

授業準備：Webclassからコードをダウンロードし、
Google colaboratoryで開いておいてください

演習授業中の質問対応について

演習授業中の質問をチューターの先生が対応させていただきます。

演習にエラーが出たなど問題があったらリアクションの挙手を押してください。

質問内容を入力して、「全員」宛てに送信してください。



2024/1/17 10:40-11:25

演習9

回帰2

本スライドは、自由にお使いください。
使用した場合は、このQRコードからアンケート
に回答をお願いします。



統合教育機構
石丸美穂

線形回帰のコードのまとめ

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

STEP0 : ライブラリの読み込み

```
from sklearn.datasets import load_diabetes
dm = load_diabetes(as_frame = True, scaled = False)
y_dm = dm.target
x_dm = dm.data
x_bmi = x_dm[['bmi']]
```

STEP1 : データの準備

```
plt.scatter(x_bmi, y_dm)
```

```
from sklearn.linear_model import LinearRegression
model_bmi = LinearRegression()
```

STEP2 : 学習モデルの選択

```
model_bmi.fit(x_bmi, y_dm)
```

STEP3 : データを入れて学習させる

```
print(model_bmi.coef_)
print(model_bmi.intercept_)
```

STEP4 : 傾きと切片を求める

```
pre = pd.DataFrame([[20]], columns=['bmi'])
print(model_bmi.predict(pre))
```

STEP5 : 予測を行う

```
print(model_bmi.score(x_bmi, y_dm))
```

STEP6 : モデルの評価

線形回帰分析

前回の復習

STEP0 : 事前準備

STEP1 : データの用意

STEP2 : 学習モデルの選択

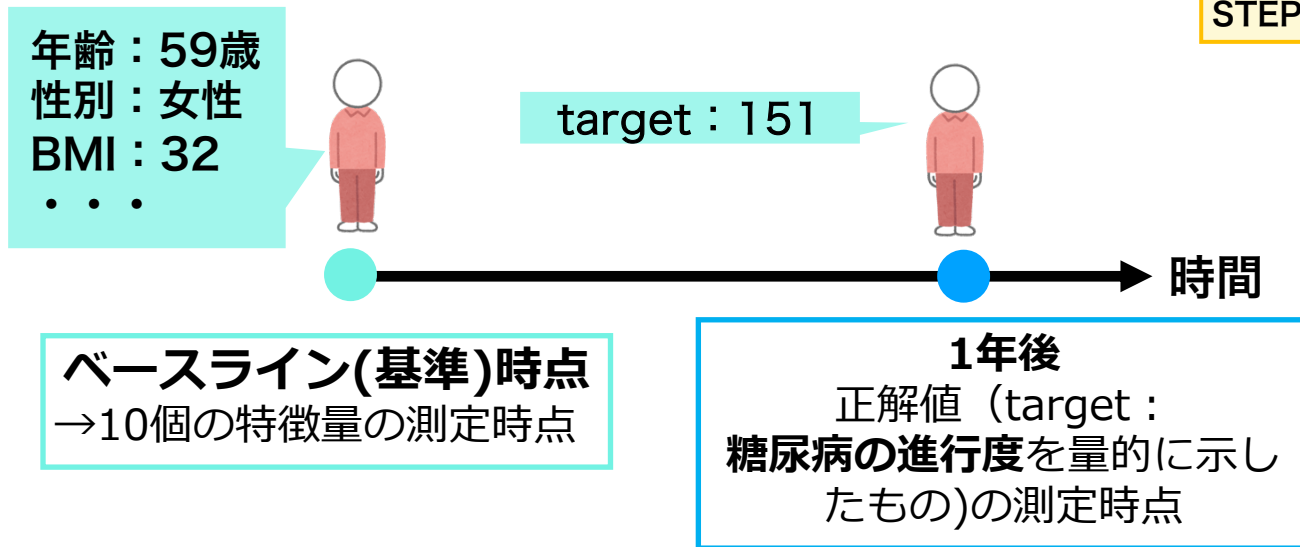
STEP3 : データを入れて学習させる

STEP4 : 傾きと切片を求める

STEP5 : 予測を行う

STEP6 : モデルの評価

STEP1 : 糖尿病データの内容確認



ベースライン時点の x (特徴量)で1年後の $target$ を y (正解値) との関連を調べる、学習モデルの作成を行う

回帰で学習を行うには、xとyの配列がPythonでは指定されている

x_bmi,

y_dm

xは2次元配列 (行列の形)

yは1次元配列 (ベクトルの形)

151.0, 75.0, 141.0...

