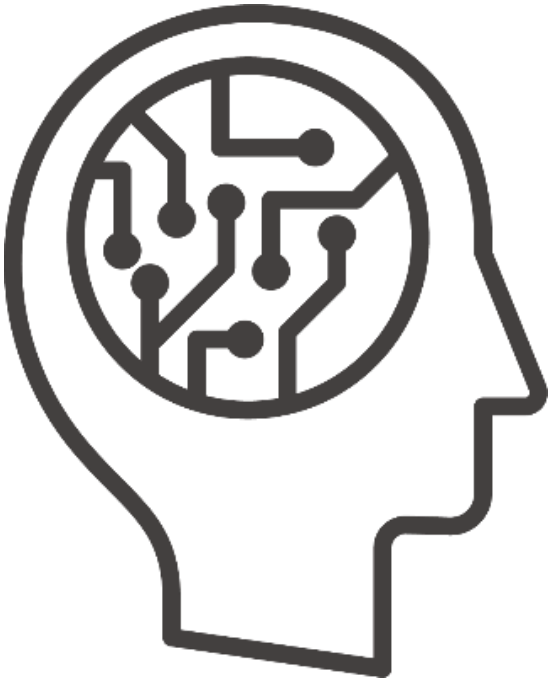
やまぐち
高校生データサイエンティスト育成講座
～day2～

本日のテーマ

AI



本日のスケジュール

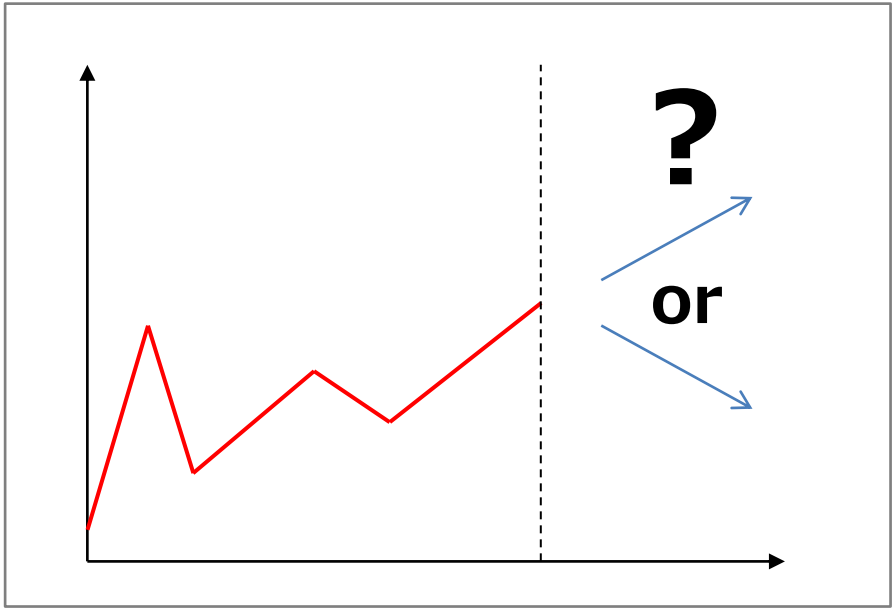
時間	内容
10：10 ～ 11：00	予測するってなに？
11：00 ～ 12：00	簡単なモデルを作ってみよう
12：00 ～ 13：00	昼休憩
13：00 ～ 14：00	重回帰モデルを作ってみよう
14：00 ～ 14：10	休憩
14：10 ～ 15：50	特徴量を作ってみよう①②
15：50 ～ 16：00	閉講・アンケート

予測するってなに？

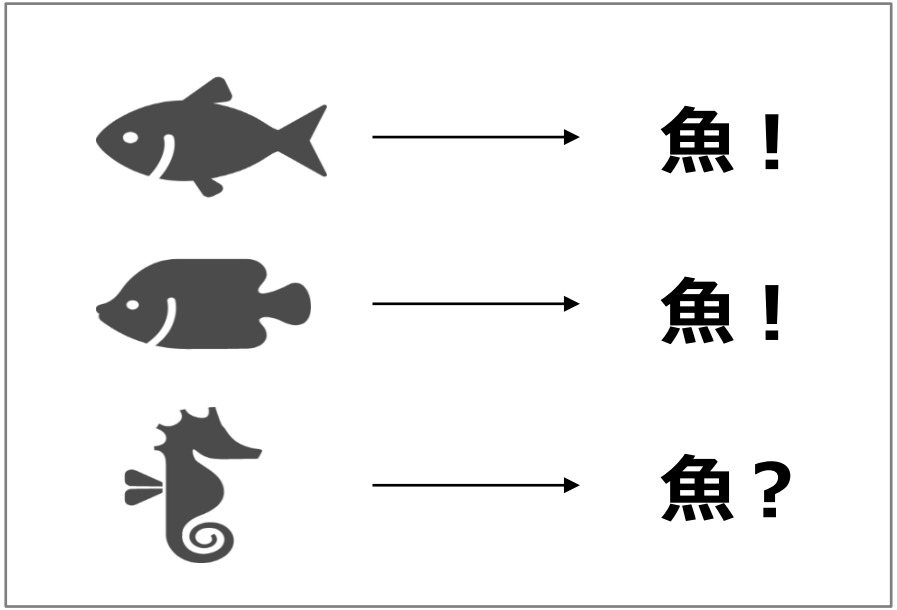
予測とは？

“**データに基づき**、ある値がどのような値となるかを想定すること”

