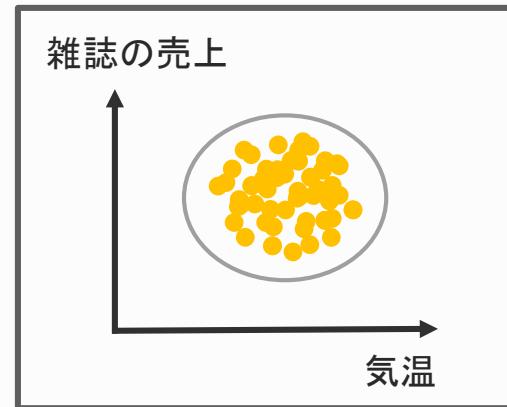
- 2つの変量の間の関係を相関という
  - 散布図を作成することで相関の有無が可視化できる

相関の3つのパターン

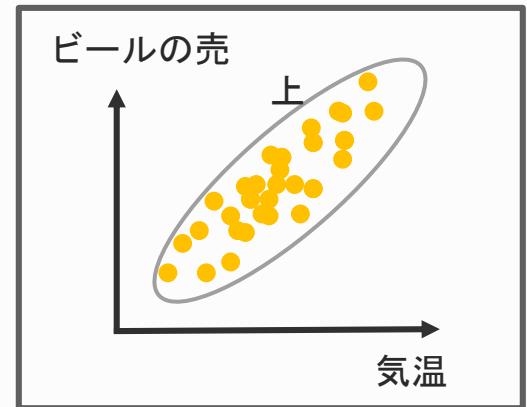
- ① ある変量が増大すると、もう一方の変数も増大 ⇒ 正の相関
- ② ある変量が増大しても、もう一方の変数には無関係 ⇒ 相関なし
- ③ ある変量が増大すると、もう一方の変数は減少 ⇒ 負の相関