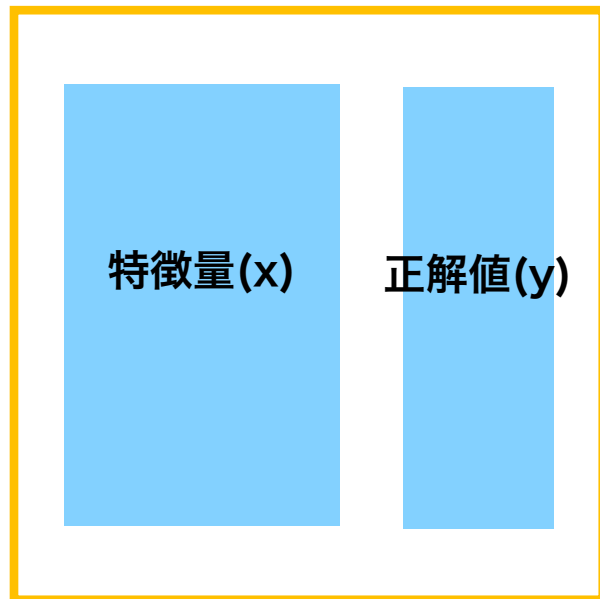
	BMI
0	32.1
1	21.6
2	30.5

配列を間違えるとエラーになりコードが動かないので、
作成したデータの配列を確認するように気を付ける

442 rows x 1 columns

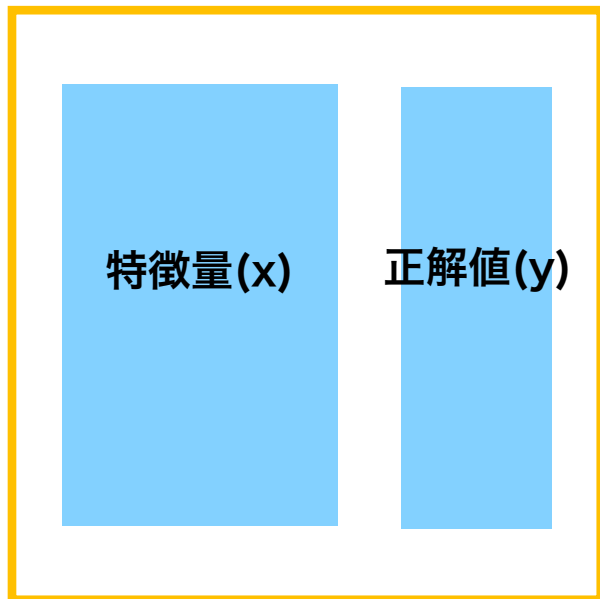
機械学習の流れのまとめ

STEP1:データの用意



機械学習の流れのまとめ

STEP1:データの用意



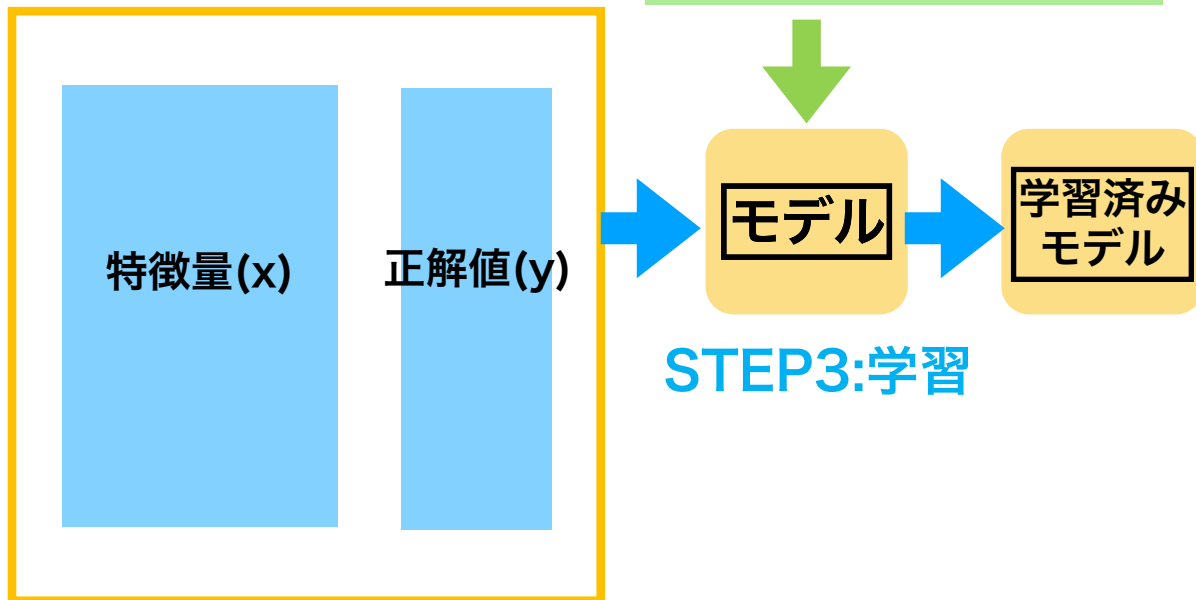
STEP2:学習モデルの選定



機械学習の流れのまとめ

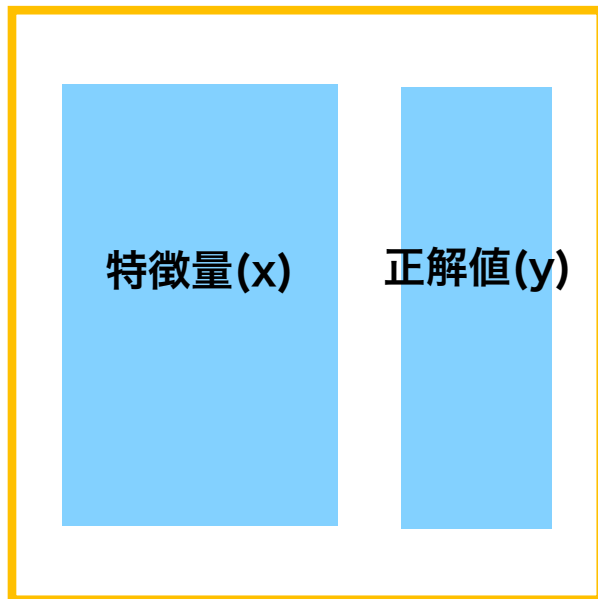
STEP1:データの用意

STEP2:学習モデルの選定

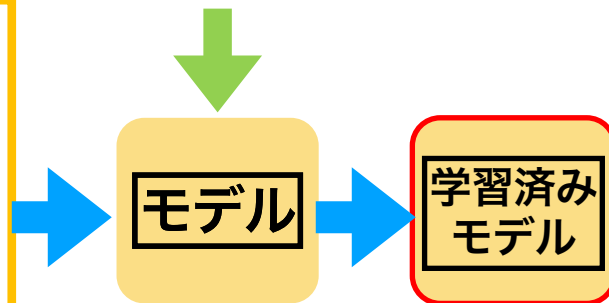


機械学習の流れのまとめ

STEP1:データの用意



STEP2:学習モデルの選定

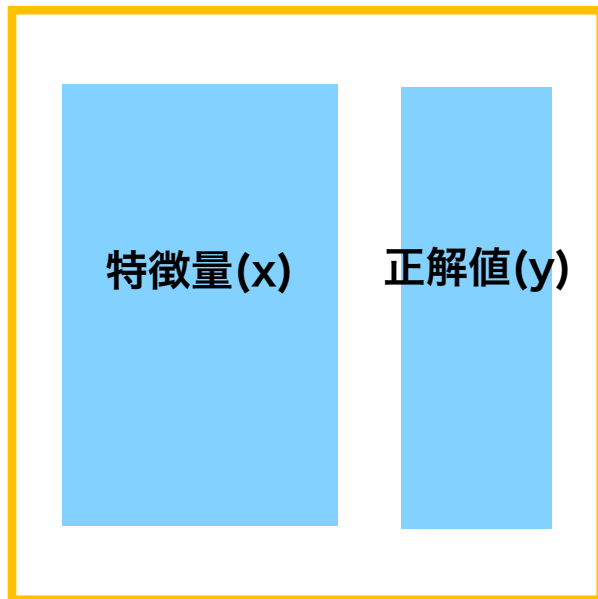


STEP3:学習

STEP4:傾きと切片

機械学習の流れのまとめ

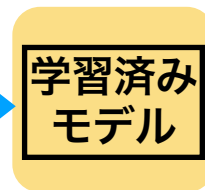
STEP1:データの用意



STEP2:学習モデルの選定



STEP4:傾きと切片



STEP3:学習

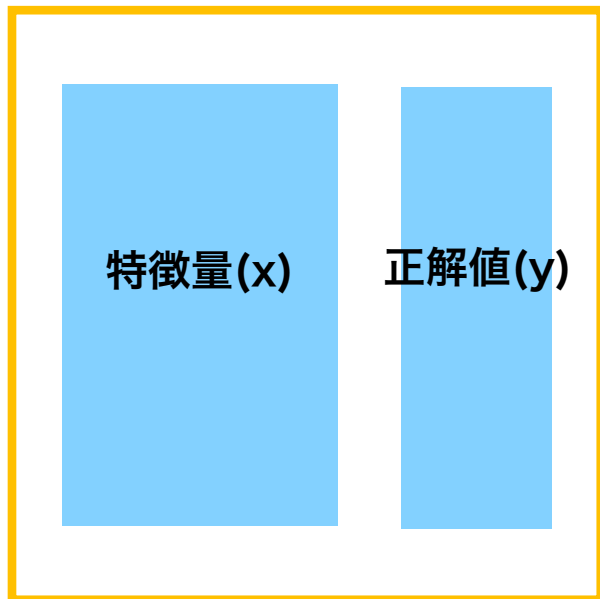


STEP5:予測

新しいxの値

機械学習の流れのまとめ

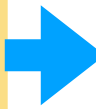
STEP1:データの用意



STEP2:学習モデルの選定



STEP4:傾きと切片



STEP3:学習



新しいxの値

STEP5:予測



STEP6:モデルの評価

線形回帰分析

STEP0 : Google Colaboratoryの立ち上げ

STEP0 : 事前準備

STEP1 : データの用意

STEP2 : 学習モデルの選択

STEP3 : データを入れて学習させる

STEP4 : 傾きと切片を求める

STEP5 : 予測を行う

STEP6 : モデルの評価

検索google colab [Colaboratory へようこそ - Colaboratory - Google](#)



演習9コード.ipynb

検索google colab

[Colaboratory へようこそ - Colaboratory - Google](#)

ノートブックを開く

例 >

最近 >

Google ドラ
イブ >

GitHub >

アップロード >

1
3



参照

または、ここにファイルをドラッグしてください

演習9コード.ipynb

線形回帰分析

STEP1：データの用意

STEP0：事前準備

