

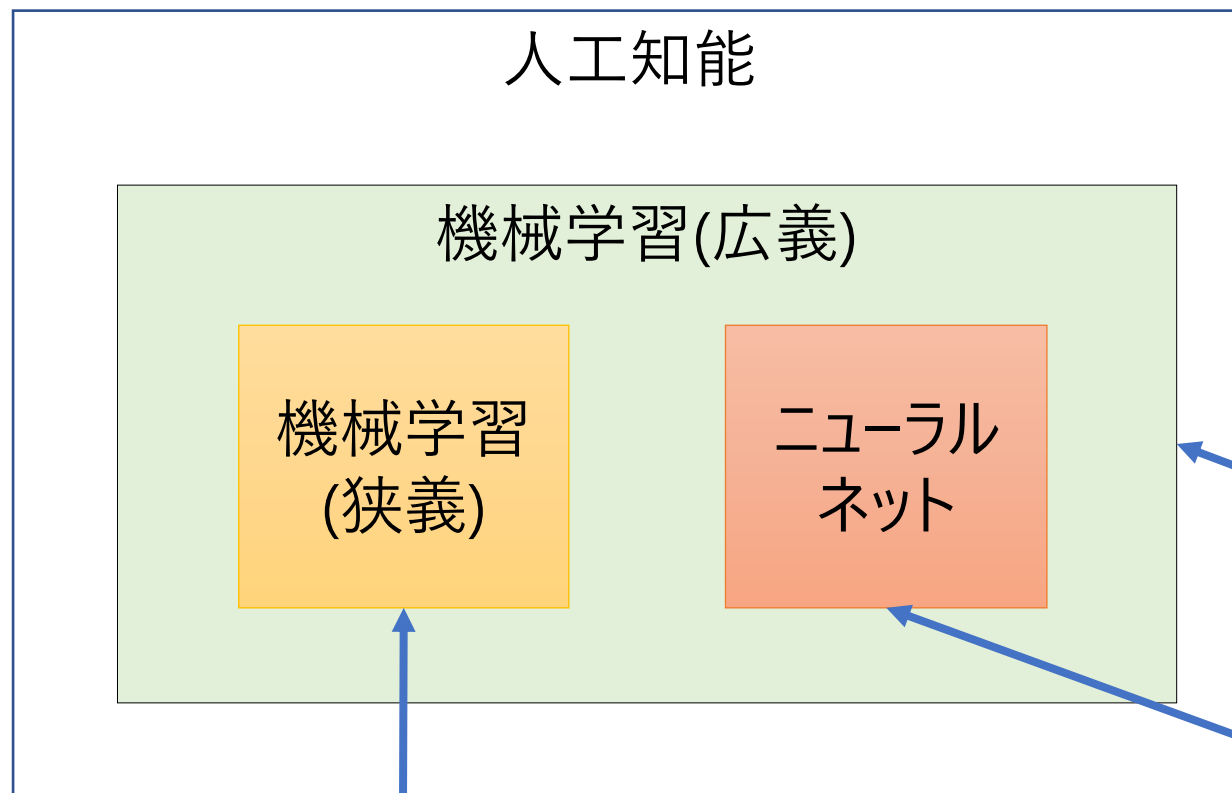
自然言語処理 —機械学習—

<https://yo-sato.com/>

機械学習（100本ノック第5章）の位置づけ

- 狭義の機械学習よりもニューラルネットが、より広範囲の問題に適用でき、性能もうわまっているようなので、手法のいろいろに潜るよりも、ニューラルネットと共通（広義の機械学習）の考え方の部分を学習します。
- 100本ノックの課題集の第5章では、具体的な手法例としてロジスチック回帰を取り上げ、それを使って、トレーニングと評価データを分けること、評価指標としての混同行列、正則化を取り上げています。これらの概念は、ニューラルネットでも共通です。なお、第6章の単語ベクトルの課題集の中で、機械学習ネタとして、クラスタリング手法 2 つと次元圧縮手法一つが取り上げられています。
- 機械学習は、個々のテクは、パッケージですでにサポートされていることが多いので、Pythonレベルでは、Logicを組むというよりも、メソッドを選び引数を塩梅するだけです。そこで、どういう手法（メソッド）が、どういうケースに利用できるのか、を具体的な適用例を通して理解することが重要です。