負の相関



相関なし

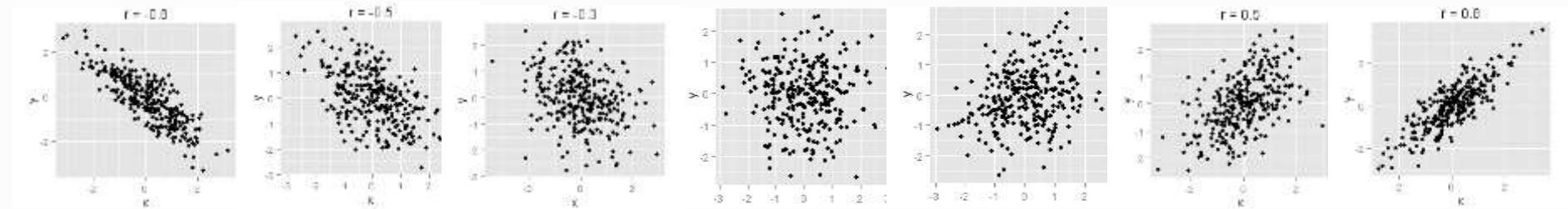


正の相関

# 相関係数

- 2つの変量(XとY)についてどれくらい関係が強いか?というものを示す定量的な数値

**相関係数 = 共分散 ÷ (Xの標準偏差 × Yの標準偏差)**



負の相関関係  
強

正の相関関係  
強

負の相関	解釈
-0.1 ~ 0	無相関
-0.3~ -0.1	弱い負の相関関係
-0.7~ -0.3	中程度の負の相関関係
-1~ -0.7	強い負の相関関係

正の相関	解釈
0~0.1	無相関
0.1~0.3	弱い正の相関関係
0.3~0.7	中程度の正の相関関係
0.7~1	強い正の相関関係

# 回帰分析

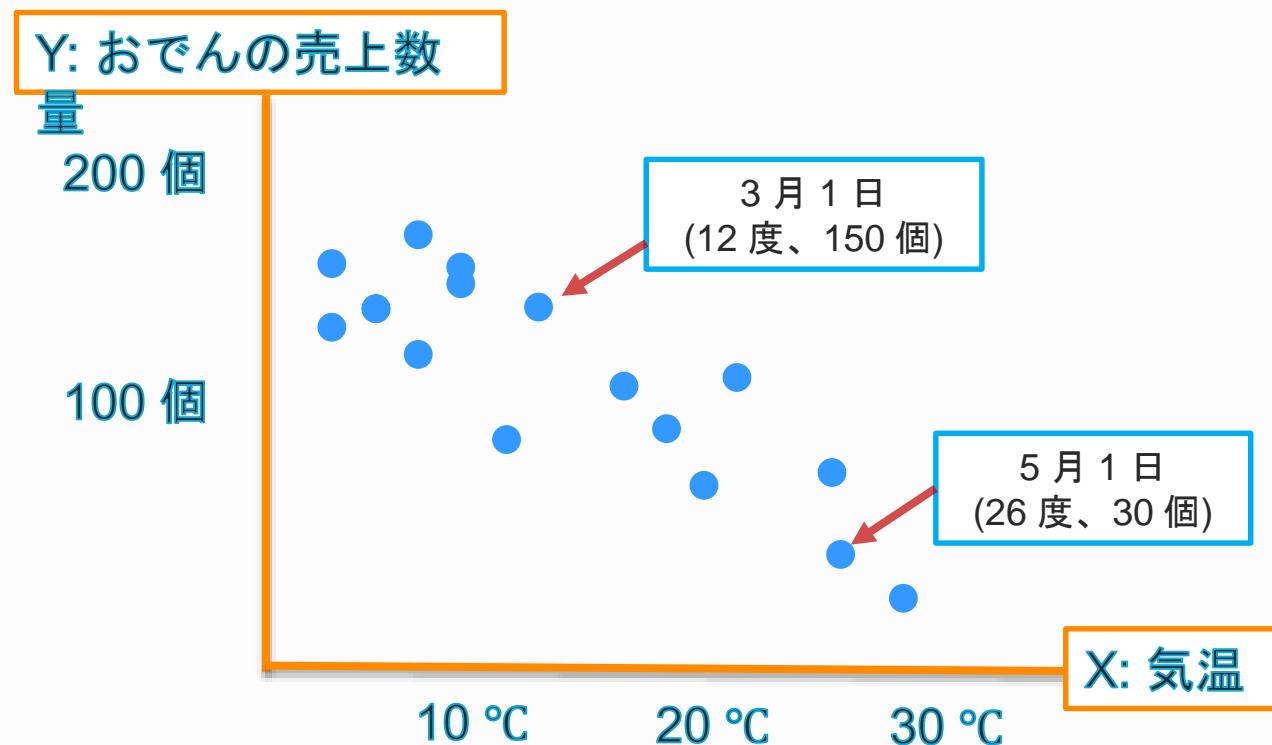
- 既存のデータから関数を作成し、未知のデータに対し、関数値を予測する

$$Y = f(X)$$

X: 説明変数  
Y: 目的変数

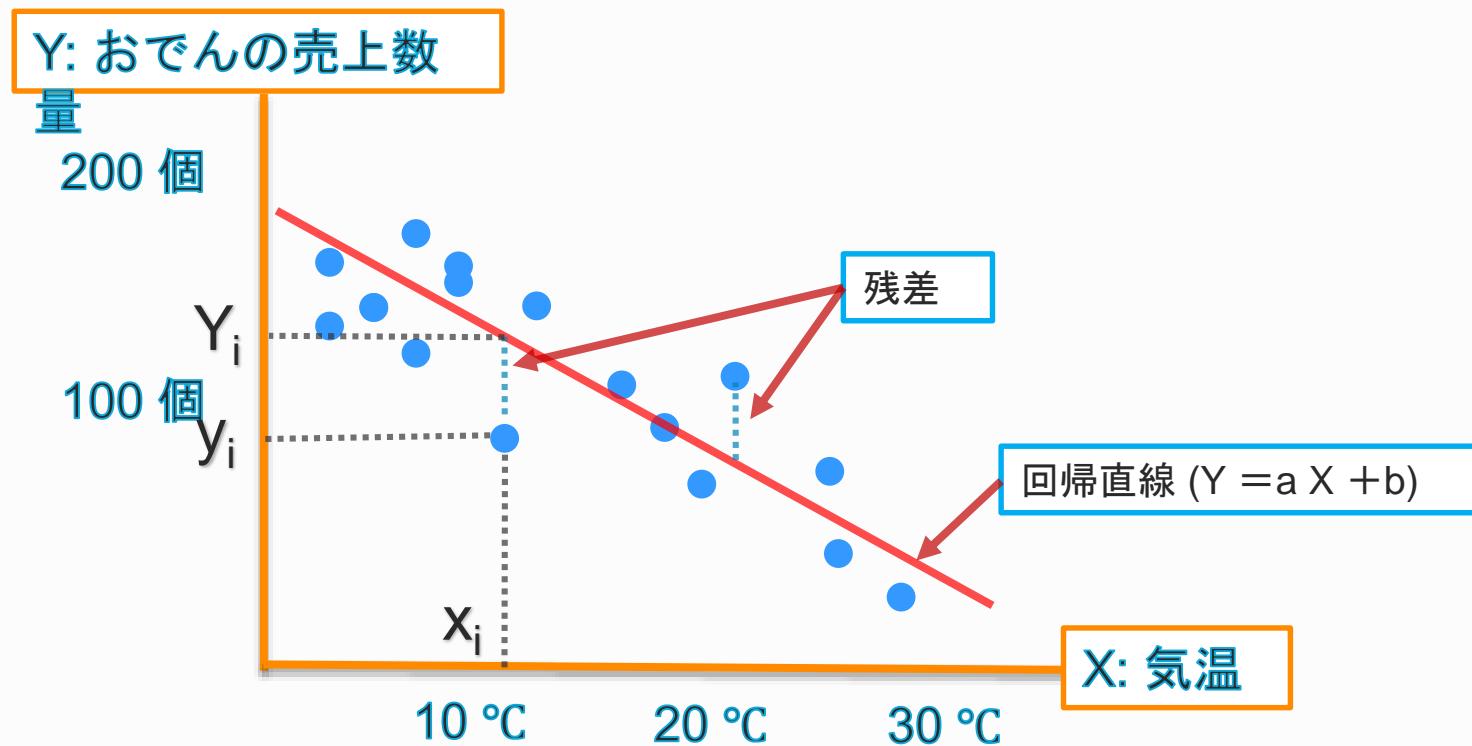
- 回帰分析の例

- 「気温」を説明変数とし、「おでんの売上数量」を目的変数とする



# 回帰分析

- 最小二乗法による回帰直線の決定
  - 全ての点について、残差(誤差の推定量)の2乗を求め、その総和が最小になる直線を決定する



# 回帰分析

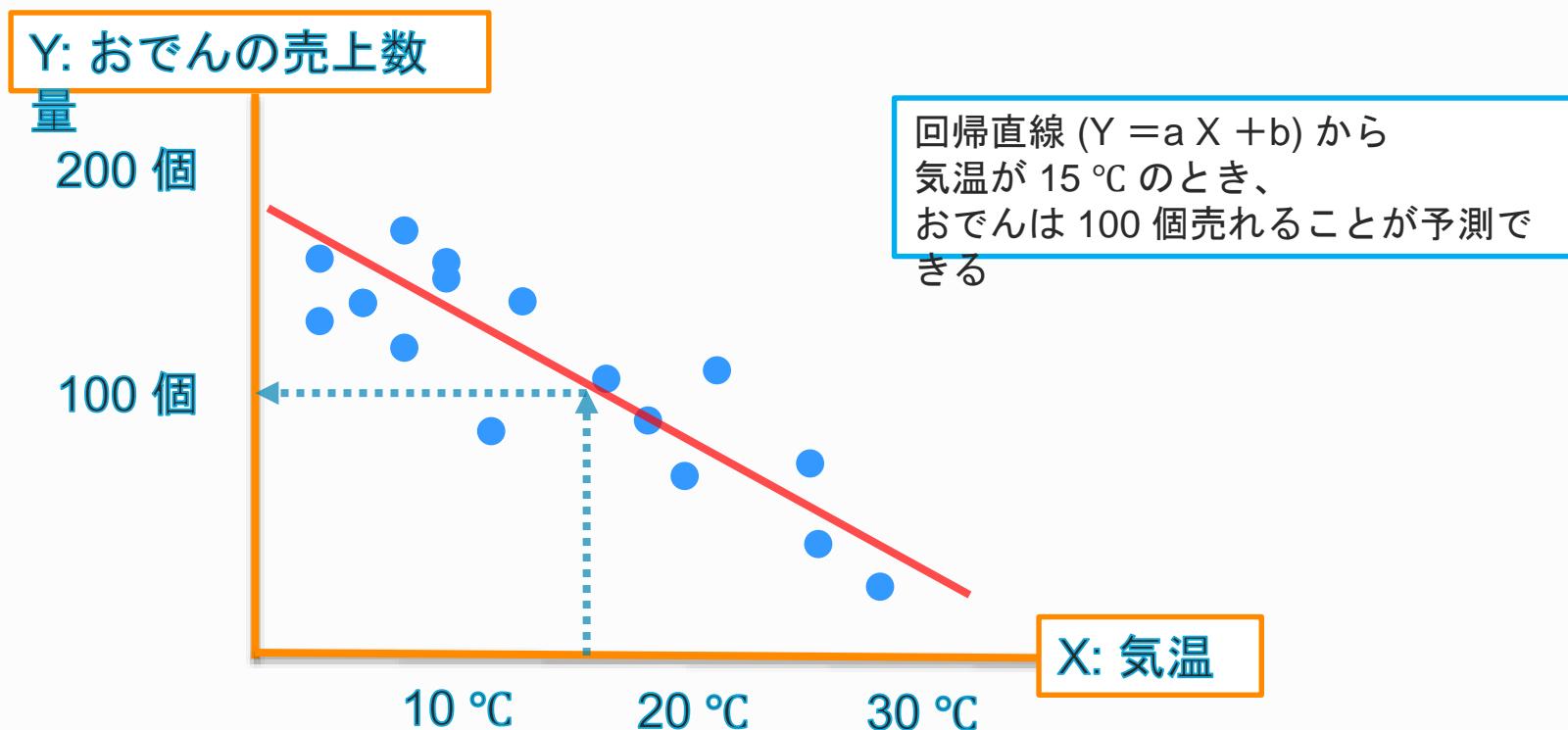
- 既存のデータから関数を作成し、未知のデータに対し、関数値を予測する

$$Y = f(X)$$

X: 説明変数  
Y: 目的変数

- 回帰分析の例

- 「気温」を説明変数とし、「おでんの売上数量」を目的変数とする



# Azure Machine Learning Studio で用意されている分析アルゴリズム

## 回帰分析

線形回帰

ベイズ線形回帰

決定木

ランダムフォレスト

Decision Jungle

ニューラルネットワーク

ポワソン回帰

順序回帰

分位点回帰

## 統計分類

ロジスティクス回帰

ベイズポイントマシン

決定木

ランダムフォレスト

Decision Jungle

ニューラルネットワーク

サポートベクタマシン

Locally-Deep  
サポートベクタマシン

平均化パーセプトロン

## クラスタリング

k平均法 (k-means)

## 異常検知

サポートベクタマシン

PCAベース異常検知

## その他

テキスト分析 (n-gram)

リコメンドエンジン

Open CV

# アルゴリズム - 回帰

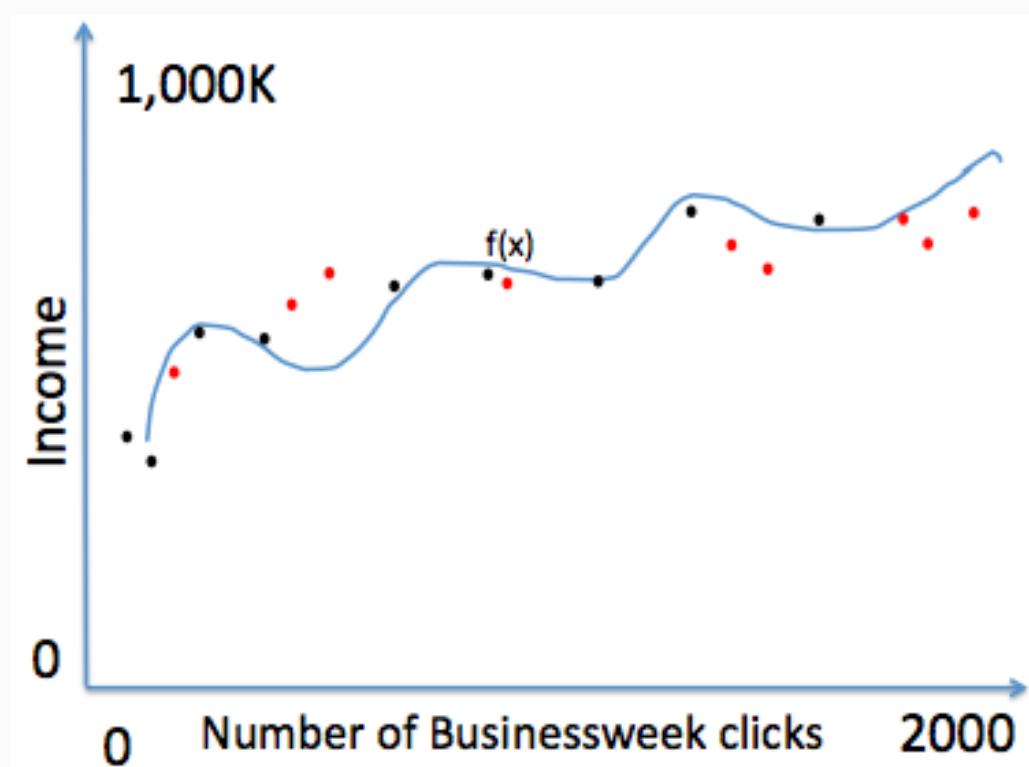
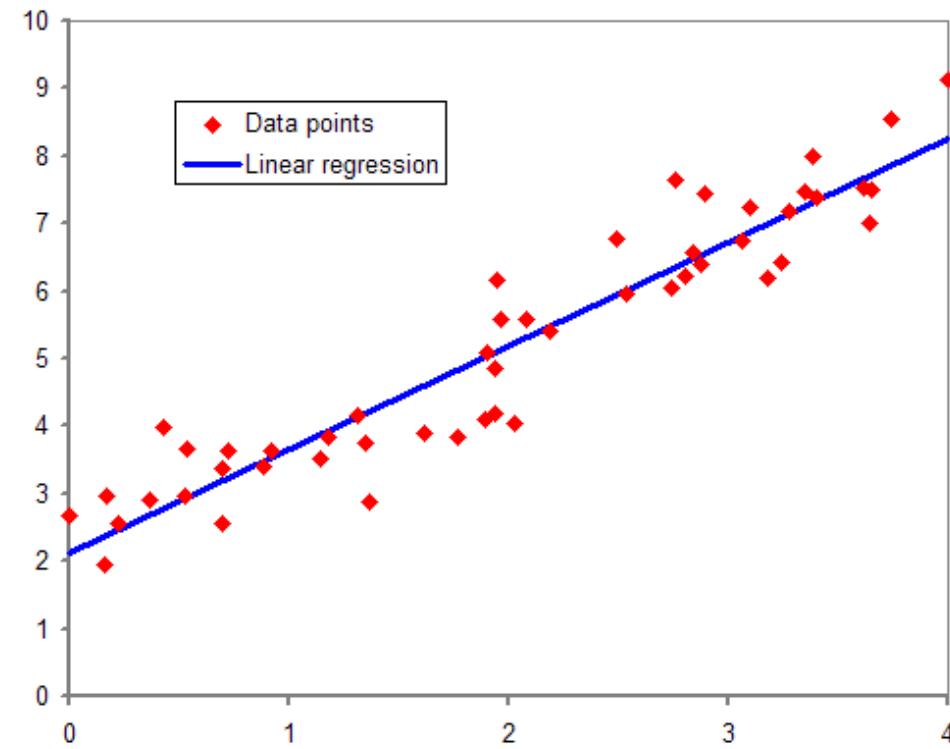
名称	日本名	説明	例
Linear Regression	線形回帰	線形モデルによる解析をしたい。あるデータが増減すると、それに伴って結果も増減する	
Bayesian Linear Regression	ベイズ線形回帰	Linear Regressionと同様だが、確率分布を用いる	
Ordinal Regression	順序回帰	順序尺度のつけたい場合	「1=良い」「2=普通」「3=悪い」
Position Regression	ポアソン回帰	特定の事情に関する回数	「1時間に通過する人数」「1日に受信するメール数」
Fast Forest Quantile Regression	高速フォレスト分位点回帰	データがどのように分布しているのか	
Decision Forest Regression	決定フォレスト回帰	分布予測するのに用いる。精度は高いが学習に時間がかかる	
Boosted Decision Tree Regression	ブースト決定木回帰	分布予測するのに用いる。精度が高く、高速だが、高メモリが必要。	
Neural Network	ニューラル	非線形モデルの決定境界を示す。精度	