STEP1：データの用意

STEP2：学習モデルの選択

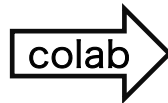
STEP3：データを入れて学習させる

STEP4：傾きと切片を求める

STEP5：予測を行う

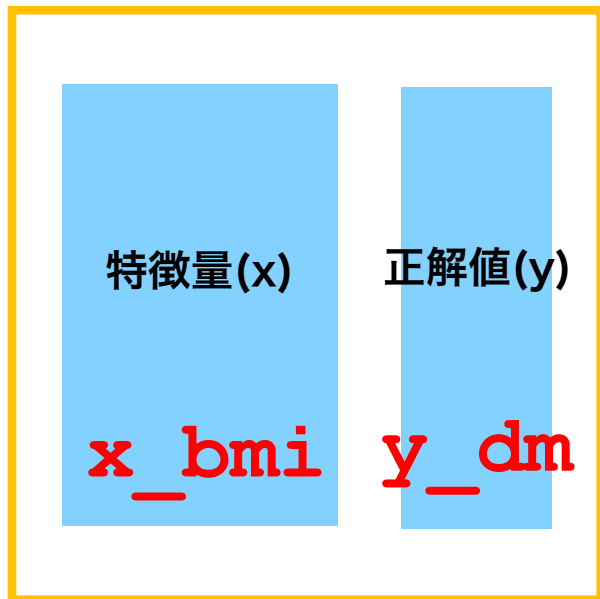
STEP6：モデルの評価

前回の続きのため、配布した演習9.ipynbの
コード9-1~コード9-2を実行してください



機械学習の流れのまとめ

STEP1:データの用意



STEP2:学習モデルの選択



線形回帰分析

STEP2：学習モデルの選択

STEP0：事前準備

STEP1：データの用意

STEP2：学習モデルの選択

STEP3：データを入れて学習させる

STEP4：傾きと切片を求める

STEP5：予測を行う

STEP6：モデルの評価

コード9-3 線形回帰を学習モデルに選択

```
from sklearn.linear_model import LinearRegression
```

クラスだけをインポート `from sklearn.linear_model import LinearRegression`

ライブラリ名

モジュール名

クラス名

クラス (Class)

属性 (引数、データ) と メソッド (処理) を一つにまとめた設計図のようなもの

クラス (設計図)に実際のデータを入れたものをインスタンスと呼ぶ

```
class BMI(weight, height)
```

属性

体重 (weight)

身長 (height)

メソッド

bmi_calc():

Weight /Height/ Height

Aさん



体重 70kg
身長 1.7m

```
Asan = BMI(70, 1.7)
```

Asan.weight

➡ 70

Asan.bmi_calc()

➡ 24.2

Bさん



体重 45kg
身長 1.5m

```
Bsan = BMI(45, 1.5)
```

Bsan.weight

➡ 45

Bsan.bmi_calc()

➡ 20

BMIクラスからインスタンス(Asan, Bsan)を作成

線形回帰分析

STEP2：学習モデルの選択

STEP0：事前準備

STEP1：データの用意

STEP2：学習モデルの選択

STEP3：データを入れて学習させる

STEP4：傾きと切片を求める

STEP5：予測を行う

STEP6：モデルの評価

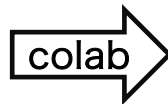
コード9-3 線形回帰を学習モデルに選択

```
model_bmi = LinearRegression()
```

- `LinearRegression()` クラスから `model_bmi` インスタンスを作成する
(この時点ではモデル構造のみが指定されている)