人工知能、機械学習、ニューラルネット



- 人の知能を模倣する。
- 評価手法

- データでトレーニングして予測する
- データから特徴を抽出して予測に使う
- ロス関数を使って最適化する。そのときに過学習を抑える正則化を行う。

- 推論根拠となる入力特徴を人が定義する
- 推論根拠がわかりやすい

- 推論根拠となる入力特徴を機械が学習する（複雑な対応関係でもOK）
- 推論根拠がわかりにくい

Package使い

- Pythonの基本的なところをやっているときは、ロジックを読む・書くという作業でした。
- ところで、Pythonは、Packageが豊富にそろっているという特徴があります。やりたいことがあったとき、クラスやメソッドとしてアルゴリズムは、すでに、たいてい、用意されているのです。
- そういう場合、コーディング上は、メソッド呼び出し一行です。
- そこで、その一行がだいたい何をやっているかのイメージを持っていないと、サンプルコードが読めません。

概念的な習得が大事

- そのため、このあたりから、ロジックというより、知識ないし概念的な理解が重要になります。
- 以下のようなことを把握していることが必要になります。
 - どのクラスのどのメソッドを、どういうときに使うか
 - メソッドに渡し受け取る引数のデータはどういう意味、構造をもつか
- 概念的な理解は、最初は難しいです。忍耐強く取り組んでください。
 - 1回目は、チンプンカンプン。
 - 2回目は、わかることがでてくる。
 - 3回目は、わかることが増えて、何がわからないかがわかる。-> ここまでくれば後は早いです。