予測するのが数値だったら・・・



予測するのがラベルだったら・・・



予測をする為には…

E.g.) お弁当の売上を予測したい

【必要となるデータ】

予測したいもの：
お弁当の売り上げ実績数

予測のヒントになりそうなもの：
天気情報、来店客数、気温

予測をする為には…

E.g.) お弁当の売上を予測したい

【必要となるデータ】

予測したいもの：
お弁当の売り上げ実績数

予測のヒントになりそうなもの：
天気情報、来店客数、気温

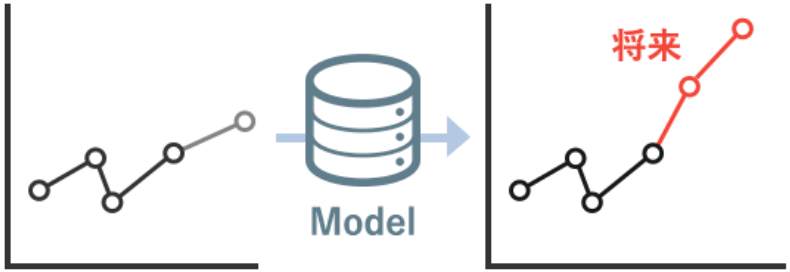
目的変数

説明変数

代表的な2種類の予測問題

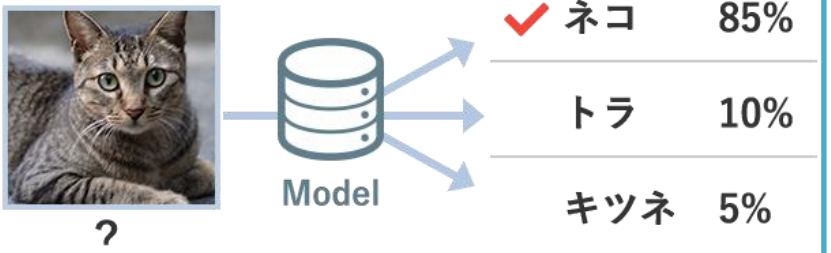
回帰問題

目的変数が**数値**



分類問題

目的変数が**カテゴリ**



それって本当に予測できてる？

①ある日・・・

このデータを使って
良い精度のモデルを作ってくれ

Bさん

Aさん

それって本当に予測できてる？

①ある日・・・

このデータを使って
良い精度のモデルを作ってくれ

わかりました！

Bさん

Aさん

それって本当に予測できてる？

②モデリング前

とりあえずモデリングしてみよう。
もらったデータを全部使って、
このデータで良い精度のモデルを作れ
ばいっか！

Aさん

それって本当に予測できてる？

③モデルの評価

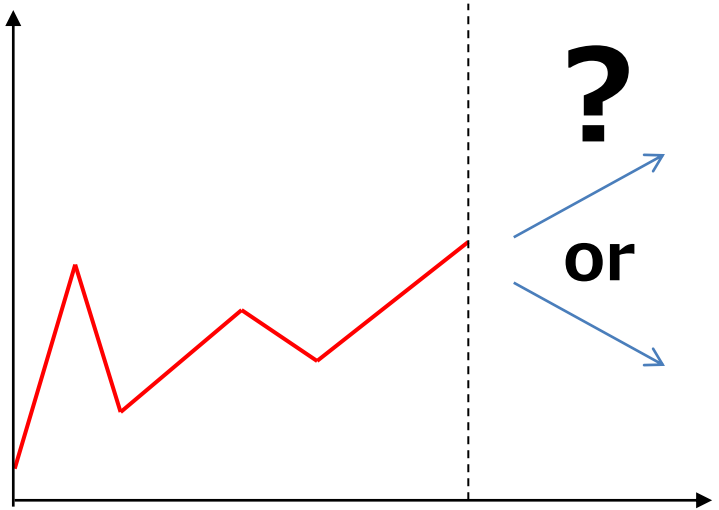
予測精度99%のモデルが出来た！
俺って天才なのかもしれない！

Aさん

それって本当に予測できてる？

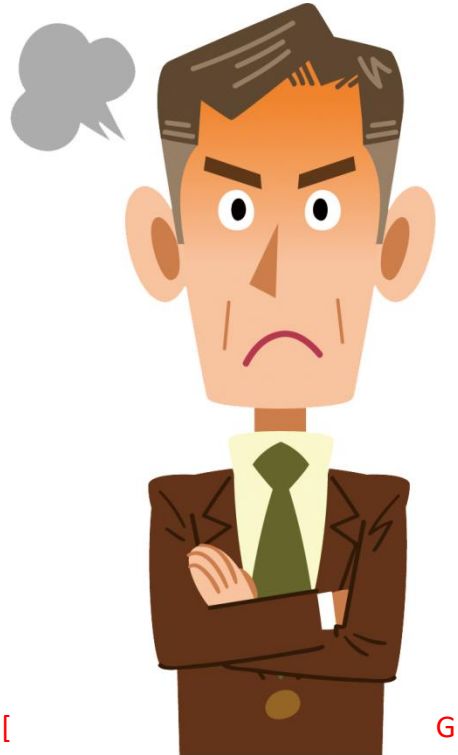
④実際に運用してみたら…

Aさん



それって本当に予測できてる？

④実際に運用してみたら…



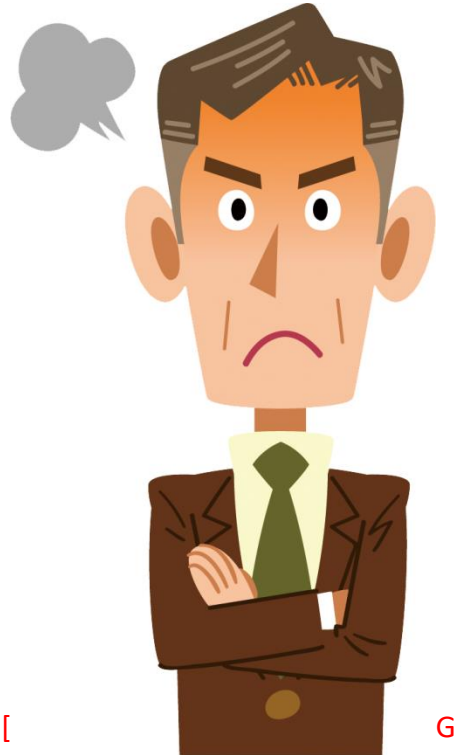