# 回帰 (Regression)

未知の数値を予測する手法

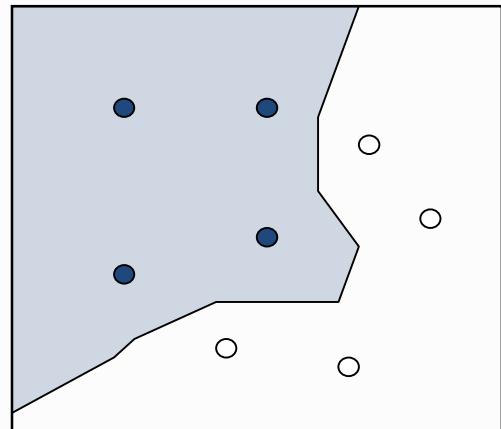
説明変数と目的変数の値の関係を表す関数を求める。

正解の数値が記載されている訓練データから、予測値を出すことを目的とする。

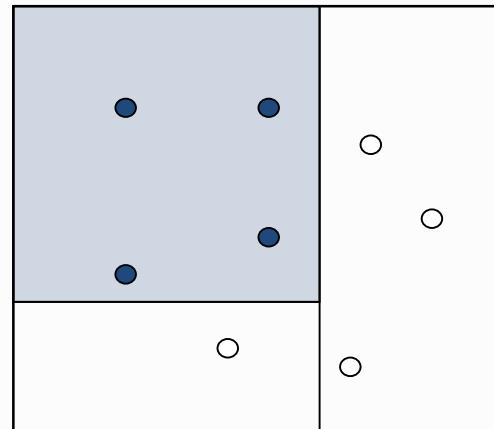


# 分類モデルの考え方

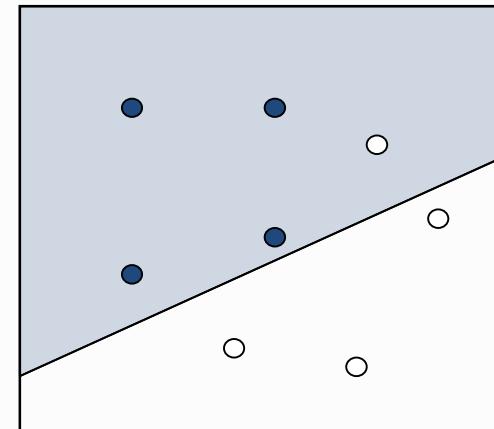
“境界”を見つけること



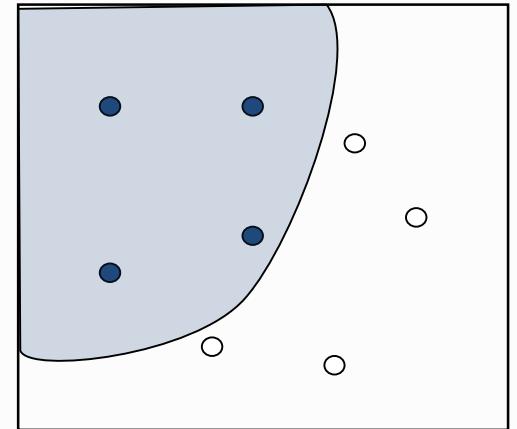
最近接値による境界



決定木



線形関数

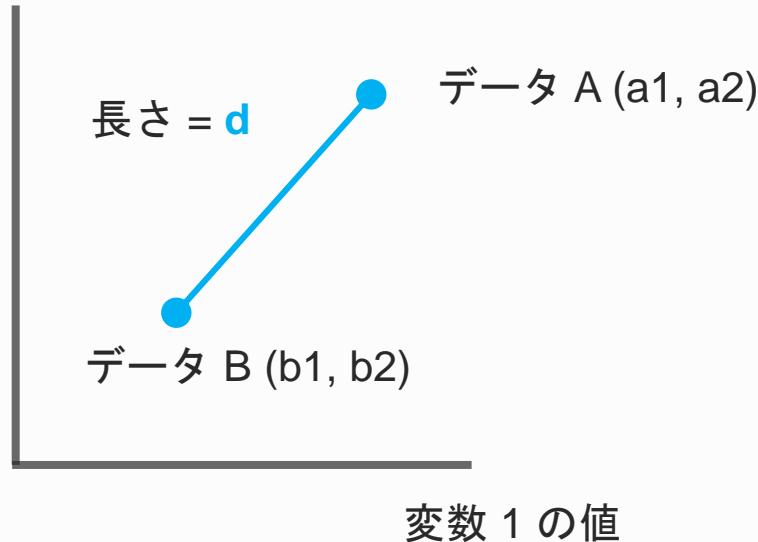


非線形関数

# 距離と類似度

- ユークリッド距離

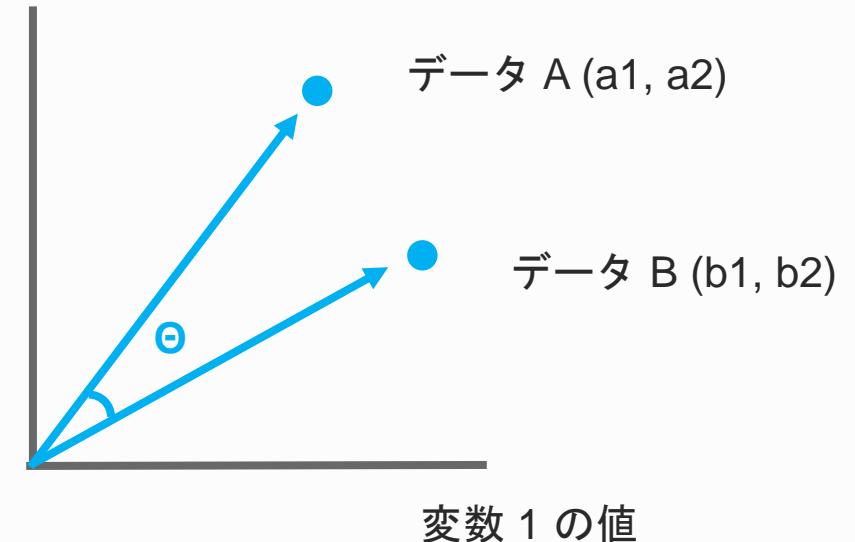
変数 2 の値



長さ  $d$  の値が短いほど類似性が高い

- コサイン類似度

変数 2 の値



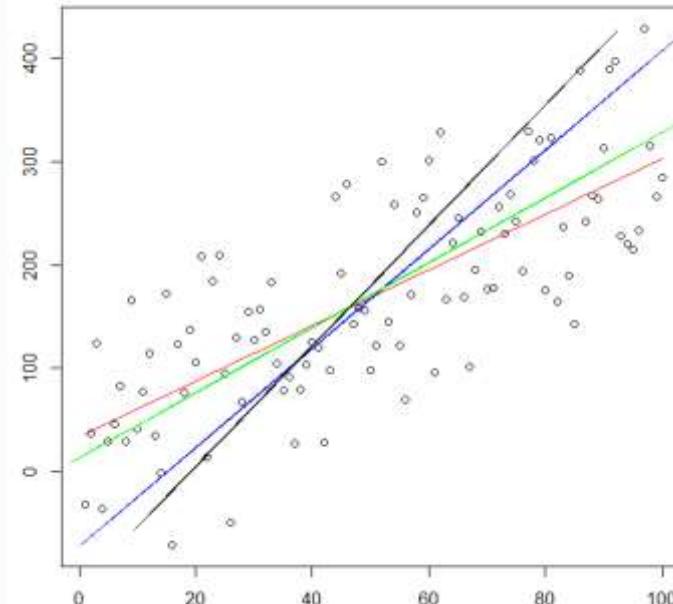
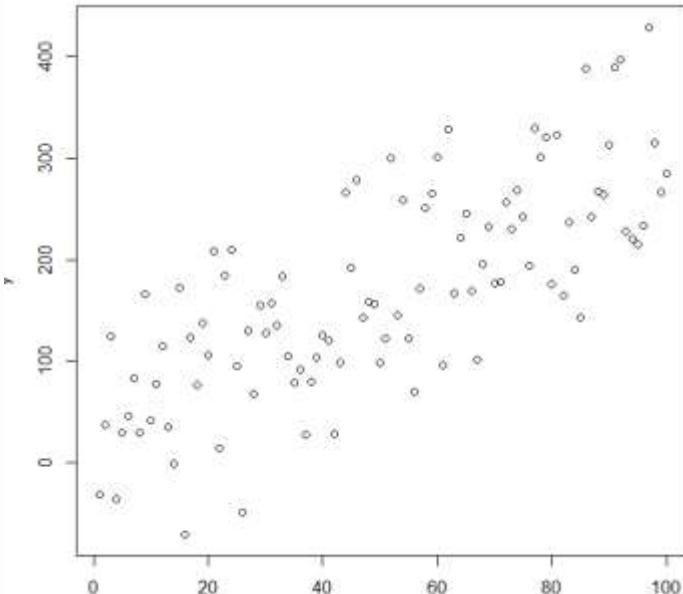
2 つのベクトルのなす角度  $\Theta$  の値が小さいほど類似度が高

# 線形回帰 (Linear Regression)

パラメータ  $w$  を調整

$$y = w_1x_1 + w_2x_2 + \dots + w_mx_m + c$$

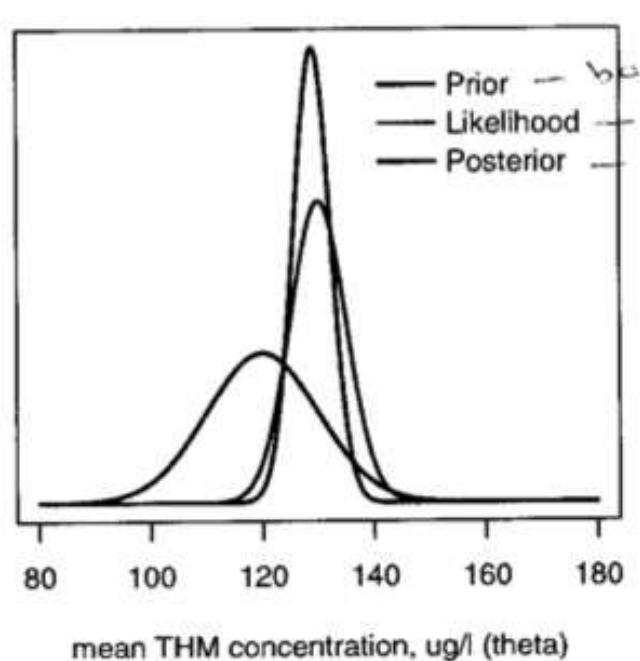
どこに線を引くか



パラメータの求め方	概要
最小二乗法 (Ordinary least squares)	学習用データにおける予測値と正解値の誤差の2乗を足した値が最小になるようにパラメータを調整
オンライン勾配降下法 (Online gradient descent)	モデルのトレーニングプロセスの各ステップでの誤差量を最小限に抑える手法