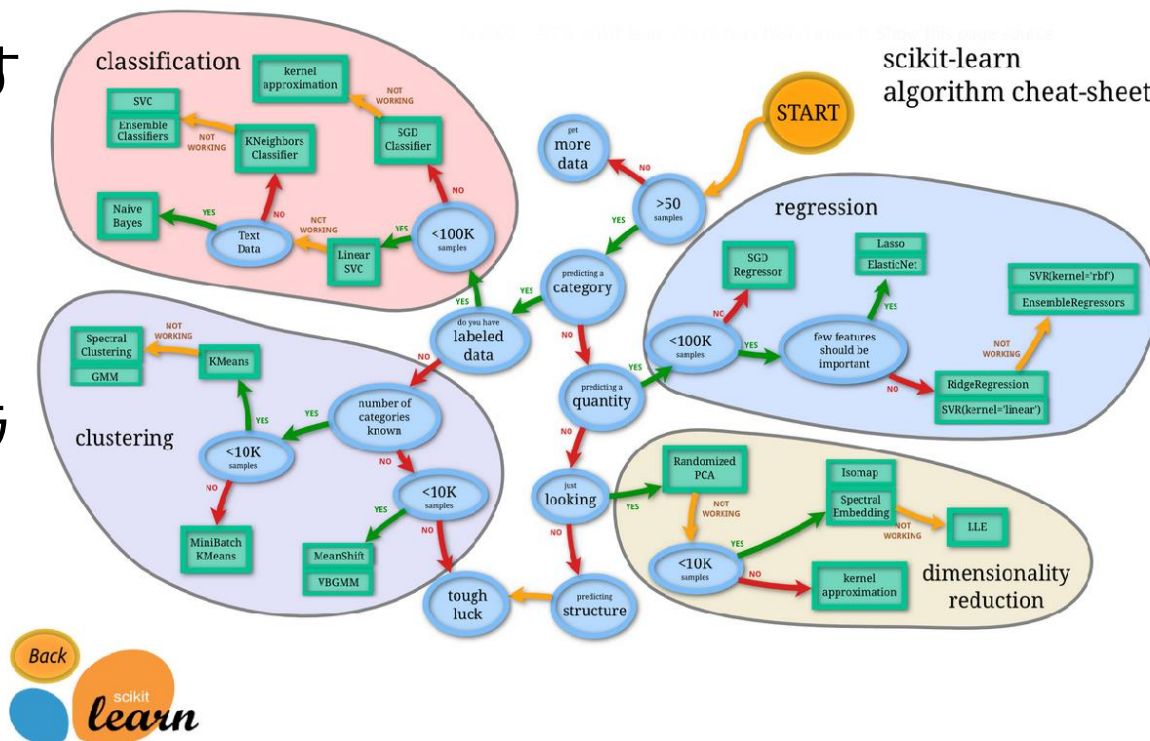
自分で書くとき

- アルゴリズムが1行で済む場合、プログラムを自分で書き起こすときは、前後の引数のデータ加工の部分で、ロジックを書く時間が必要となります。
- そこで、リスト処理、Numpy、Pandasなどが活躍します。

Python Package
scikit-learn (サイキット・ラーン)

分類（クラス分け）：
データを複数のクラス
（グループ）に分類すること

自動グルーピング（クラ
スタリング）：自動的
に分類すること



回帰分析：連続尺度
の従属変数（目的変
数）と独立変数（説
明変数）の間に予測