全然予測精度が出ないじゃないか！
どうなってるんだ！！！

Bさん

Aさん

それって本当に予測できてる？

④実際に運用してみたら…



全然予測精度が出ないじゃないか！
どうなってるんだ！！！

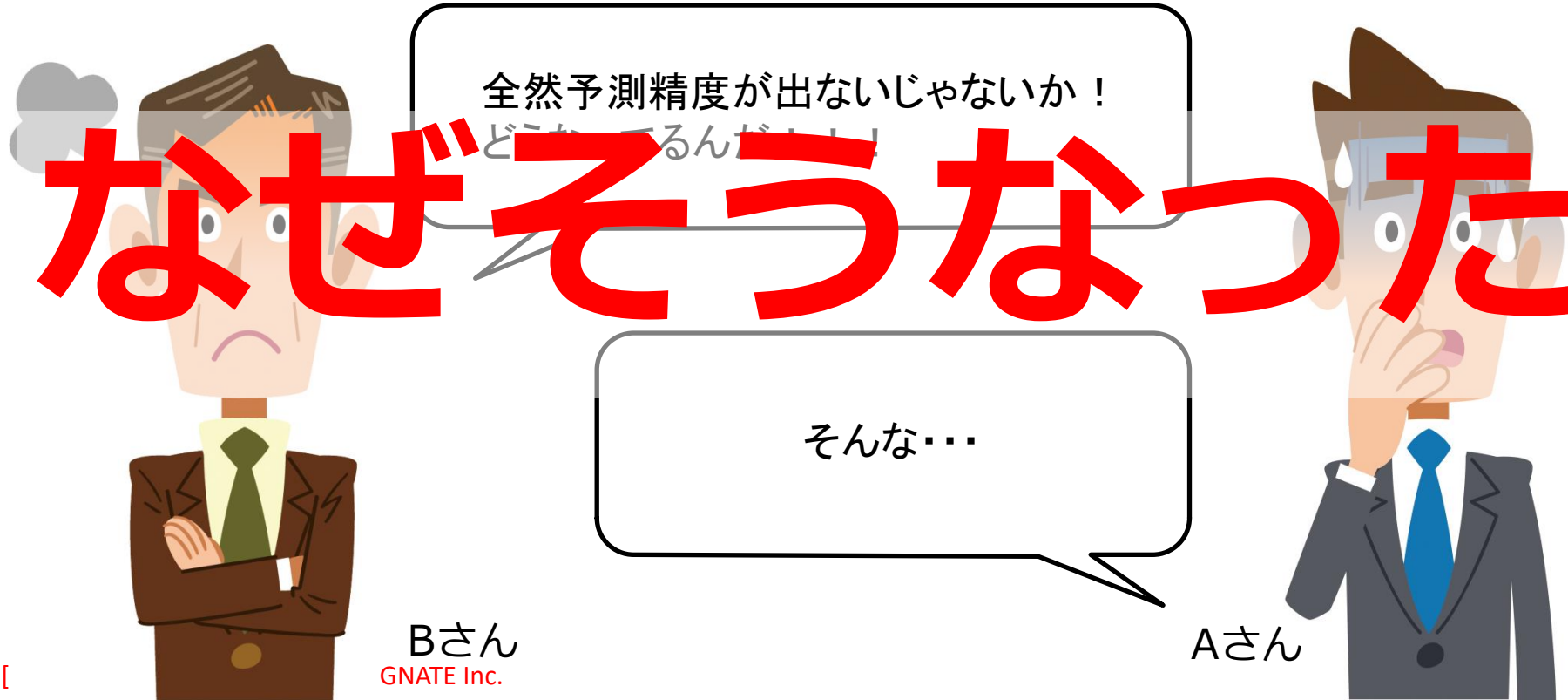
そんな・・・

Bさん

Aさん

それって本当に予測できてる？

④実際に運用してみたら…



何がいけなかったのか？

- いきなりモデルを作り始めた
  - 基礎分析を怠ることなかれ

- モデルを作るときに全てのデータを使ってしまった
  - その結果、モデルが過学習と呼ばれる状態になってしまった

過学習（Overfitting）とは？

- モデル作成に使ったデータだけに特化し過ぎたモデルを作ってしまうこと

例．文系or理系出身かを判定するモデルを作る

【部署Aでは】

部署Aでは、
・理系の男性は全員メガネをしている
・文系の女性は全員メガネをしていない

だから、男性でメガネをしていれば理系、女性でメガネをしていなければ文系と判別すれば、精度100%のモデルを作れるぞ！

ところが…

【部署Bでは】

部署Bでは、
・理系の男性でメガネをしていない男性が多くいた
・文系の女性でメガネをしている女性も多くいた

全体の傾向を考えずに部署Aだけのデータだけに引っ張られた結果、他部署では予測できていないモデルになってしまった…

作ったモデルの精度を検証するには？

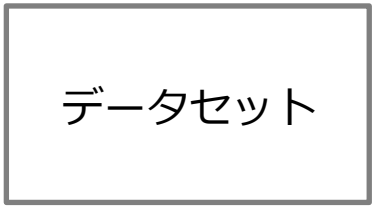
- **予測モデルのゴール**
  - **“未知のデータ”**も予測できるような汎用性あるモデルを作ること

- **どうすればいい？**
  - データを分割して擬似的に未知のデータを作る
    - 片方でモデルを作り、残りを未知のデータする
    - この未知のデータを上手く予測できることを目標とする