モデル名は何でも良い → 今回は `model_bmi`

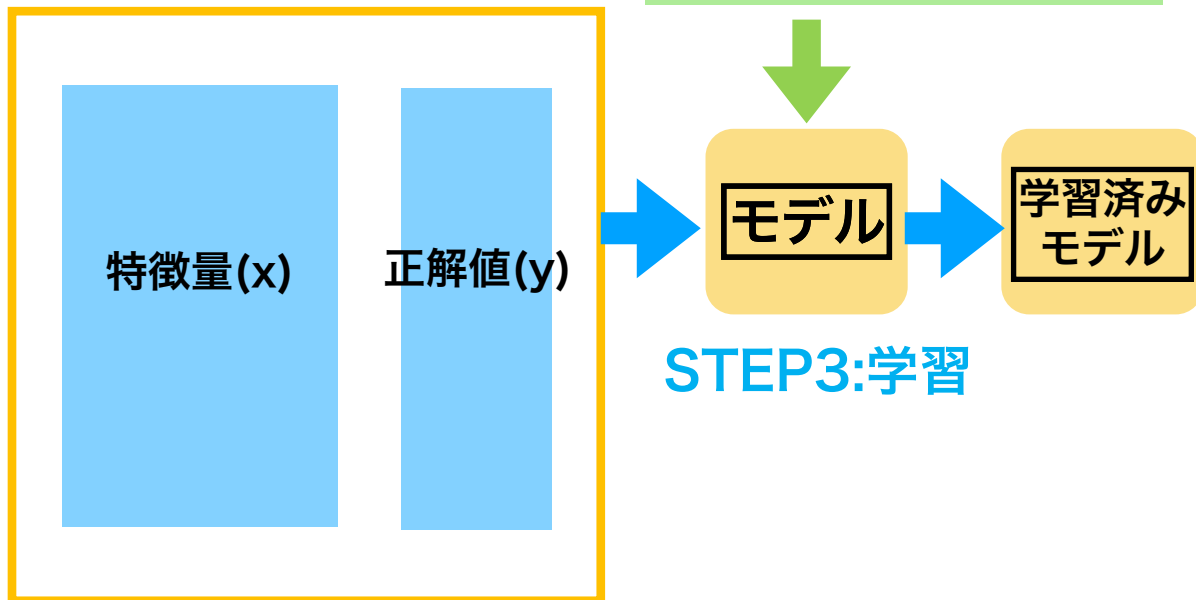
* `model1, mode2...` と作ると何を入れたものか忘れるので、
何のモデルなのか明示しておく方が便利



機械学習の流れのまとめ

STEP1:データの用意

STEP2:学習モデルの選定



線形回帰分析

STEP3：データを入れて学習させる

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

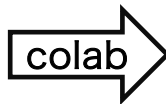
コード9-4 データを入れて学習させる

```
model_bmi.fit(x_bmi, y_dm)
```

- (モデル名).**fit(x,y)**メソッドで学習できる
- 今回の特徴量はx_bmi, 正解値はy_dmのデータ

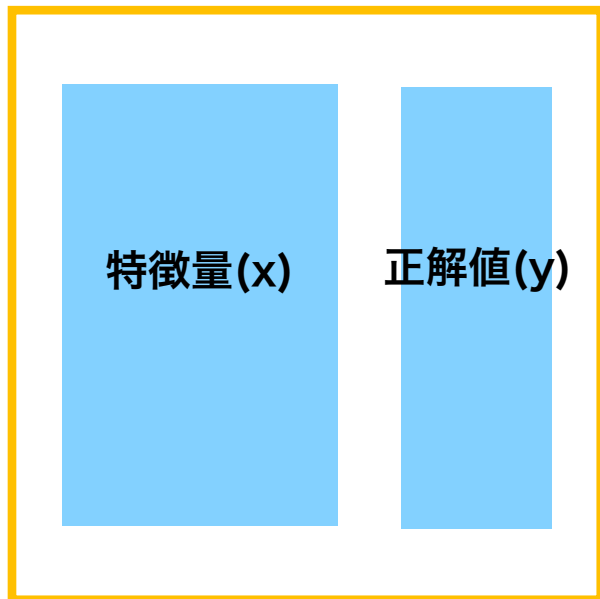
 **これだけで「学習」は完了!**

この後からはmodel_bmiは学習済みモデルになる

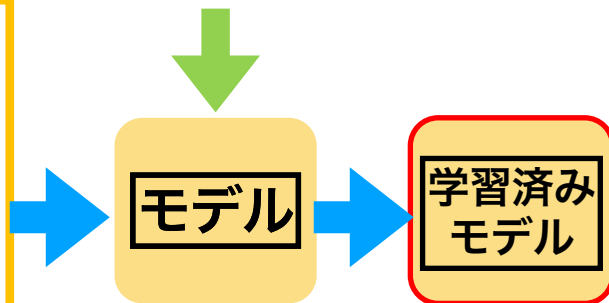


機械学習の流れのまとめ

STEP1:データの用意



STEP2:学習モデルの選定



STEP3:学習

STEP4:傾きと切片

線形回帰分析

STEP4：傾きと切片を求める

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-5 学習済みモデルの傾きと切片を求める

```
print(model_bmi.coef_)  
print(model_bmi.intercept_)
```

→ `[[10.23312787]]`

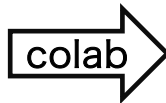
`[-117.77336657]`

(モデル名).coef_：係数 (coefficient)

(モデル名).intercept_：切片 (intercept)

model_bmiの回帰式の完成!!