# ベイズ線形回帰 (Bayesian Linear Regression)

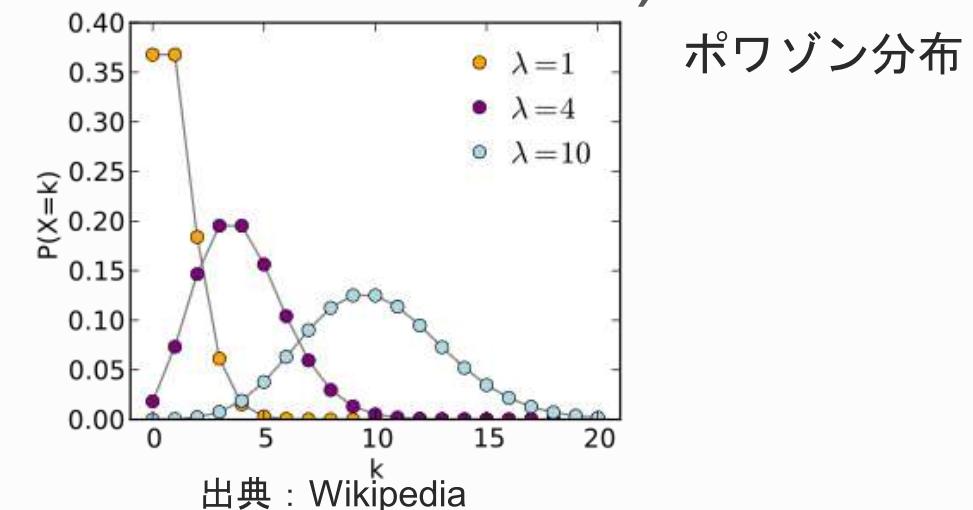
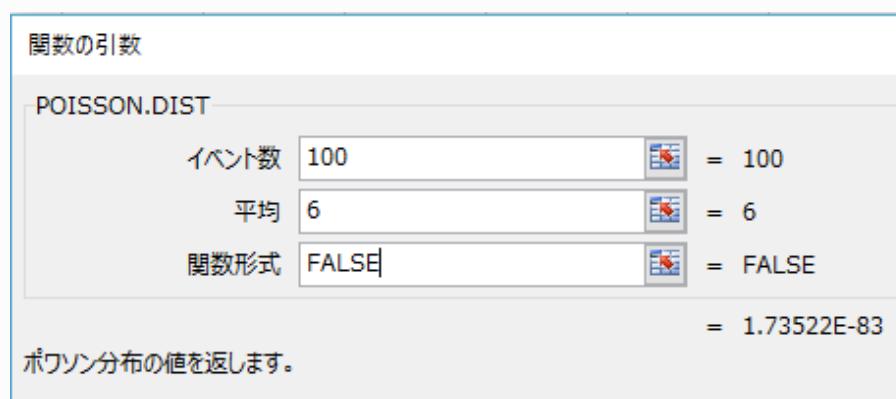
- ・ パラメータを特定の値に定めず、確率分布として扱い予測値を計算
- ・ 確率を用いて予測値の誤差の大小を予測することが可能
- ・ 学習前の確率分布(分布)を使って推定
- ・ 過学習に強い



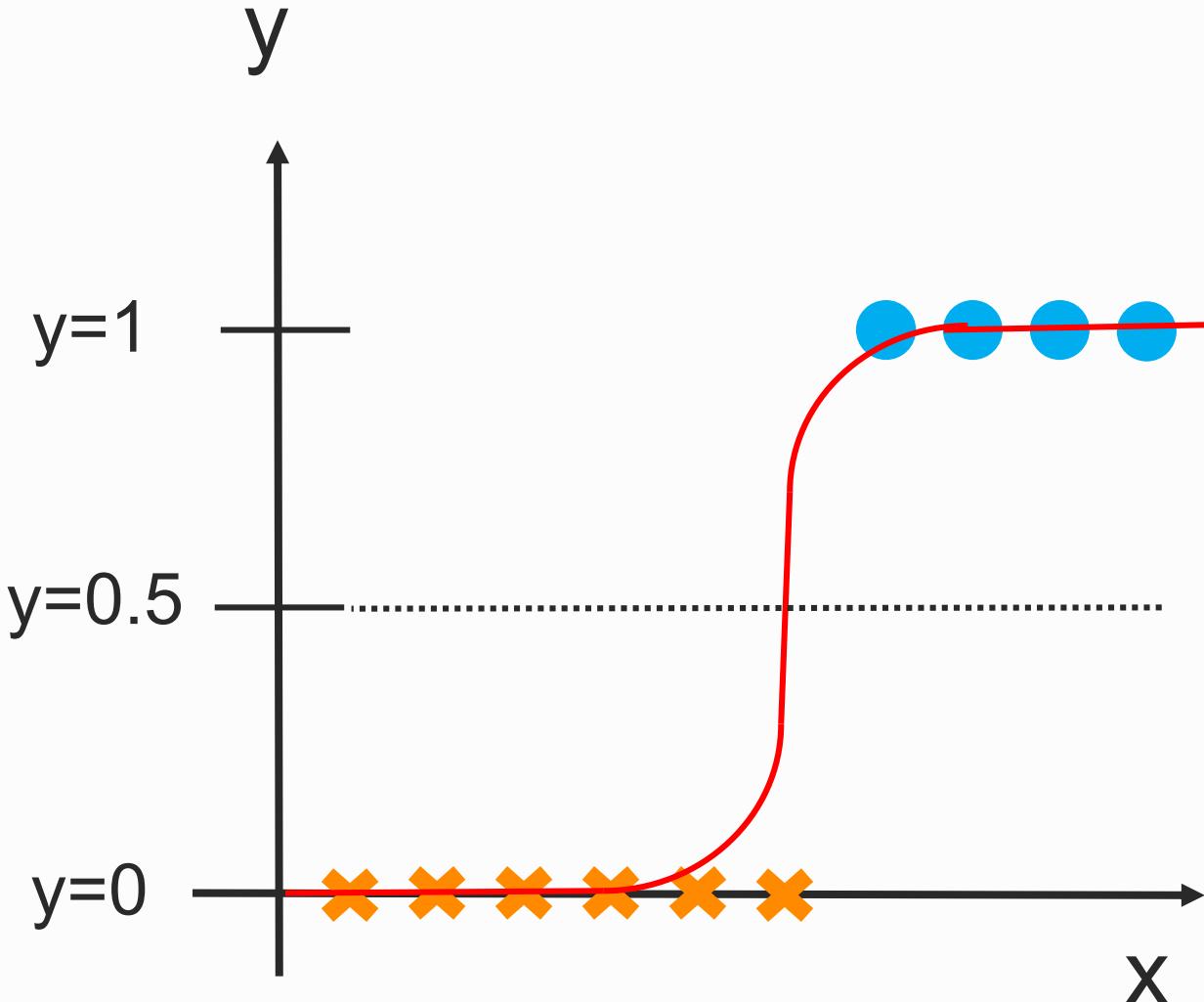
の確率分布(事後分布)

# ポワソン回帰 (Poisson Regression)

- 分布がポワソン分布に従うと想定した数値予測
- 基本的には「件数」に用いる
- 例：飛行機のフライト時に雨になる回数  
プロモーション広告提供中の問合せ件数
- Excel にも同様の関数 (POISSON/POISSON.DIST)

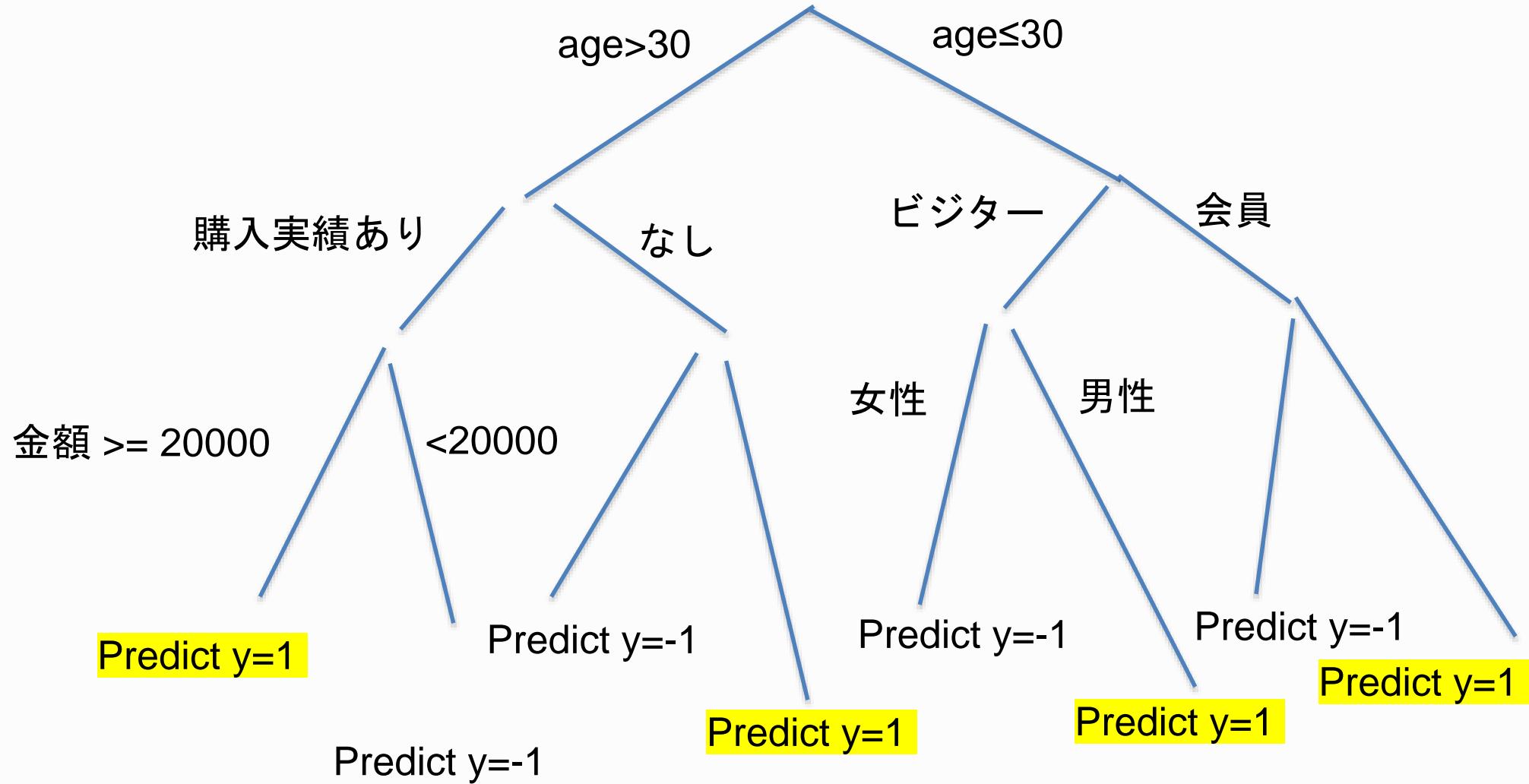


# ロジスティック回帰 (Logistic Regression)



- 結果を  $0 \sim 1$  の値で返す
- $\geq 0.5$  の際は 1,  $<0.5$  の際は 0

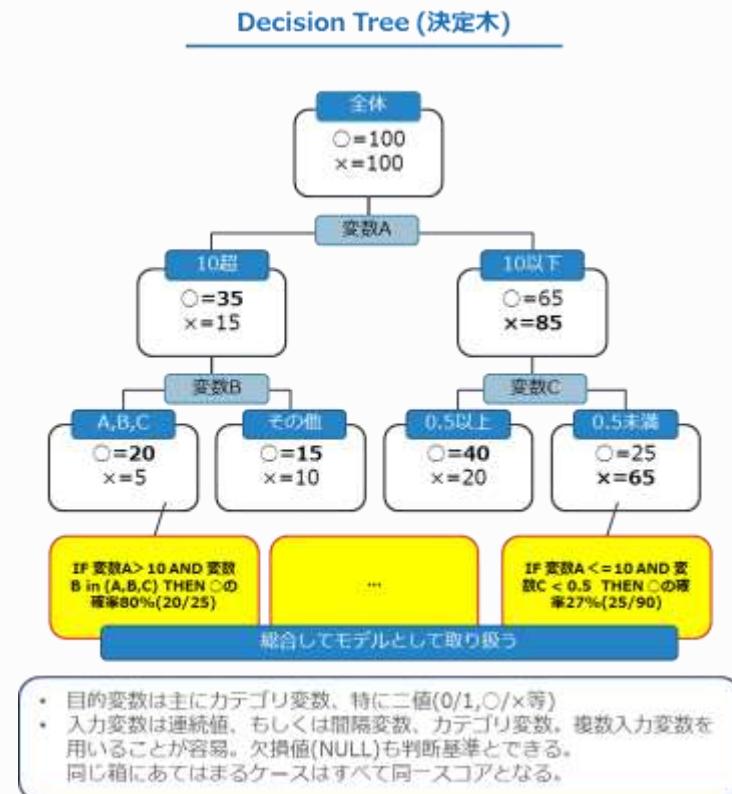
# 決定木 (Decision Tree)