未知のデータを予測できる？

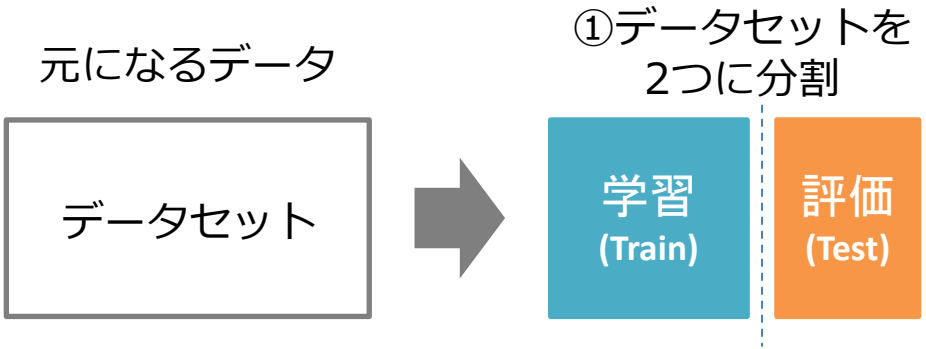
“**未知のデータ**でも予測できるか？（汎用性）”
を評価することが大切

元になるデータ



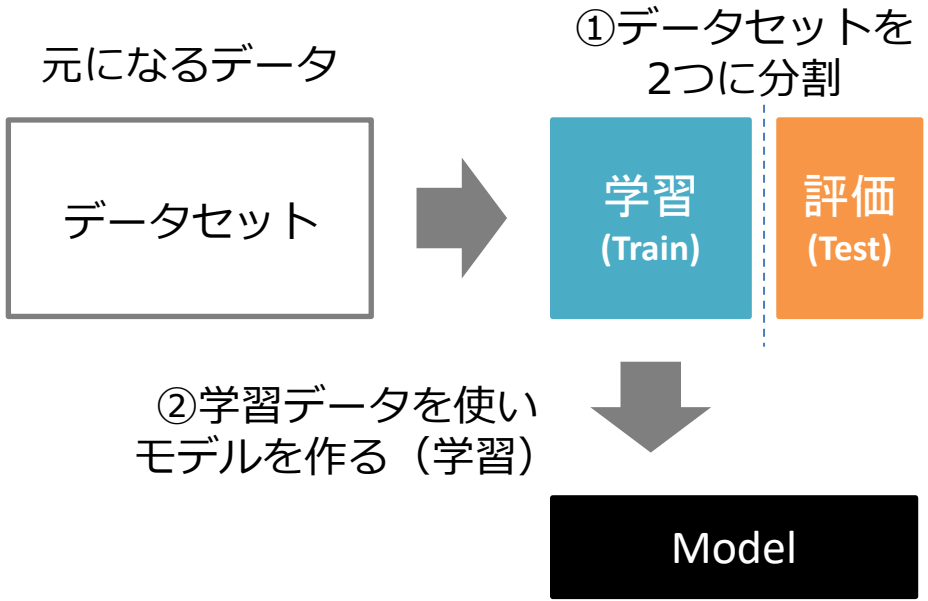
未知のデータを予測できる？

“**未知のデータ**でも予測できるか？（汎用性）”
を評価することが大切



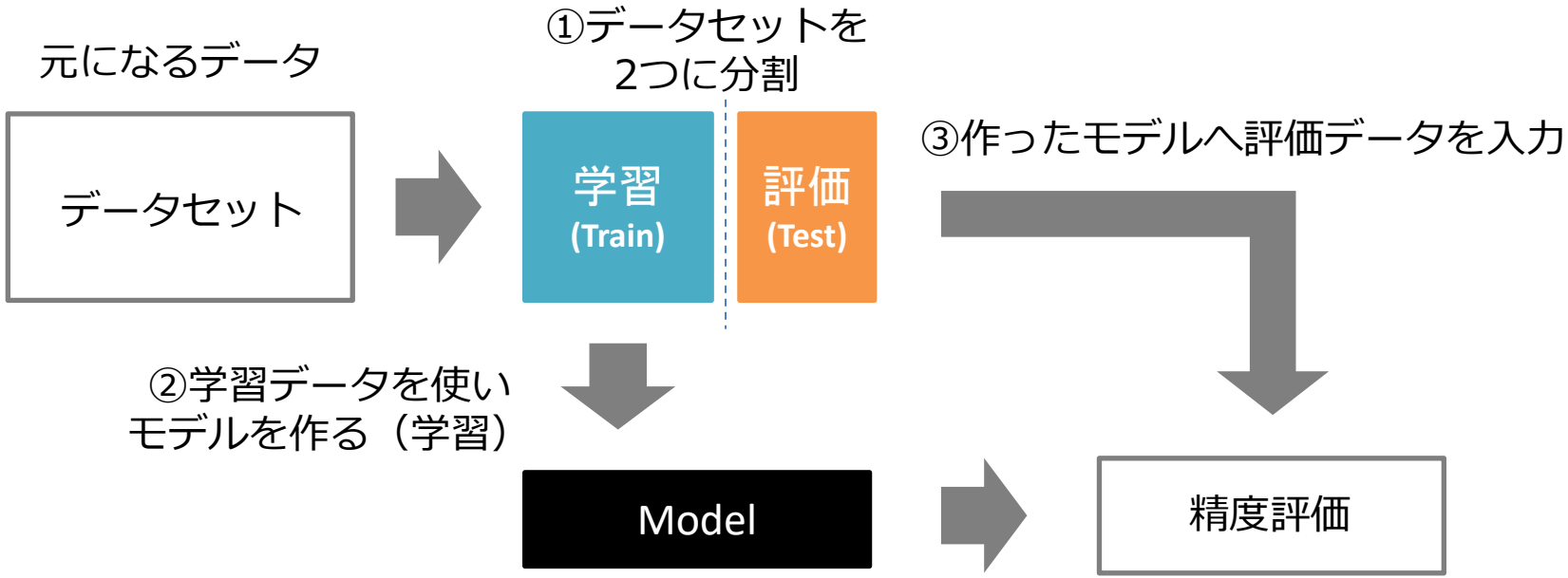
未知のデータを予測できる？

“**未知のデータ**でも予測できるか？（汎用性）”
を評価することが大切



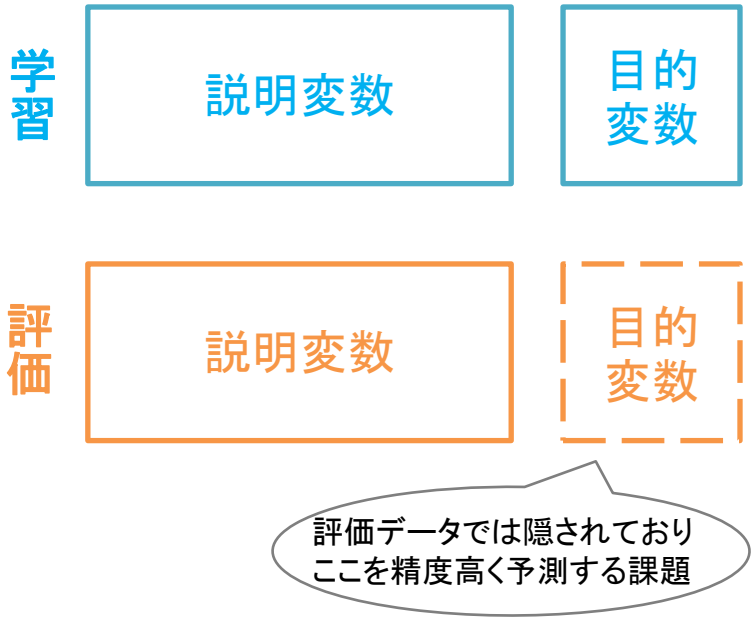
未知のデータを予測できる？

“**未知のデータ**でも予測できるか？（汎用性）”
を評価することが大切



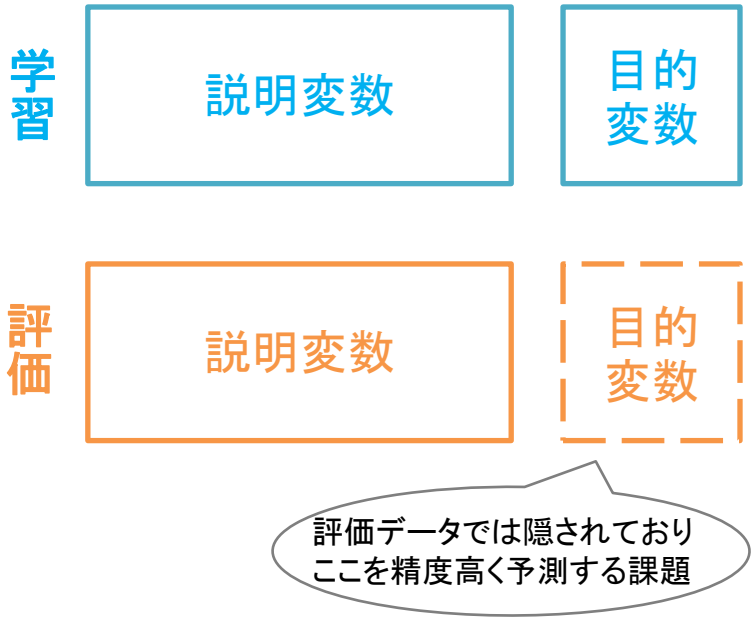
Challenge Missionでは

既にtrainとtestに分割されて配布されます



Challenge Missionでは

既にtrainとtestに分割されて配布されます



【分析のステップ】