$$y = -117.77 + (10.23 \times \text{BMI})$$

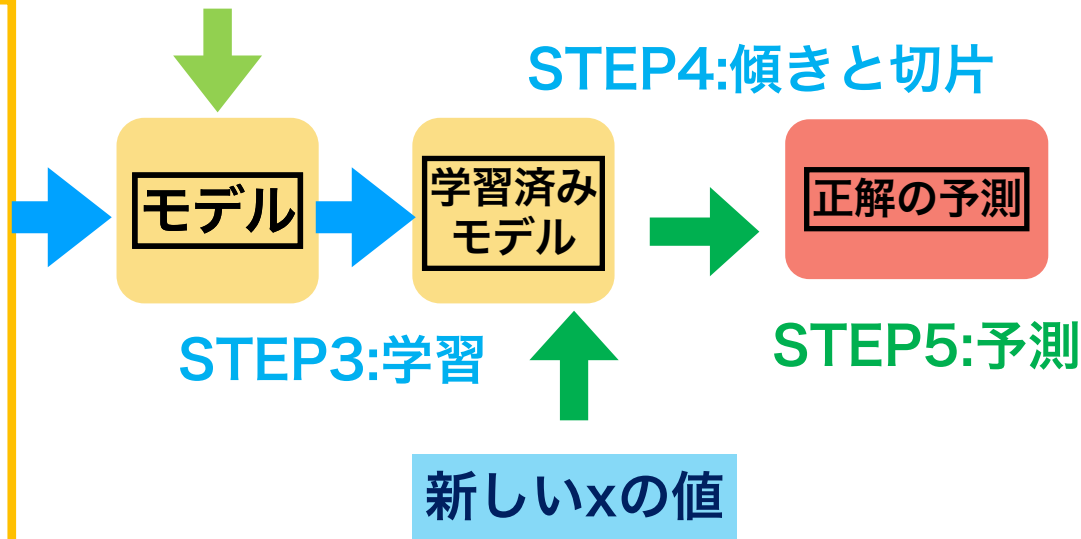
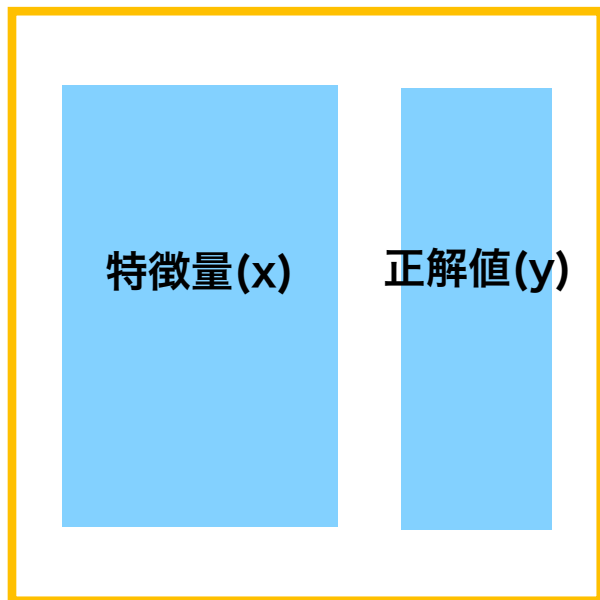


機械学習の流れのまとめ

STEP1:データの用意

STEP2:学習モデルの選定

STEP4:傾きと切片



線形回帰分析

STEP5：正解値の予測

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-6 新しいxの値 (bmi=20の人) データ作成

```
pre = pd.DataFrame([[20]], columns=['bmi'])
```

- preという変数に、bmi=20の1行1列のデータを作成する
 - x (特徴量データ) は2次元配列である必要がある
- ➡ そのため、今回はpd.DataFrame型にして、column名に'bmi'をつける



線形回帰分析

STEP5：正解値の予測

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-7 作成したpreのデータを確認

```
pre #データを出力
```



bmi

```
0 20
```

```
pre.shape #データの配列構造を出力
```



```
(1, 1)
```

1行1列の2次元配列(行列)

```
type(pre) #データの型を出力
```



```
pandas.core.frame.DataFrame
```

線形回帰分析

STEP5：正解値の予測

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-8 新しいxの値(pre)で正解値の予測

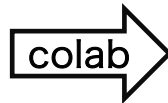
```
print(model_bmi.predict(pre))
```

➡ [86.88919084]

- (モデル名).predict(pre) でmodel_bmiの学習済みモデルのxにpreを代入し、yの値を求める
- print() で結果を出力する

STEP4で求めた回帰式にbmi=20を代入する

$$y = -117.77336657 + (10.2331787 \times 20) = \underline{86.88919084}$$



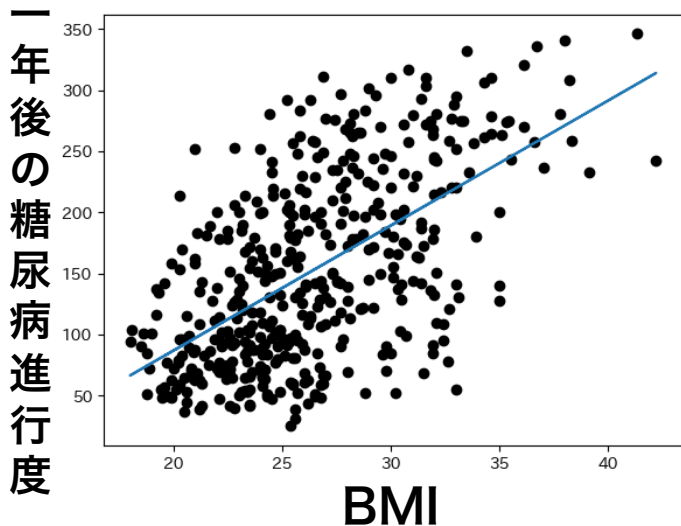
線形回帰分析

STEP5：プロット

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-9 散布図と回帰線を描出する

```
plt.scatter(x_bmi, y_dm, color='black')  
plt.plot(x_bmi, model_bmi.predict(x_bmi))
```



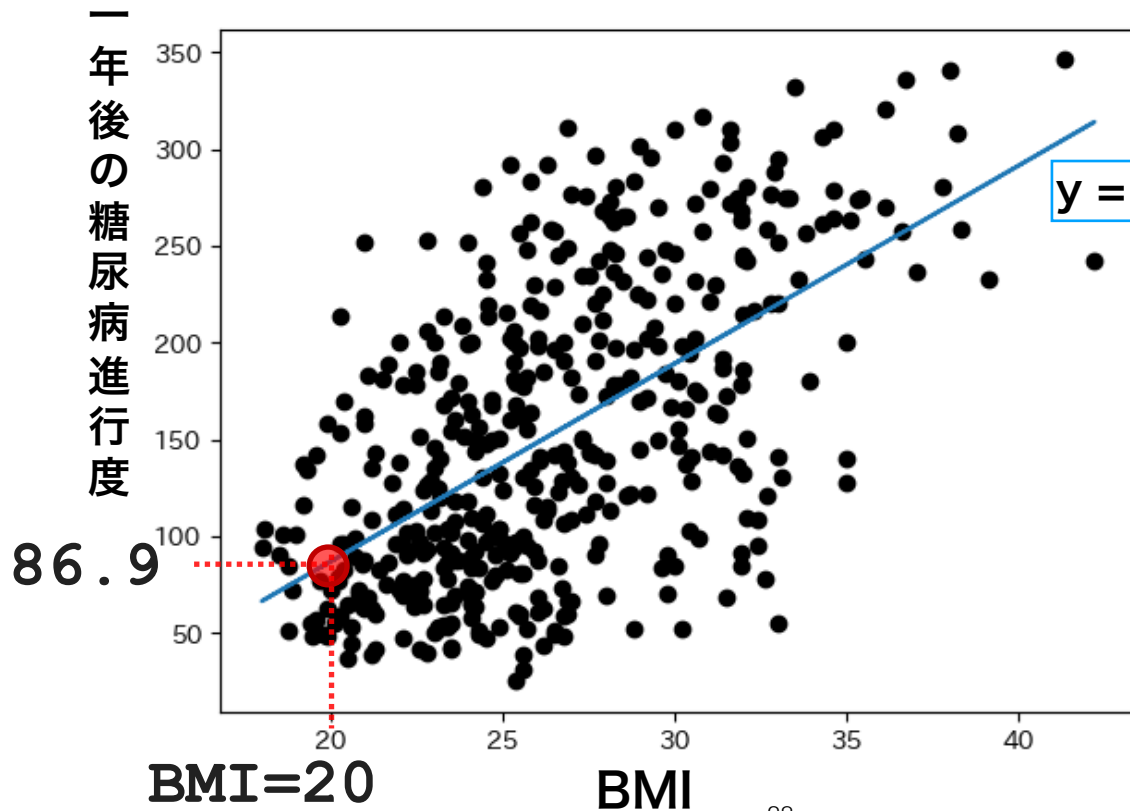
$$y = -117.77 + (10.23 \times \text{BMI})$$