# Decision Forest

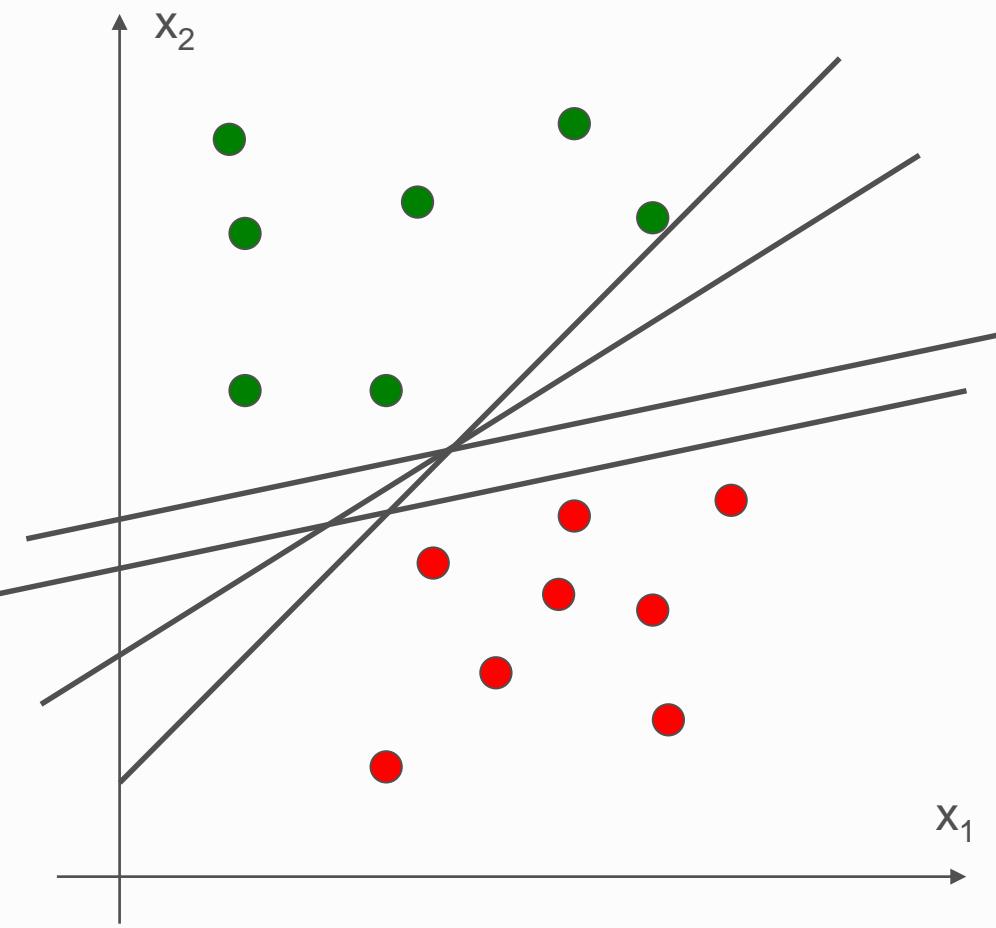
Decision Forest（ランダムフォレスト）とは？

Decision Treeが進化したもので、マシンパワーの飛躍的な向上で実現できるようになった最新モデル化技術。 Decision Treeモデルはわかりやすく、汎用的である一方で、モデルによっては偏りが高く、人間によるモデル修正が必要とされていた。 Decision Forestは多サンプルによる合議制(Ensemble方式)を取り、モデル構築データ依存度の低い（低バリアンス）なメソッドである。