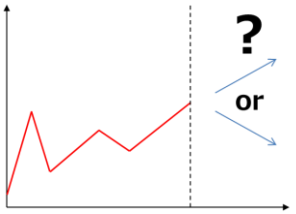
①学習データを使い、モデルを作成する

②学習データを使い、良いモデルが作れているかどうかを確認する

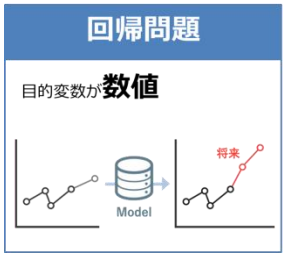
③作成したモデルを使い、評価データを予測し、投稿する

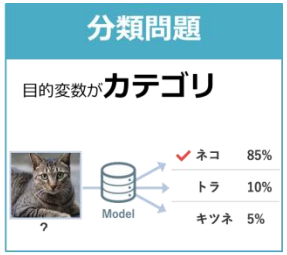
まとめ

- **予測とは**
  - データに基づき、値を想定すること

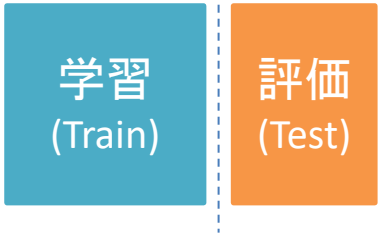


- **代表的な予測問題**
  - 回帰と分類





- **汎用的な予測モデルを作るべし**
  - その為に、Train/Testにデータを分割
  - 過学習にも注意しよう



モデリングのキホンの「キ」

モデリングの手順

① **説明変数を決め、データを準備**
- どのデータを使ってモデルを作るか？を決めます
- 欠損がある場合は前処理が必要です
- **学習**（Train）データからは説明変数と目的変数、**評価**（Test）データからは説明変数のみを取り出す
- 学習データと評価データから取り出す説明変数は同じでなければなりません