- `plt.plot(x,y)` で各点を実線で結ぶ線が描出できる
- 回帰線のyは実測値ではなく、モデルで予測されたyの値であるため、`(モデル名).predict(x_bmi)` となる

線形回帰分析

STEP5：プロット

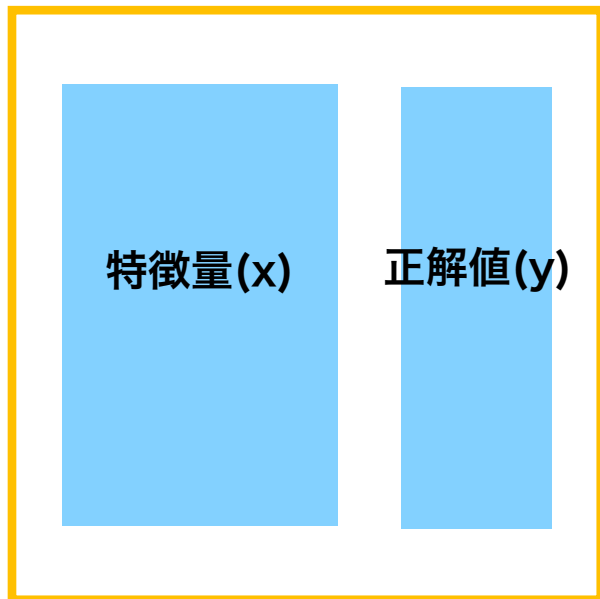
STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価



$$y = -117.77 + (10.23 \times \text{BMI})$$

機械学習の流れのまとめ

STEP1:データの用意



STEP2:学習モデルの選定



STEP3:学習

STEP4:傾きと切片



STEP5:予測



新しいxの値

正解との比較・評価

STEP6:モデルの評価

線形回帰分析

STEP6：モデルの予測性能評価

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-10 モデルがどれくらい予測できているかを調べる

```
print(model_bmi.score(x_bmi, y_dm))
```

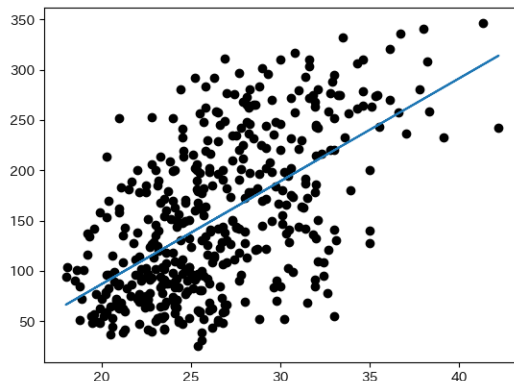
➡ 0.3439237602253802

(モデル名).score(x,y) で決定係数 (R^2) を計算する

*決定係数：xとyの相関係数の二乗の値

1~0の範囲をとる

1に近いほどxでyを予測していると言える



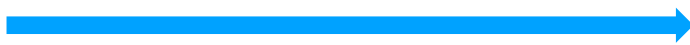
線形回帰分析

STEP6：モデルの予測性能評価

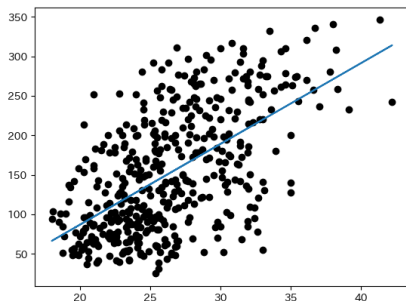
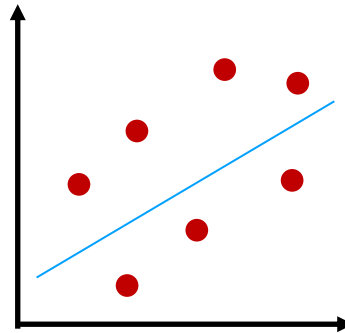
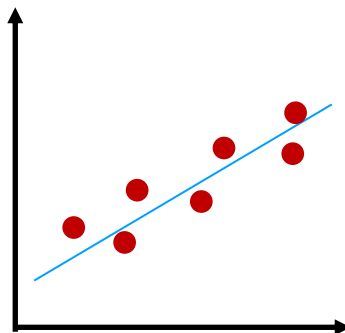
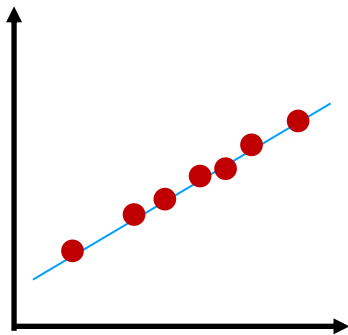
決定係数 R^2