モデルを作ること

次元圧縮
（Dimensionality
Reduction：次元を
下げて圧縮すること。

https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html#ml-map

以下を眺めてください。Google Chromeの動的な翻訳で、なかなかいい日本語で読めます。

<https://scikit-learn.org/stable/>

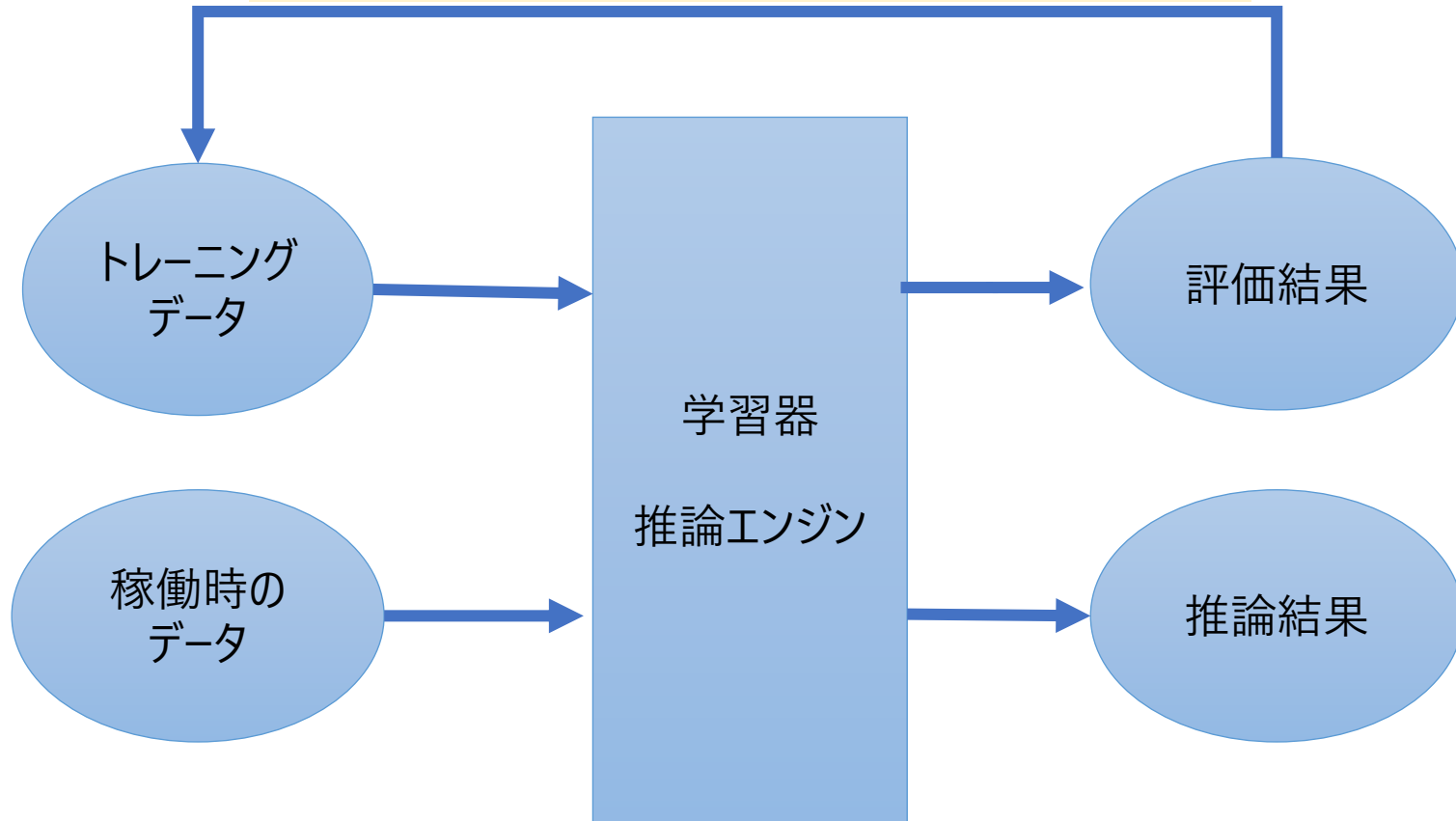
<https://scikit-learn.org/stable/modules/classes.html#>