② **モデルの準備**
- どの手法を使ってモデルを作るか？を決めます

③ **モデルの作成**
- 学習データから取り出した説明変数と目的変数のデータを使い、モデルを作成します

④ **モデルを使い予測**
- 評価データから取り出した説明変数のデータを使い、②で作ったモデルに当てはめることで、予測値を出します

⑤ **モデルの評価**
- 評価データに対するモデルの予測結果と実際の値を比較することでモデルを評価します

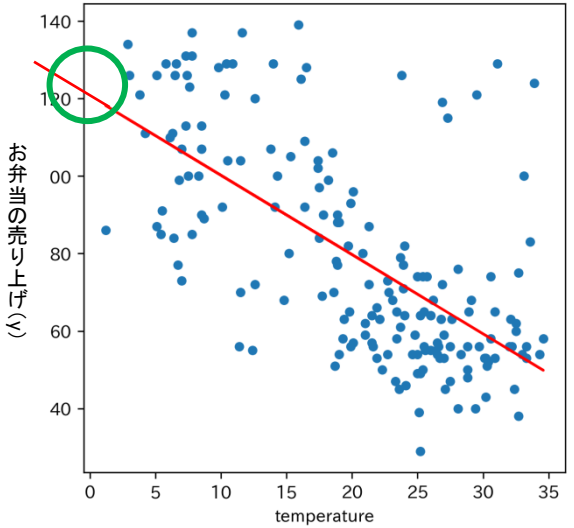
単回帰モデル

- １つの目的変数を１つの説明変数のみでモデル化する方法

$$y = \underline{ax} + \underline{b}$$

傾き　　切片

傾き…直線の傾き具合、正だと右肩上がり、負だと右肩下がり
切片…直線とx=0がぶつかるところ（右図だと○部分）



$$y = -2x + 120.6$$

モデルの評価方法

- モデルの評価はとても重要
- 予測問題に応じて評価方法が異なる

<E.g.>

- 豆腐の需要予測
  - 予測値と実測値の誤差がどのくらいあるか？
- ネット広告のクリック予測
  - 予測がどれくらい信用できるか？
- 検索エンジン
  - ユーザが見たい記事が検索できているか？

モデルの評価方法

- モデルの評価はとても重要
- 予測問題に応じて評価方法が異なる

<E.g.>

- 豆腐の需要予測
  - 予測値と実測値の誤差がどのくらいあるか？
- ネット広告のクリック予測
  - 予測がどれくらい信用できるか？
- 検索エンジン
  - ユーザが見たい記事が検索できているか？

評価関数

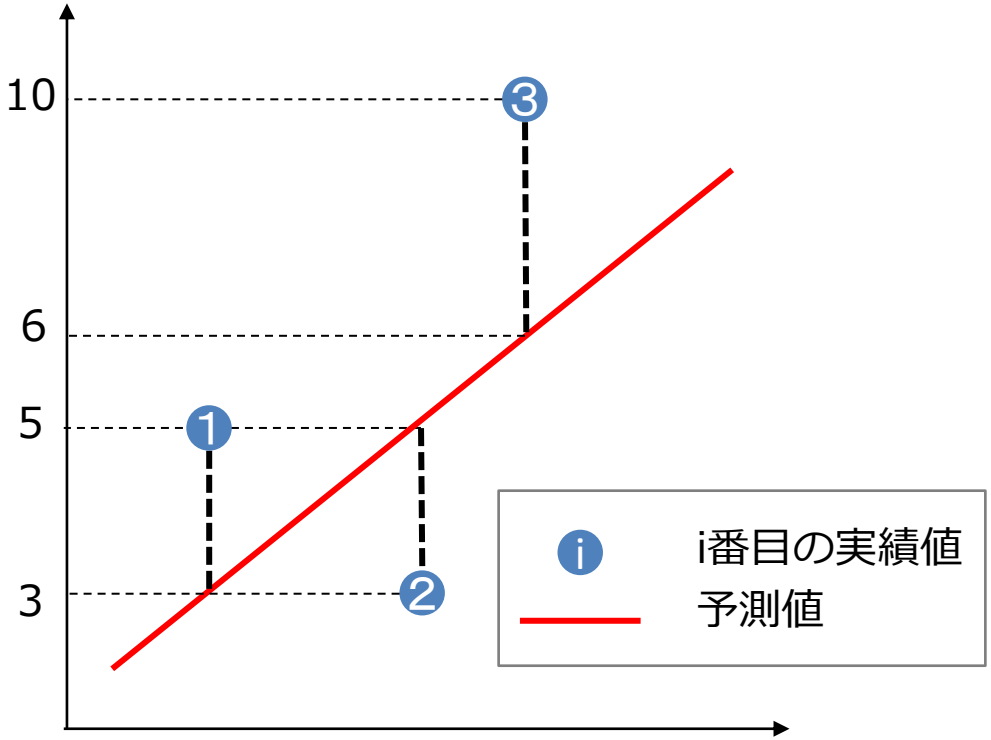
評価関数とは？

- モデルの予測精度を評価する数式
  - 今回はRoot Mean Squared Error(RMSE)
  - RMSEは誤差を表す指標の為、少ないほど良い