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

高い



低い



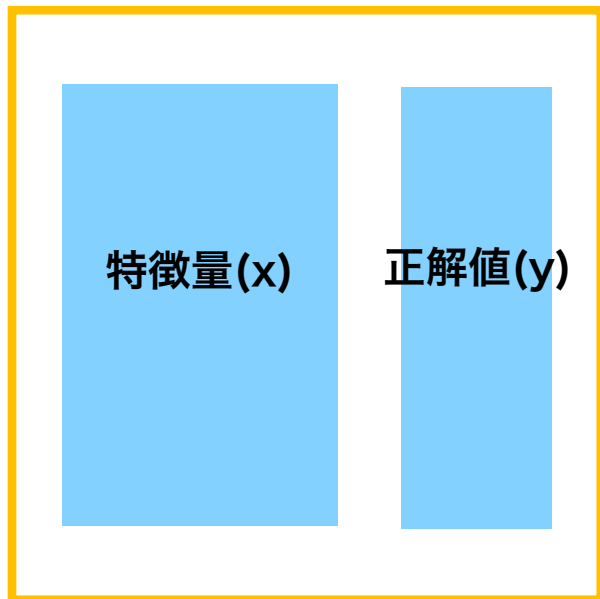
$$R^2 = 0.34$$



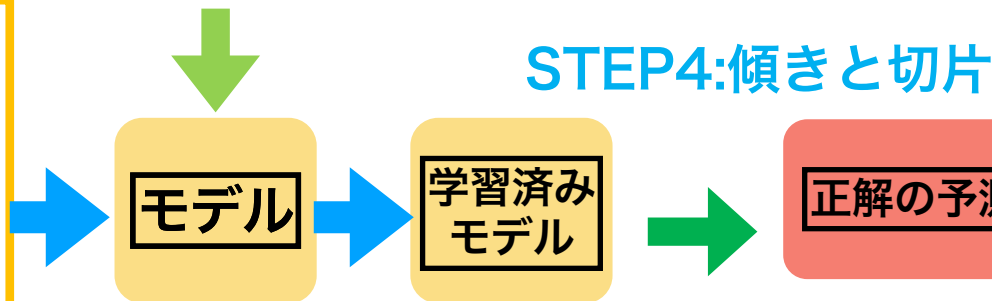
今回のモデルはそれほど
予測性能は高くない

機械学習の流れのまとめ

STEP1:データの用意



STEP2:学習モデルの選定



STEP3:学習

新しいxの値

STEP4:傾きと切片



STEP5:予測

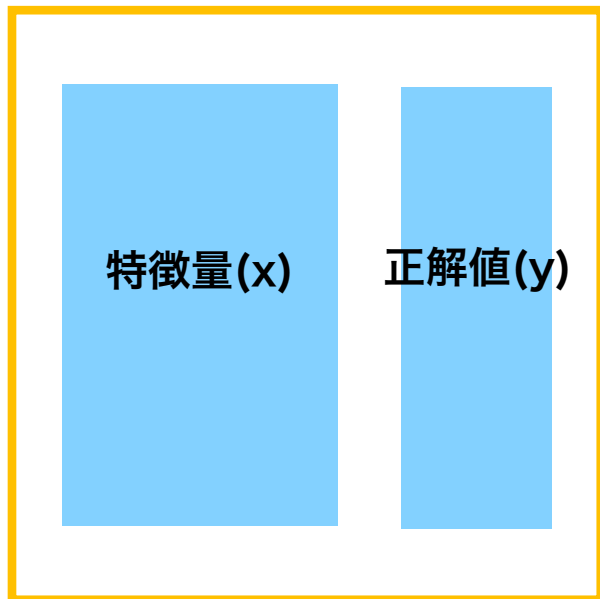


STEP6:モデルの評価

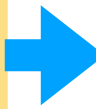
新しく `model_dm10` で 10 個の特徴
量で線形回帰を行う

機械学習の流れのまとめ

STEP1:データの用意



STEP2:学習モデルの選定



STEP3:学習

STEP4:傾きと切片



STEP5:予測

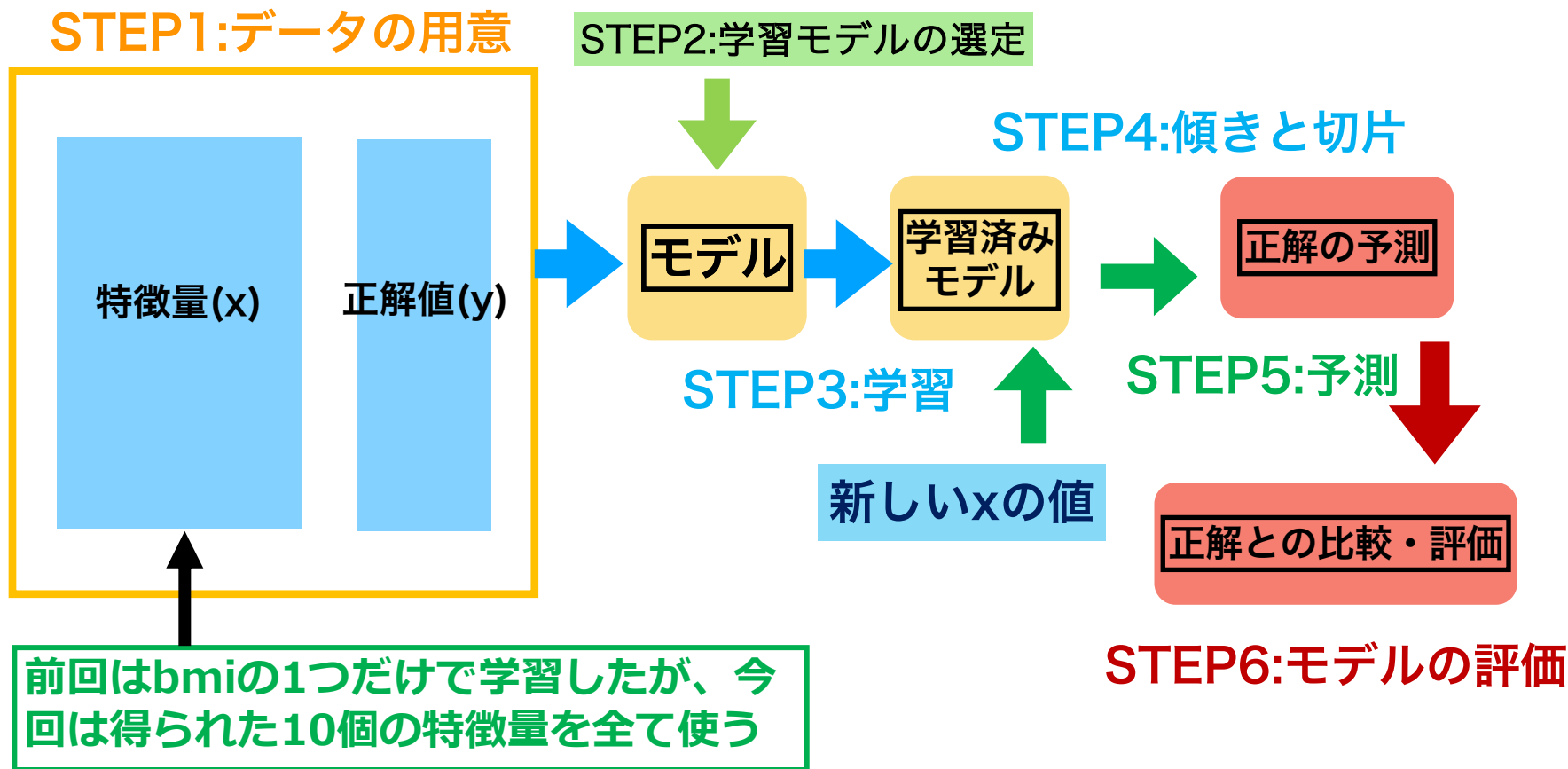


新しいxの値



STEP6:モデルの評価

機械学習の流れのまとめ



重回帰分析

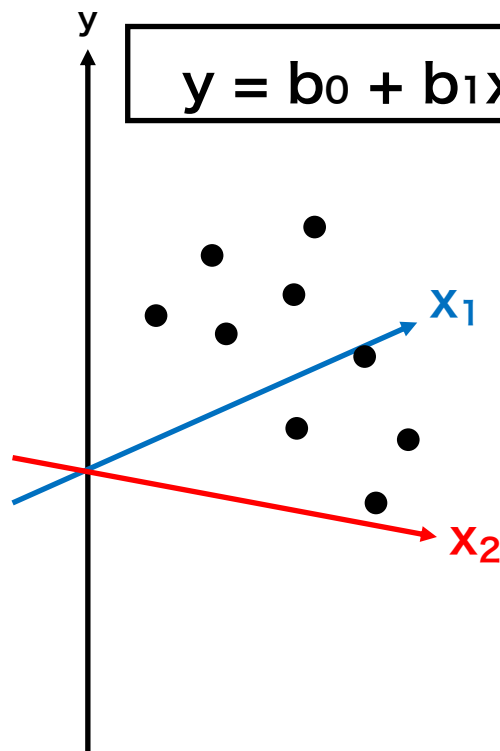
単回帰に対して、複数の変数で回帰することを**多変量回帰**や**重回帰**と呼ぶ

重回帰式

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_6 + b_7x_7 + b_8x_8 + b_9x_9 + b_{10}x_{10}$$

(y : 予測値、 x_n : 特徴量、 b_0 : 切片、 b_n : 傾き)

重回帰分析



複数のxがあるときは多次元になる

10個の変数があるときは10次元のデータになる

線形回帰分析：複数の特徴量で予測

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-11 新しくmodel_dm10インスタンスを作成し、
10個の特徴量(x_dm)でy_dmを予測する

```
model_dm10 = LinearRegression()  
model_dm10.fit(x_dm, y_dm)
```

```
model_bmi.fit(x_bmi, y_dm)
```

x_dmの内容（再掲）

	年齢	性別	BMI	平均血圧	TC (総コレステ ロール値)	LDL コレステ ロール	HDL コレステ ロール	TCH (TC/HDL)	LTG (血液中の 中性脂肪値 の対数)	GLU (血糖値)
	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
count	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000
mean	48.518100	1.468326	26.375792	94.647014	189.140271	115.439140	49.788462	4.070249	4.641411	91.260181
std	13.109028	0.499561	4.418122	13.831283	34.608052	30.413081	12.934202	1.290450	0.522391	11.496335
min	19.000000	1.000000	18.000000	62.000000	97.000000	41.600000	22.000000	2.000000	3.258100	58.000000
25%	38.250000	1.000000	23.200000	84.000000	164.250000	96.050000	40.250000	3.000000	4.276700	83.250000