トレーニングするときのデータの使い
分け：

train, valid, test の役割

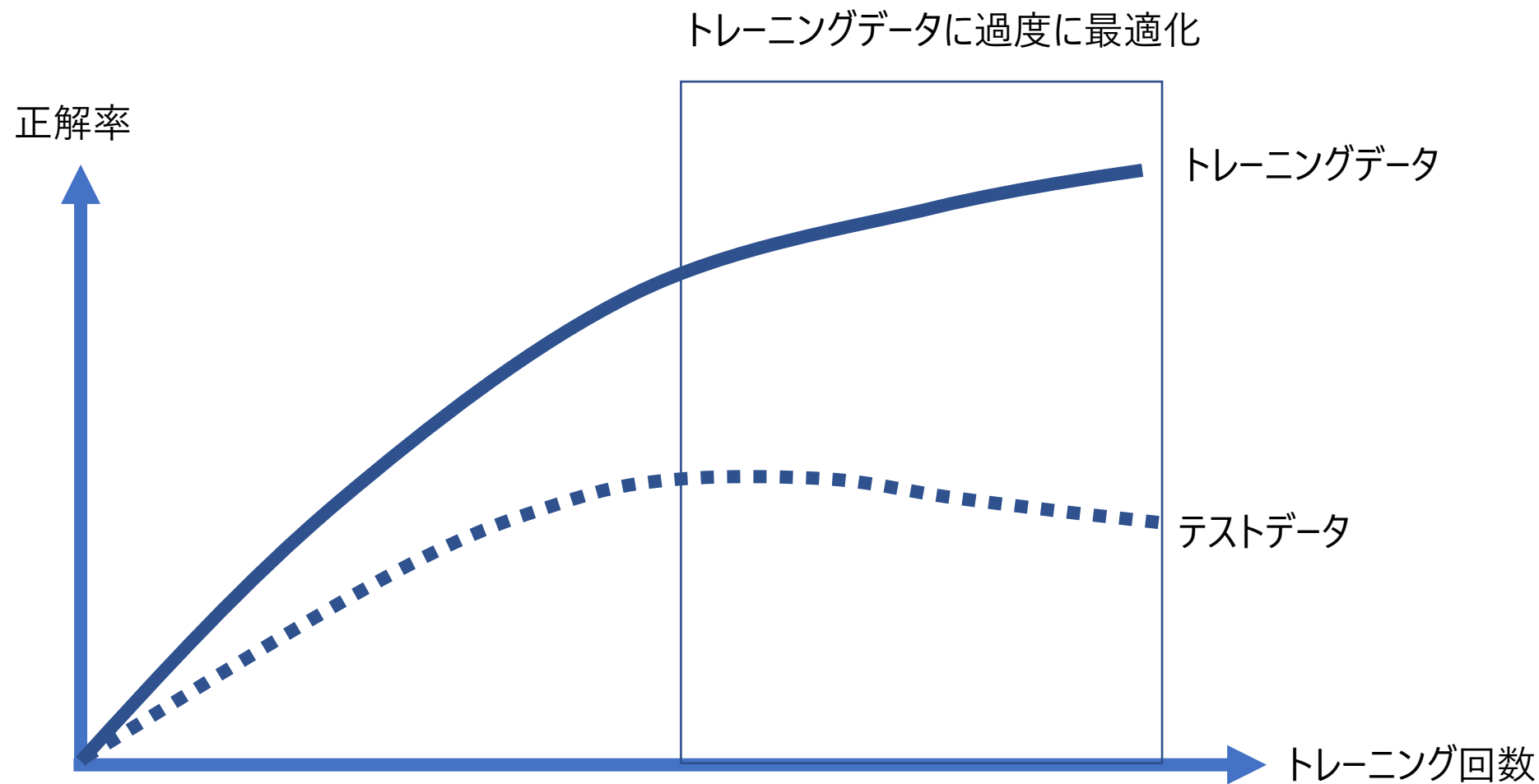
過学習

いいぞ！ （しかし実はトレーニングデータに対して過度にフィットしすぎかも）



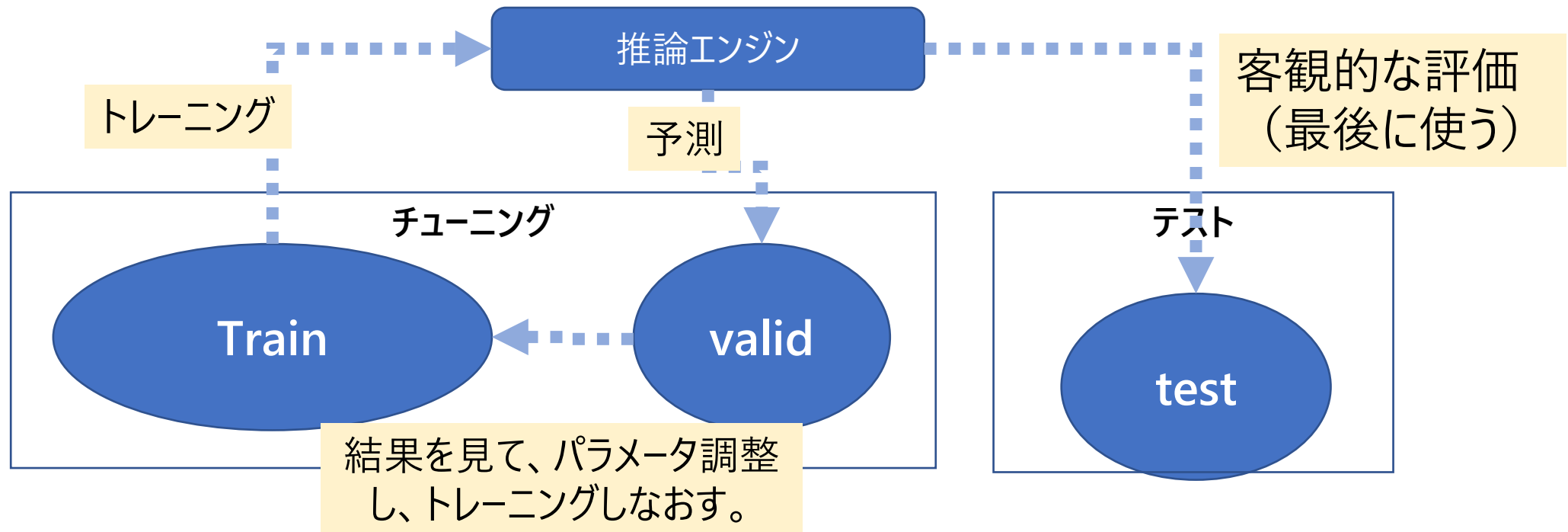
あれ？ 使いものにならない

過学習が起きている



train, valid, test の役割

- Trainデータの結果だけを見ていると、**過学習**しているかどうかわからないので、trainとは別のデータvalidの結果を使ってチューニングする。
- チューニングに使ったデータ(train/valid)で評価しても客観的にならないため、最終的にはチューニングと無関係のtestデータで客観的な評価をする。




文、文書の素朴な表現

頻度ベクトル

I like a dog, and she likes a dog.