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_i - p_i)^2}$$

N:予測対象数
y_i:i番目の実績値
p_i:i番目の予測値

RMSEの計算の例



	1	2	3
実績値	5	3	10
予測値	3	5	6
実績値-予測値	2	-2	4
↑の二乗	4	4	16

$$RMSE = \sqrt{\frac{1}{3}(4+4+16)} = \sqrt{8} \fallingdotseq \mathbf{2.828}$$

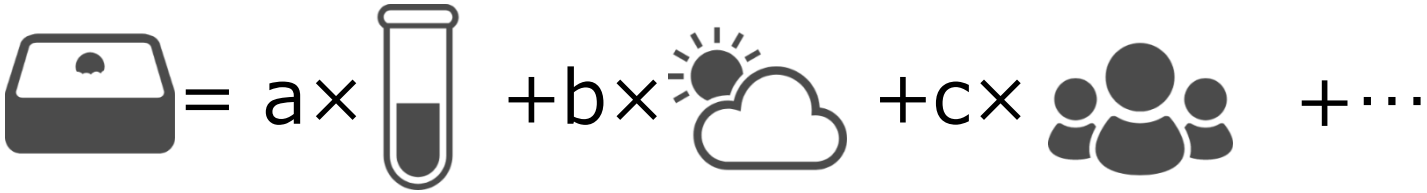
色々な評価関数

評価関数	予測対象	値域	見方
AUC	分類精度を測る e.g) 医療診断	0～1	値が大きいほど良い
LogLoss	分類精度を測る e.g) 画像分類	0～∞	値が小さいほど良い
Accuracy	分類精度を測る e.g) 画像分類	0～1	値が大きいほど良い
Precision	正確性を測る e.g) 検索エンジン	0～1	値が大きいほど良い
Recall	カバー率を測る e.g) 検索エンジン	0～1	値が大きいほど良い
MAE	誤差を測る e.g) 需要予測	0～∞	値が小さいほど良い
MAP@N	検出精度を測る e.g) 推薦エンジン	0～1	値が大きいほど良い
nDCG	ランキング精度を測る e.g) 推薦エンジン	0～1	値が大きいほど良い

重回帰とは？

- ２つ以上の説明変数を使った回帰モデル
  - 予測する為の手掛かりをたくさん使う

$$y = ax_1 + bx_2 + cx_3 + \cdots$$

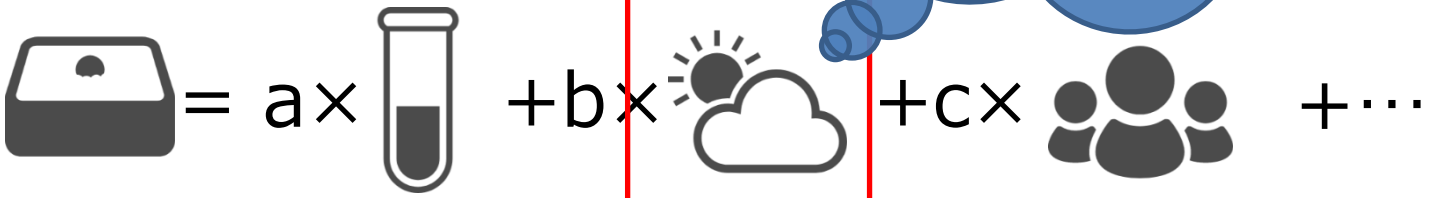


お弁当の売上　気温　天気　来店数

重回帰とは？

- ２つ以上の説明変数を使った回帰モデル
  - 予測する為の手掛かりをたくさん使う

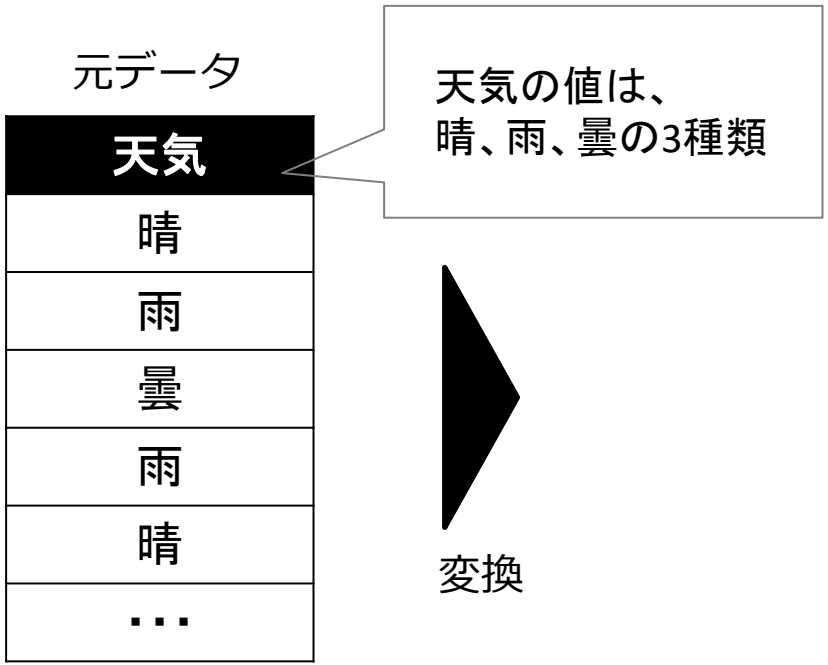
$$y = ax_1 + bx_2 + cx_3 + \cdots$$



数値じゃないのにどうするの？？？

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）



ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

変換

1-of-K表現

晴	雨	曇
…		

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

変換

1-of-K表現

晴	雨	曇
1	0	0
…		

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

1-of-K表現

晴	雨	曇
1	0	0
0	1	0
…		

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

1-of-K表現

晴	雨	曇
1	0	0
0	1	0
0	0	1
…		

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

1-of-K表現

晴	雨	曇
1	0	0
0	1	0
0	0	1
0	1	0
…		

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

1-of-K表現

晴	雨	曇
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0
…		

ダミー変数

- 質的データを数値データへ変換する
  - 1-of-K表現（one hot encoder）

元データ

天気
晴
雨
曇
雨
晴
…

変換

1-of-K表現

晴	雨	曇
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0
…		

予測モデルの予測精度を上げるには？

- 特徴量＝説明変数
  - 機械学習の領域で使われる
- 予測精度を上げる為には？

①特徴量を作る

②特徴量を選ぶ

特徴量の作成

- 与えられたデータや外部データを加工し、予測の手掛かりとなりそうな新たな特徴を作ること

例１）基本統計量を作る

日付	店舗A	店舗B
4/1	100	50
4/2	60	40
4/3	70	50
4/4	140	90

AとBの平均
75
50
60
115

例２）データを集約する

Id	name	age
001	山田	23
002	鈴木	31
003	斉藤	18
004	藤井	29

年齢層
20代
30代
10代
20代

特徴量の選択

- 特徴量が多すぎるのも良くない
  - 過学習のリスクがある

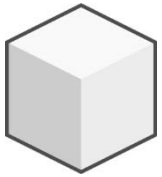
- 数ある特徴量から重要なもののみを選択することが大切
  - 単変量解析
    - 目的変数と各説明変数を1対1で確認し、取捨選択
    - E.g.) 分散分析など
  - モデルベース選択
    - モデルにとっての各変数の重要度を算出し、取捨選択
    - E.g.) ツリー系機械学習手法の重要度をみる
  - 反復選択
    - 特徴量を増減させながらモデルを生成し、良い特徴量を探索する
    - E.g.) ステップワイズ法など