50%	50.000000	1.000000	25.700000	93.000000	186.000000	113.000000	48.000000	4.000000	4.620050	91.000000
75%	59.000000	2.000000	29.275000	105.000000	209.750000	134.500000	57.750000	5.000000	4.997200	98.000000
max	79.000000	2.000000	42.200000	133.000000	301.000000	242.400000	99.000000	9.090000	6.107000	124.000000

colab

線形回帰分析：複数の特徴量で予測

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-12 model_dm10の傾きと切片を求める

```
print(model_dm10.coef_)  
print(model_dm10.intercept_)
```

e+00 : $\times 10^0$
e+01 : $\times 10^1$
e-01 : $\times 10^{-1}$

➡ [-3.63612242e-02 -2.28596481e+01 5.60296209e+00 1.11680799e+00
-1.08999633e+00 7.46450456e-01 3.72004715e-01 6.53383194e+00
6.84831250e+01 2.80116989e-01]
-334.5671385187877

傾きは特徴量ごとに出るので、10個出力

➡ 回帰式に当てはめる

$$y = -334.57 + (-0.036 \times \text{Age}) + (-22.85 \times \text{Sex}) + (5.60 \times \text{BMI}) + (1.12 \times \text{BP}) + (-1.09 \times \text{TC}) + (0.75 \times \text{LDL}) + (0.37 \times \text{HDL}) + (6.53 \times \text{TCH}) + (68.5 \times \text{LTG}) + (0.28 \times \text{GLU})$$

線形回帰分析：複数の特徴量で予測

STEP0：事前準備
STEP1：データの用意
STEP2：学習モデルの選択
STEP3：データを入れて学習させる
STEP4：傾きと切片を求める
STEP5：予測を行う
STEP6：モデルの評価

コード9-13 複数の特徴量を入れたモデルの評価を行う

```
print(model_dm10.score(x_dm, y_dm))
```

➡ 0.5177484222203498

(再掲) `print(model_bmi.score(x_bmi, y_dm))`
→ 0.3439237602253802

今回は一つの特徴量(bmi)より10個の特徴量を入れて予測する方が予測能が向上した（1に近いほど予測能が高いと言える）

➡ 特徴量が多いほどyを予測できる性能が高かった
一方、多ければ多いほど必ず予測性能が高くなるとも限らない

➡ 機械学習では予測性能を高めることが非常に重要な目的

演習9：課題

Webclassで課題を提出してください。締め切りは**2024/01/31 23:59**まで

- (1)dmデータの特徴量データからage, sex, bmi, s6(血糖値)の4つを抽出してx_dm4を作成してください(コードを記載してください)
- (2)(1)の特徴量を使用して、線形回帰モデルをmodel_dm4と名前をつけて作成し、重回帰を行ってください。model_dm4の切片を求めて記載してください。 (出力結果)
- (3)(2)の決定係数 R^2 を回答してください。 (出力結果)