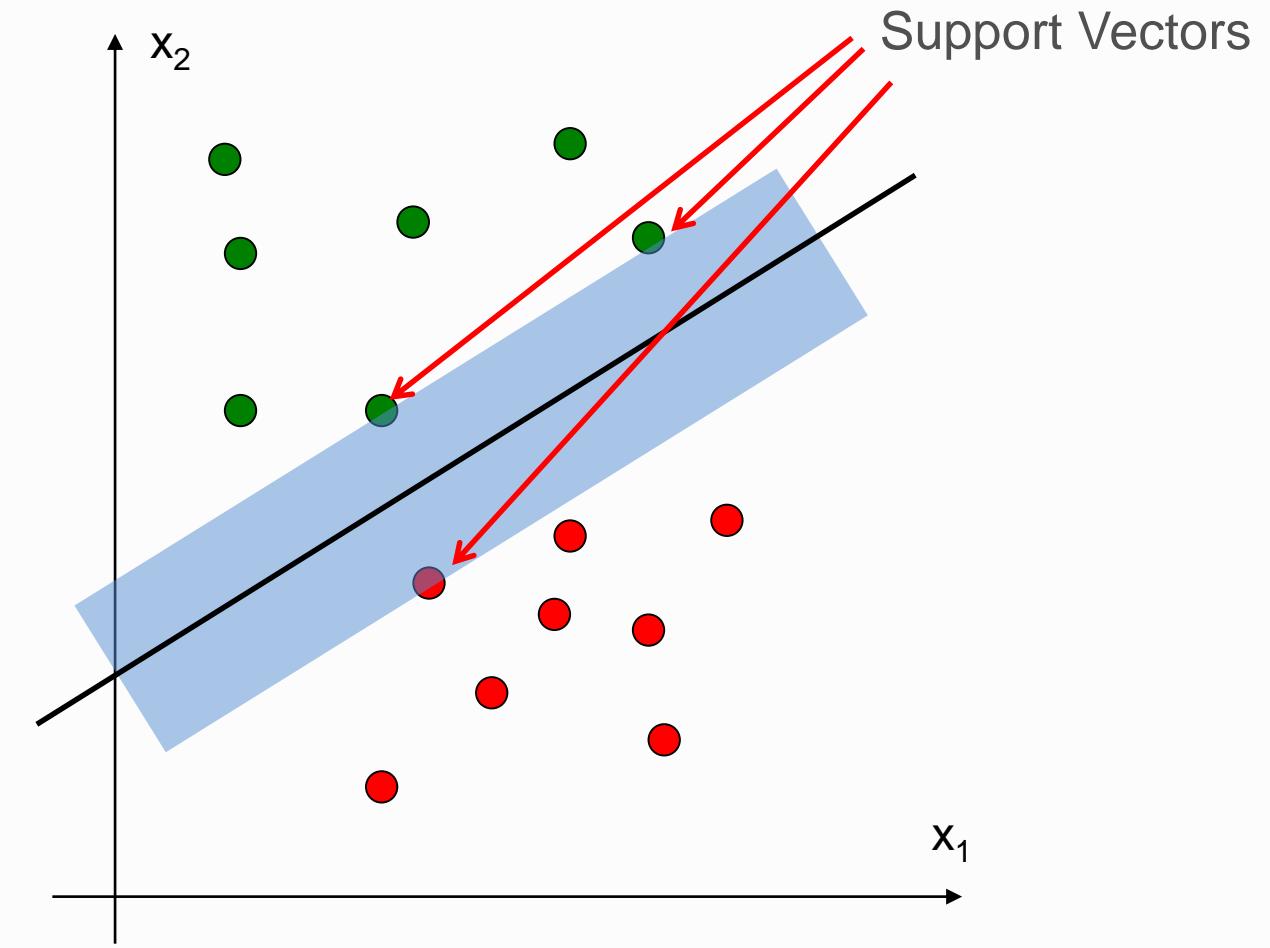


# サポートベクターマシン (SVM)

2 クラス分類



マージンの最大化



# 分類

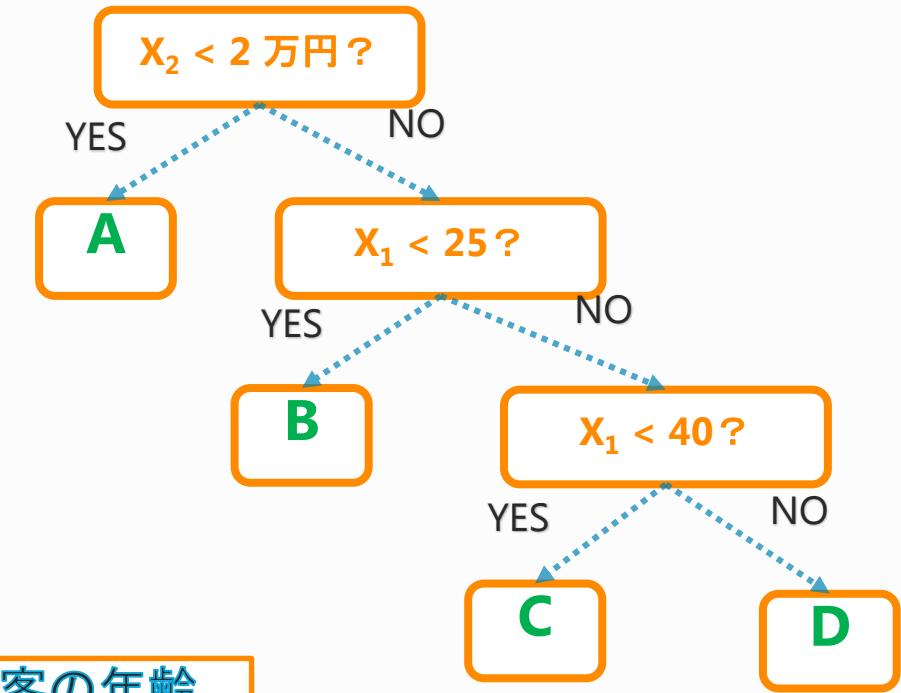
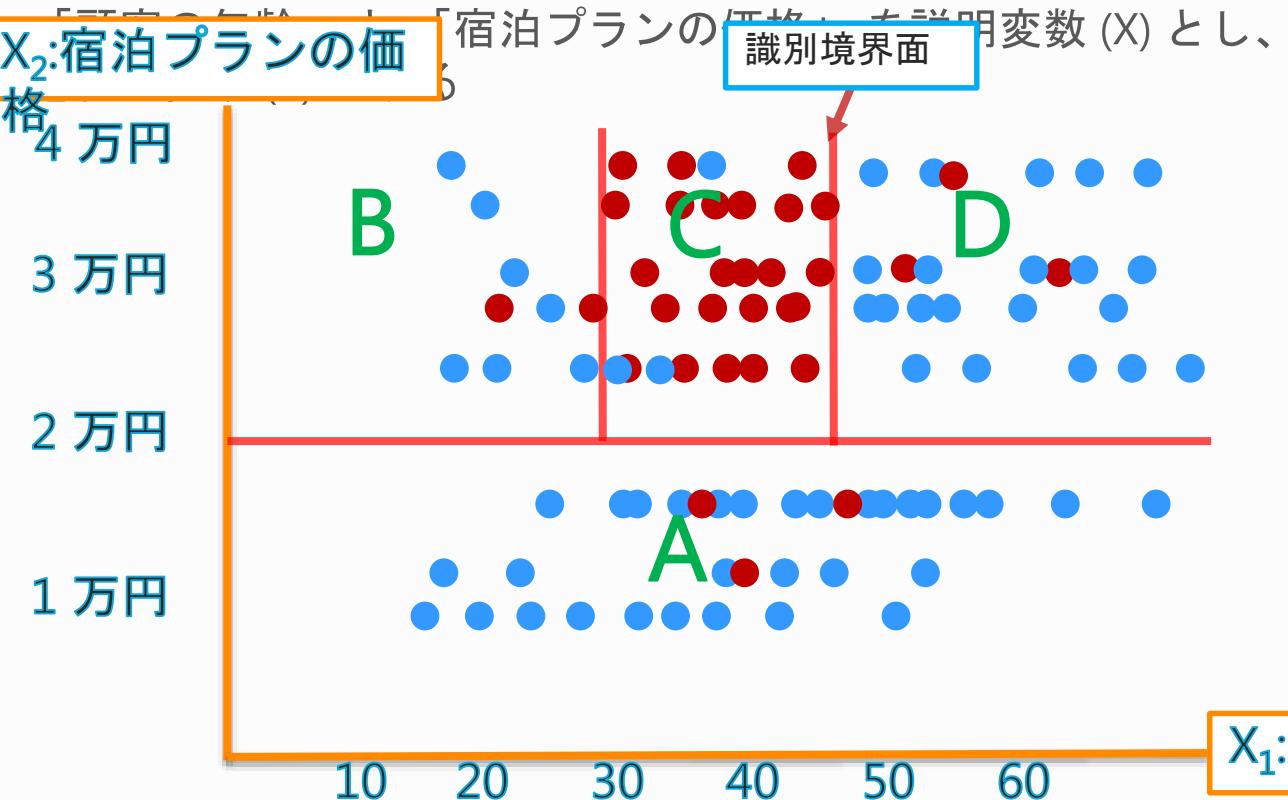
分類アルゴリズムは分類器と呼ばれ、その出力は新しいインスタンスのラベルを予測するため使用される

- 分析の種類
 

分析の種類	説明	学習方法
クラス分類	母集団に属する要素が、ある基準で分類された集合のどこに属するかを予測する • どのメールがスパムか？(メールの属性データを使用し、「スパムである」「スパムでない」に分類)	「教師あり」学習
クラスタリング	特定の分類基準を与えず、属性データから類似性から、母集団をグルーピングする • どの顧客グループにどのような製品を提案するべきか？	「教師なし」学習
- 「教師あり」学習
  - データからの予測
  - 正解付きデータ(入力と出力の組からなるデータ)をもとにトレーニングを行い、出力が未知である入力値に対し、出力値を予測
- 「教師なし」学習
  - データの分類
  - トレーニングデータ(外部から与えられる条件)を用いずにデータ構造を推定する

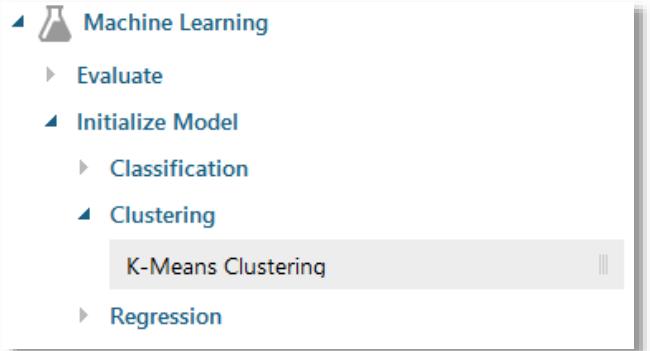
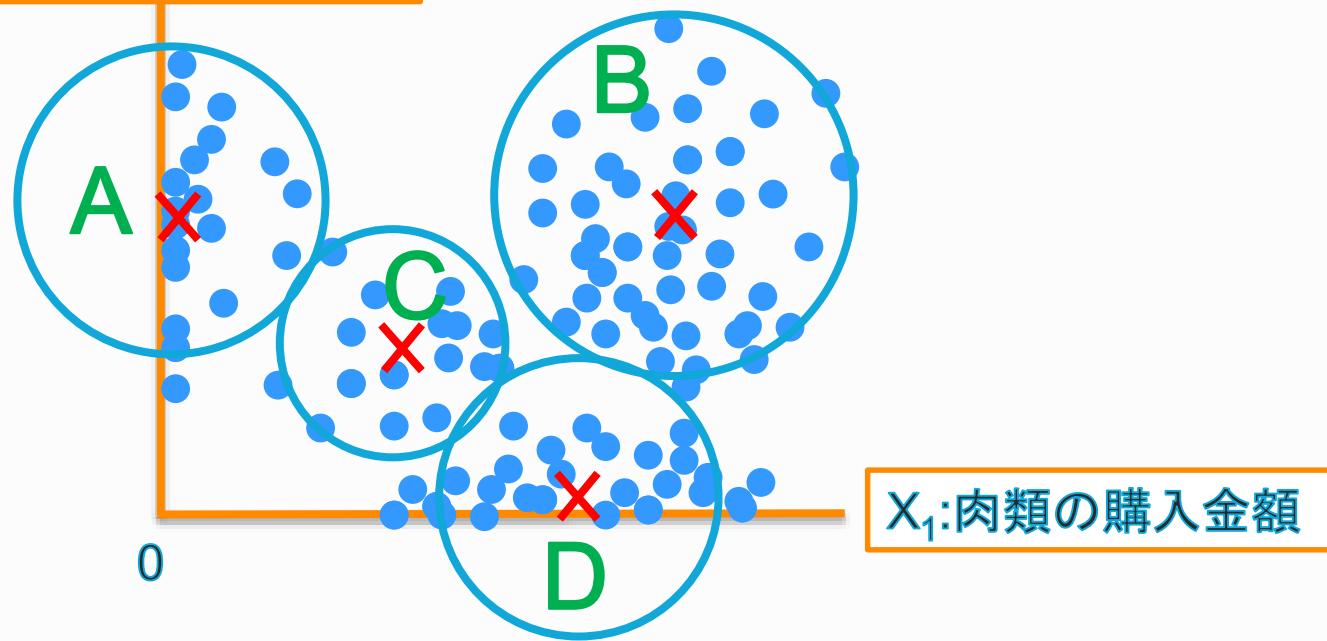
# クラス分類

- 特徴(説明変数)とクラス(目的変数)を持つ既存データを使用して対象データがどこに分類されるかを特定する
  - デシジョンツリーを使用したクラス分類の例
  - $X_2:$ 宿泊プランの価格



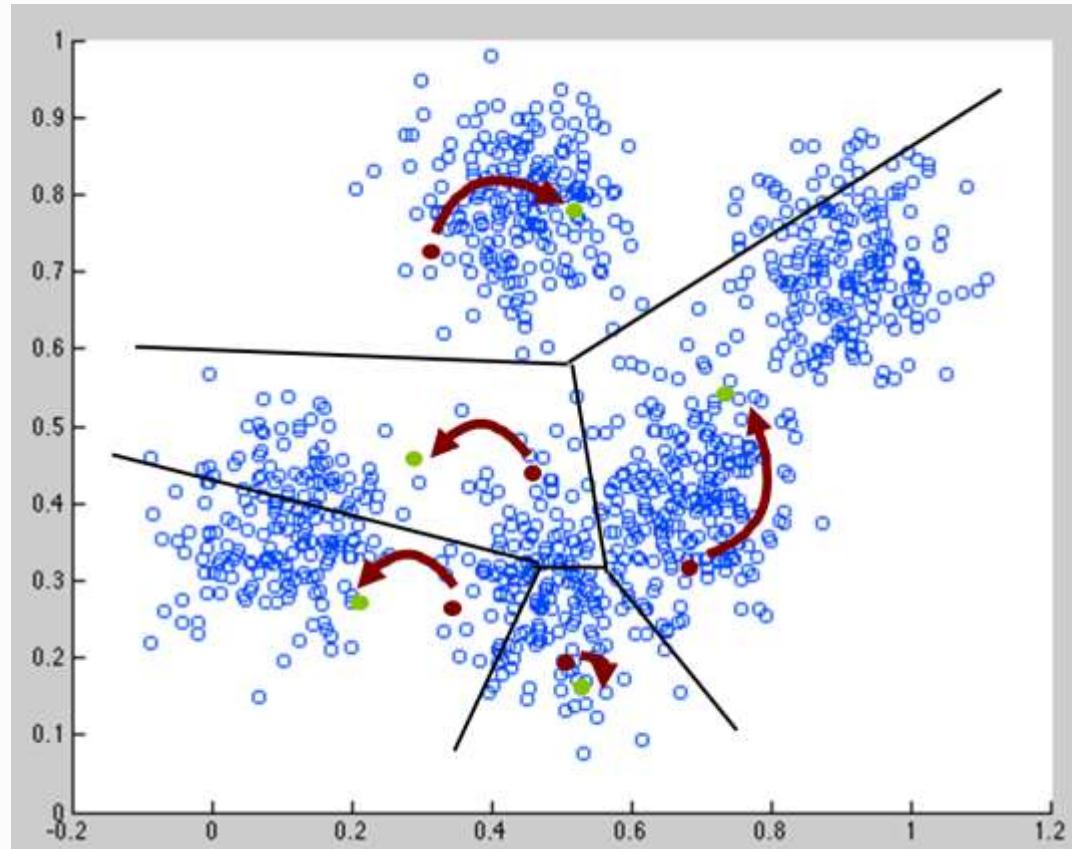
# クラスタリング

- データをクラスタと呼ぶグループに分ける
  - クラスタとは同じクラスタのデータならば互いに似ていて、違うクラスタならば似ていないようなデータの集まり
- K 平均法 (k-means) アルゴリズム
  - 中心点と各データの間の距離が最小になるように中心点とデータの分割を決める
- 購入した食品の種類により 4 つのクラスターにグループ化した例