世の中に6個の単語しかないとする。
各単語をベクトルのある位置に対応させ、
単語の出現回数をその位置の値とする。

この文のベクトル表現



i	like(s)	a	dog	and	she
1	2	2	2	1	1

TF-IDF (Term Frequency, Inverse Document Frequency)

- 頻度だと重要でない高頻度語（「である」など）がノイズになるので、ある単語がある文書に登場したことの重要度を測る

TF・IDF

= 文書内出現度・文書に登場する珍しさ

$$= \frac{\text{文書内単語出現頻度}}{\text{文書内全単語出現数}} \cdot \log \left(\frac{\text{総文書数}}{\text{単語が出現する文書数}} \right)$$

↑
全文書に登場すると $\log(1)=0$

TF-IDFベクトル

I like a dog, and she likes a dog.

i	like(s)	a	dog	And	she
0.4	0.6	0.1	0.9	0.2	0.4

TF-IDFベクトルは、頻度ベクトルの頻度の代わりにTF-IDF（重要度）をいれたもの

ロジスティック回帰

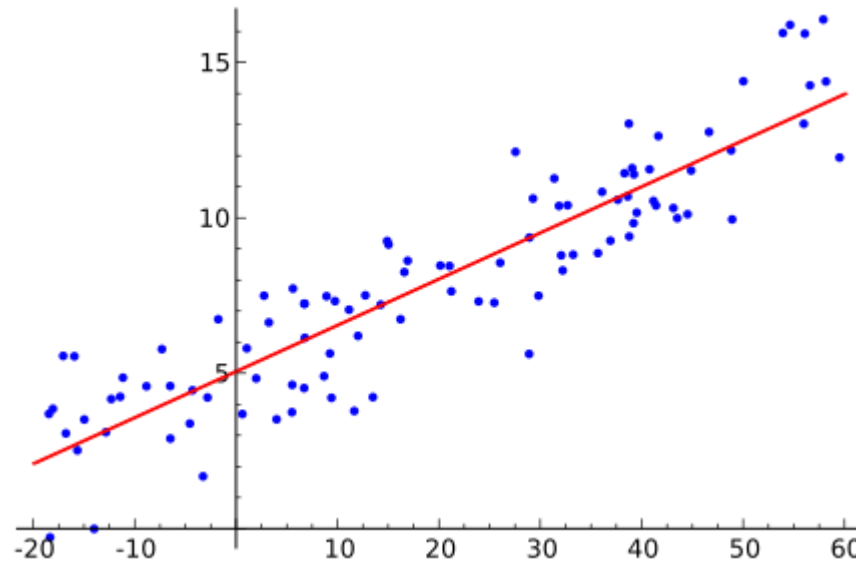
ロジスティック解説動画



目的変数と説明変数

目的変数 y : ...ために、結果がこうなった

従属変数 y : 結果は原因に従属して決まる



モデル $y = ax + b$

説明変数 x : x がこれであるために...

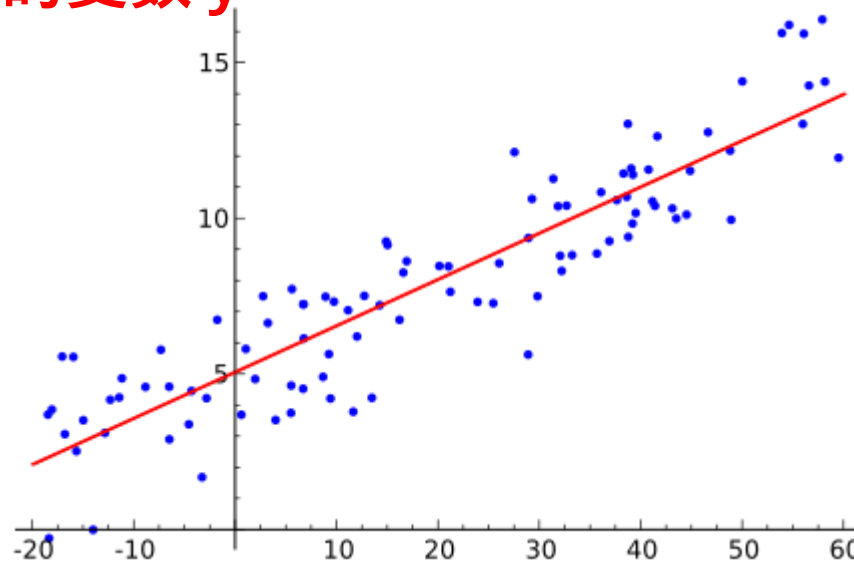
独立変数 x : 原因 x は任意の値をとりうる



線形回帰モデル

モデル $y = ax + b$ (線形式)