「特徴量を作ってみよう」の前に…

- 特徴量を作る為にはデータ加工が必要
- データ加工に便利な２つの関数
  - split関数
    - 文字列を分割する関数
  - apply関数
    - データの各値に数式や関数を適用する関数

split関数

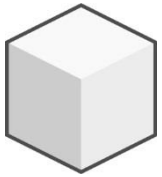
- 文字列を指定した文字で分割する関数



.　split(”区切り文字”)

split関数

- 文字列を指定した文字で分割する関数



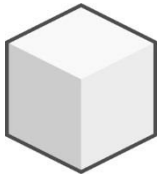
. split(”区切り文字”)

<E.g>

terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

split関数

- 文字列を指定した文字で分割する関数



. split(”区切り文字”)

<E.g>

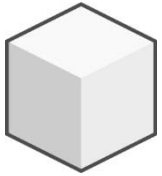
terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

⇒colors = terms.split(“,”)

カンマ

split関数

- 文字列を指定した文字で分割する関数



. split(”区切り文字”)

<E.g>

terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

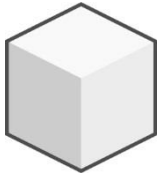
⇒colors = terms.split(“,”)

★colorsの中身を取り出したい

[“red”, “blue”, “green”, “yellow”]

split関数

- 文字列を指定した文字で分割する関数



. split(”区切り文字”)

<E.g>

terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

⇒colors = terms.split(“,”)

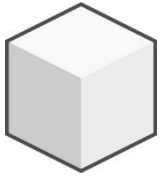
★colorsの中身を取り出したい

[“red”, “blue”, “green”, “yellow”]

0 1 2 3

split関数

- 文字列を指定した文字で分割する関数



. split(”区切り文字”)

<E.g>

terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

⇒colors = terms.split(“,”)

★colorsの中身を取り出したい

[“red”, “blue”, “green”, “yellow”]

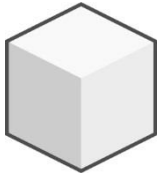
0 1 2 3

redを取り出したい

yellowを取り出したい

split関数

- 文字列を指定した文字で分割する関数



. split(”区切り文字”)

<E.g>

terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

⇒colors = terms.split(“,”)

★colorsの中身を取り出したい

[“red”, “blue”, “green”, “yellow”]

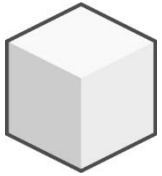
0 1 2 3

redを取り出したい⇒colors[**0**]

yellowを取り出したい

split関数

- 文字列を指定した文字で分割する関数



. split(”区切り文字”)

<E.g>

terms = “red,blue,green,yellow”を**カンマ**で区切り、結果を変数colorsに代入したい

⇒colors = terms.split(“,”)

★colorsの中身を取り出したい

[“red”, “blue”, “green”, “yellow”]

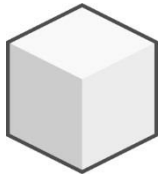
0 1 2 3

redを取り出したい⇒colors[**0**]

yellowを取り出したい⇒colors[**3**]

apply関数

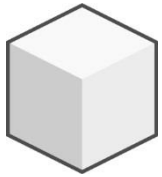
- 各値に数式や関数を適用する為の関数（pandas）



.　apply（lambda x : 〇〇）

apply関数

- 各値に数式や関数を適用する為の関数（pandas）



.　apply（lambda x : ○○）

数式 or 関数

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

dat["value"].apply(lambda **x** : x*2+1)

dat =

Index	value
1	4
2	7
3	3
4	8

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

dat[“value”].apply(lambda x : x*2+1)

4*2+1

dat =

Index	value
1	4
2	7
3	3
4	8

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

4*2+1

9

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

dat["value"].apply(lambda **x** : x*2+1)

dat =

Index	value
1	4
2	7
3	3
4	8

9

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

dat[“value”].apply(lambda x : x*2+1)

dat =

Index	value
1	4
2	7
3	3
4	8

7*2+1

9

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

7*2+1

9
15

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

dat[“value”].apply(lambda **x** : x*2+1)

dat =

Index	value
1	4
2	7
3	3
4	8

9
15

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

3*2+1

dat =

Index	value
1	4
2	7
3	3
4	8

9
15

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

3*2+1

9
15
7

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

9
15
7

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

8*2+1

dat =

Index	value
1	4
2	7
3	3
4	8

9
15
7

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

8*2+1

9
15
7
17

apply関数

- 各値に数式や関数を適用する為の関数（pandas）

<E.g> datのvalueの各値を2倍して1を足したい

```
dat["value"].apply(lambda x : x*2+1)
```

dat =

Index	value
1	4
2	7
3	3
4	8

各値に「x*2+1」という
数式が適用された

9
15
7
17

次回について

次回について

次回からグループワークのはじまりです！

次回の宿題

①グループ課題

以下をグループで話し、資料にまとめて次回発表してください
（5分程度、スライドフォーマットは自由）

- チーム名を決める
- チームメイトの自己紹介
- チーム名のコンセプト
- Jリーグについて調べたことの発表
  e.g) どうやって勝ち負けが決まる？何試合？何チーム？等々

②E-learning課題

- **自動車環境性能の改善（6h）**
- スポーツのチケット価格の最適化
- pandas入門道場（2h）
- 評価関数入門（3h）

おしまい

Let's Enjoy DataScience!!!