- 適用分野/業務
  - セグメンテーション
  - 顧客グルーピング
  - メール キャンペーン
  - 異常値検出

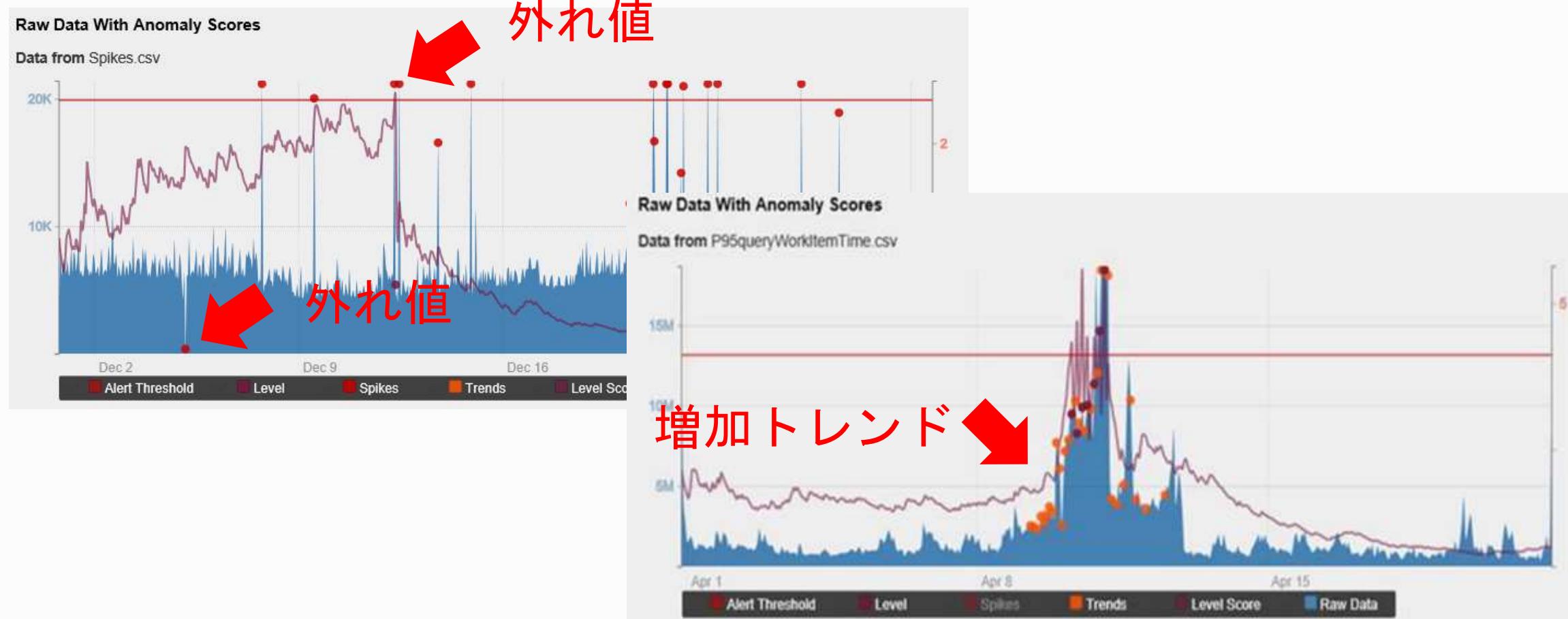
# K 平均法 (K-means)



1. クラスタ数の分だけ、ランダムに点をプロットする
2. 各点にランダムにクラスタを割り当て、クラスタの中心を計算する
3. できたクラスタの中心になるようにポイントを更新する
4. 2. - 3. を収束するまで繰り返す

# 異常検知

- ・入力データから外れ値、外れトレンドを検出



# 異常検知の主要なアルゴリズム

- One-Class SVM
- One-Class (1 種類) のデータを利用  
→ 異常値 (教師データ) を用意しづらいシナリオで主に利用される
- libsvm ライブラリを利用 (Azure ML でも利用)

# リコメンデーション

## ■ Matchbox Recommender

- Microsoft Research が開発したラージ・スケール・ベイジアン・リコメンダー・システム
- 例えば映画やコンテンツ、その他の商品などのアイテムの評価を利用し、学習することができる
- リクエストごとにユーザーに対して新しいアイテムをリコメンドすることができる
- Matchbox のアルゴリズムは協調フィルタリングおよびコンテンツベースのリコメンド方式がベース
  - ※ Matchbox の詳細については以下のサイトからご確認頂けます

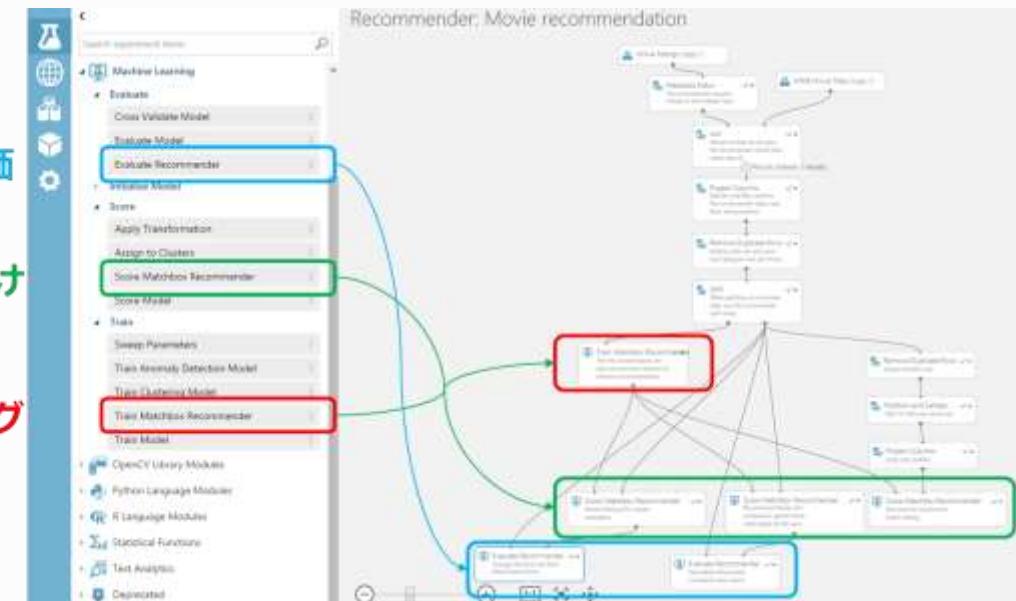
Matchbox Recommender System  
<http://research.microsoft.com/en-us/projects/matchbox/>

Matchbox: Large Scale Online Bayesian Recommendations  
 (アルゴリズムの詳細)  
<http://msr-waypoint.com/pubs/79460/www09.pdf>

### – 考慮点

- 類似度の計算を選択する事ができず、類似度の値を確認するには工夫が必要
- 評価の際に、混同行列から各種評価値を作る場合には工夫が必要

評価  
スコア付け  
トレーニング



# レコメンデーションの用途

- ・ バスケット分析 (併売分析)
- ・ 商品ベースのマッチング
  - ・ 商品 A と商品 B は似ている
  - ・ → コンテンツベース フィルタリング
- ・ ユーザーベースのマッチング
  - ・ ユーザー A とユーザー B は似ている)
  - ・ → 協調フィルタリング

# レコメンデーション

アリゾナ・トノベ  
アリゾナ・ラーナベ  
アリゾナ・データーベ  
64 - 後編  
植物図鑑

A さん	5	4	7	4
B さん	5	6	7	?
C さん	1	7	-	7
D さん	2	8	6	-

# レコメンデーション

アリス・イン・ランド  
フジイ・ノーテイク  
64 - 後編 植物図鑑

アリス・イン・ランド  
フジイ・ノーテイク  
64 - 後編 植物図鑑

A さん  
B さん  
C さん  
D さん

5	4	7	4
5	4	7	?
3	8	-	8
2	5	6	-

5	4	7	4
5	4	7	?
3	8	-	8
2	5	6	-

# モデルのトレーニング

- トレーニングの種類

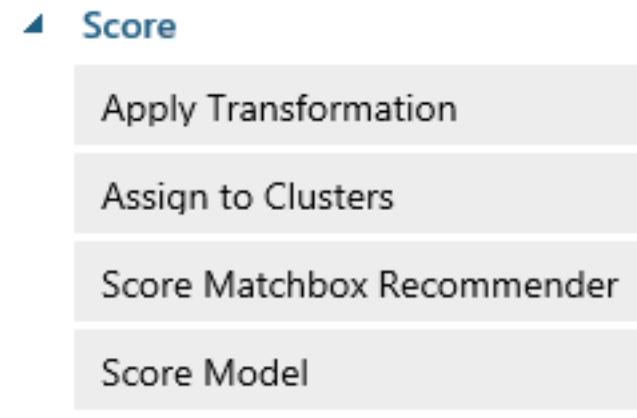
トレーニングの種類	説明
Sweep parameters	最適なパラメーター設定を決定するために、モデルのパラメーターの絞り込みを行う
Train Anomaly Detection Model	外れ値検知モデルのトレーニング
Train Clustering Model	クラスター モデルのトレーニングとクラスターへのデータ割り当て
Train Matchbox Recommender	マッチ ボックス レコメンダー (Matchbox Recommender) モデルのトレーニング
Train Model	回帰 (Regression) と分類 (Classification) モデルのトレーニング

Train
Sweep Parameters
Train Anomaly Detection Model
Train Clustering Model
Train Matchbox Recommender
Train Model

# モデルのスコア付け

- モデルのスコア付け
  - モデル評価のためのテスト データを使用したスコアリング

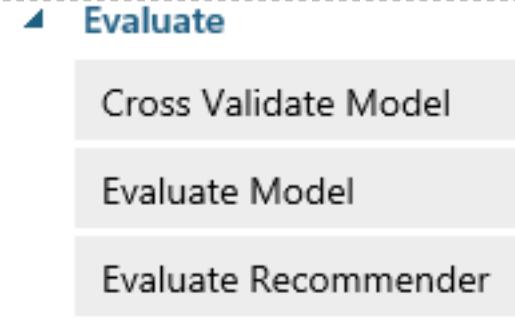
スコア付けの種類	説明
Apply Transformation	データセットへの事前定義されたデータ変換の適用
Assign to Clusters	トレーニング済みのクラスター モデルを使用したデータの割り当て
Score Matchbox Recommender	マッチ ボックス レコメンダー (Matchbox Recommender) モデルのスコア付け
Score Model	回帰 (Regression) と分類 (Classification) モデルのスコア付け



# モデルの評価

- ・ モデルの評価
  - ・ 複数スコア付けされたモデルの比較

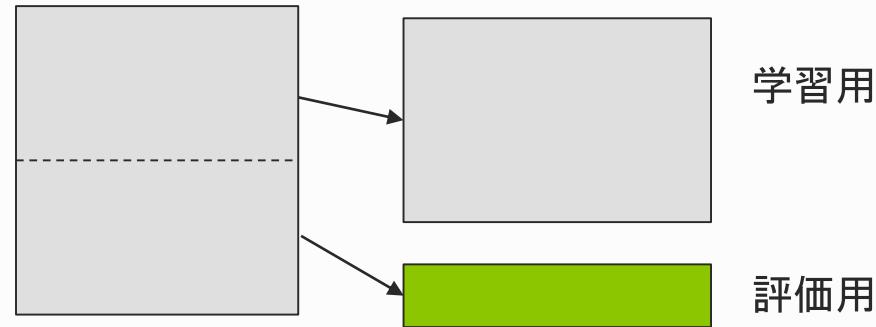
評価の種類	説明
Cross Validate Model	回帰 (Regression) モデルと分類 (Classification) モデルのクロス評価
Evaluate Model	スコア付けされた回帰 (Regression) または、分類 (Classification) モデルの評価
Evaluate Matchbox Recommender	スコア付けされたマッチボックス レコメンダー (Matchbox Recommender) モデルの評価



# モデルの評価方法

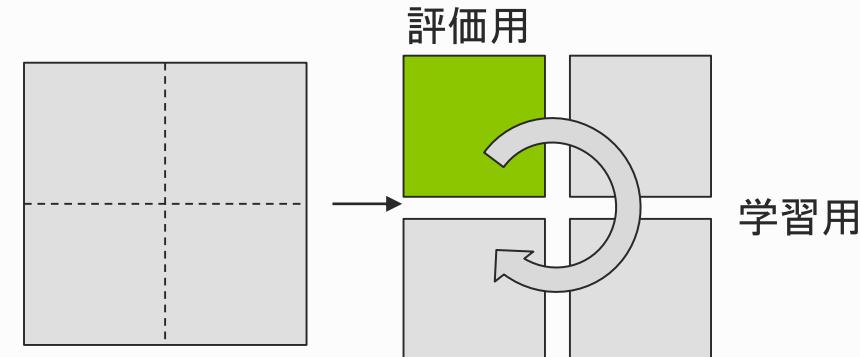
- ホールドアウト法

- データを訓練用と評価用データに分割
- 簡単、早い
- 評価のために訓練用データを減らす必要がある



- クロスバリデーション

- データを複数個に分割し、1つのデータセットを評価用、残りのデータセットを訓練用として利用、その後に別のデータセットを評価用に変更し同様に評価することを反復
- データを効果的に利用できる
- 時間がかかる



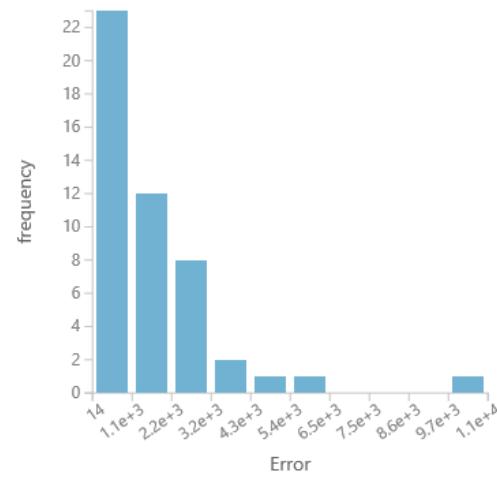
# 回帰モデルの評価

自動車価格の予測 - 比較 › Evaluate Model › Evaluation results

## Metrics

Mean Absolute Error	1656.147651
Root Mean Squared Error	2456.983209
Relative Absolute Error	0.276606
Relative Squared Error	0.089608
Coefficient of Determination	0.910392

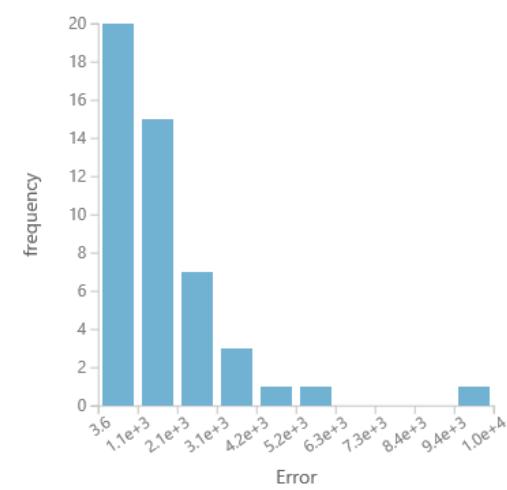
## Error Histogram



## Metrics

Mean Absolute Error	1701.447535
Root Mean Squared Error	2494.640026
Relative Absolute Error	0.284172
Relative Squared Error	0.092376
Coefficient of Determination	0.907624

## Error Histogram



## 評価方法

平均絶対誤差  
(Mean Absolute Error: MAE)

## 概要

絶対誤差の平均値

平均二乗誤差  
(Root Mean Squared Error: RMSE)

誤差の二乗の平方根値

相対絶対誤差  
(Relative Absolute Error: RAE)

絶対誤差の合計を正規化して  
合計相対誤差で除算した値

相対二乗誤差  
(Relative Squared Error: RSE)

二乗誤差の合計を正規化して  
合計二乗誤差で除算した値

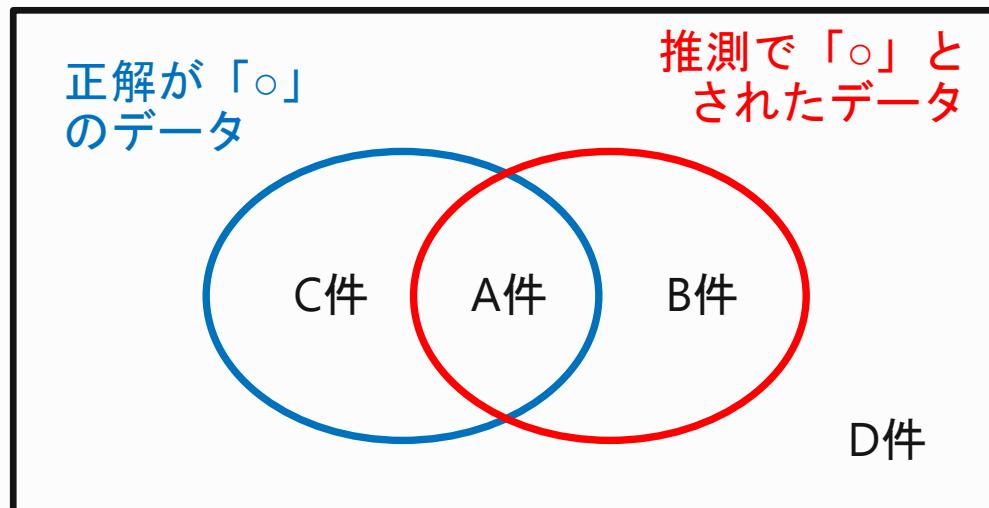
決定係数 ( $R^2$ )  
(Coefficient of Determination)

特徴量がラベルを決定す  
る度合 (悪:0 ~ 良:1)

# 分類モデルの評価 (2クラス分類モデル)

## 予測結果例

検証用データ		予測で得たクラス	
		○	×
正解の クラス	○	A件	C件
	×	B件	D件

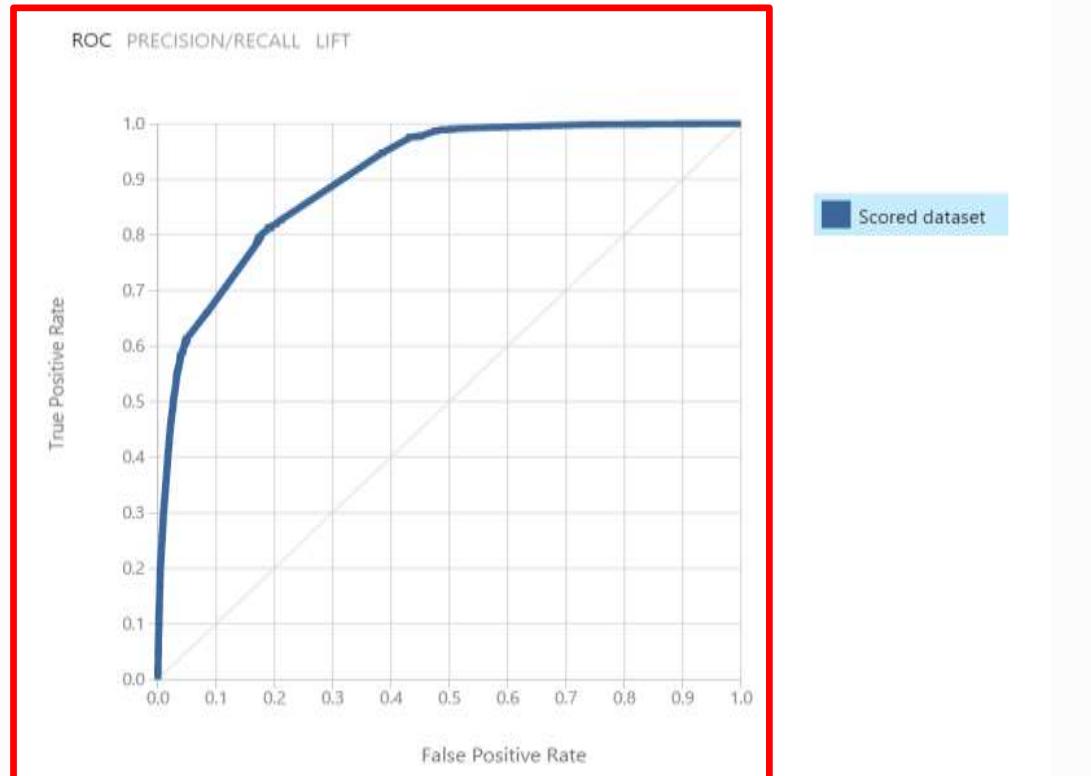


## 主な評価指標

- ① True Positive (真陽性) : 100%に近いほど良好  
 $\Rightarrow A/(A+C)$
- ② False Positive (偽陽性) : 0%に近いほど良好  
 $\Rightarrow B/(B+D)$
- ③ True Negative : 100%に近いほど良好  
 $\Rightarrow D/(B+D)$
- ④ False Negative : 0%に近いほど良好  
 $\Rightarrow C/(C+D)$
- ⑤ Accuracy (正解率) : 100%に近いほど良好  
 $\Rightarrow 「○」「×」を正しく予測できた割合$   
 $\Rightarrow (A+D)/(A+B+C+D)$  : 100%に近いほど良好
- ⑥ Precision (適合率) : 100%に近いほど良好  
 $\Rightarrow A/(A+B)$
- ⑦ Recall (再現率) : 100%に近いほど良好  
 $\Rightarrow ①$ に同じ
- ⑧ F1 Score : 1.0に近いほど良好  
 $\Rightarrow ⑥、⑦$ の複合指標  
 $\Rightarrow 2 \times (⑥ \times ⑦) / (⑥ + ⑦)$

# 分類モデルの評価 (2クラス分類モデル)

## その他の評価指標 (ROC曲線とAUC)



- ROC曲線  
⇒ 真偽の判断の閾値を変化させて  
    真陽性、偽陽性をの推移をグラフかしたもの  
⇒ 縦軸が真陽性率、横軸が偽陽性率  
⇒ 偽陽性をあまり悪化させることなく、  
    真陽性を大きく向上できる  
(=左上に張り付くような曲線) ほど  
    優れたモデルであることを示す
- AUC  
⇒ ROC曲線の下の面積  
⇒ 1に近いほど優れたモデルであること示す

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
7481	1897	0.811	0.818	0.5	0.904
False Positive	True Negative	Recall	F1 Score		
1669	7835	0.798	0.808		
Positive Label	Negative Label				
1	0				