目的変数 y



モデルパラメータ a, b を適切に選ぶことで、目的変数 y が説明変数 x から、よく予測できる。

説明変数 x



ロジット関数

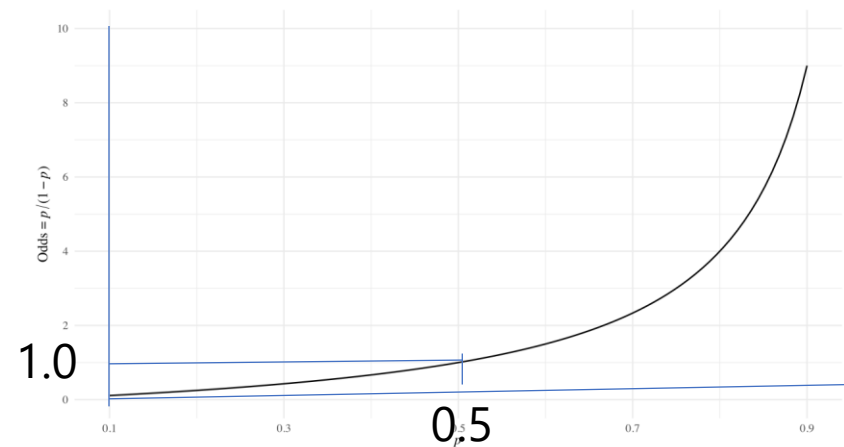
- ロジット：オッズ $p/(1-p)$ の対数。

$$\text{logit}(p) = \ln \left(\frac{p}{1-p} \right)$$

\ln はeを底とする対数

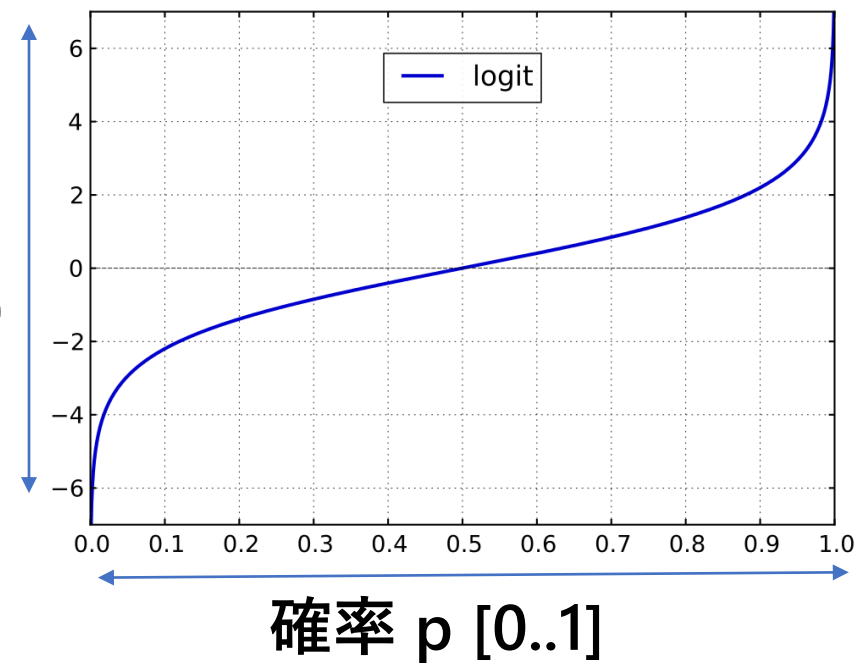
- 実数 $\text{logit}(p)$ と確率 p を対応付ける。

$$\frac{p}{1-p}$$

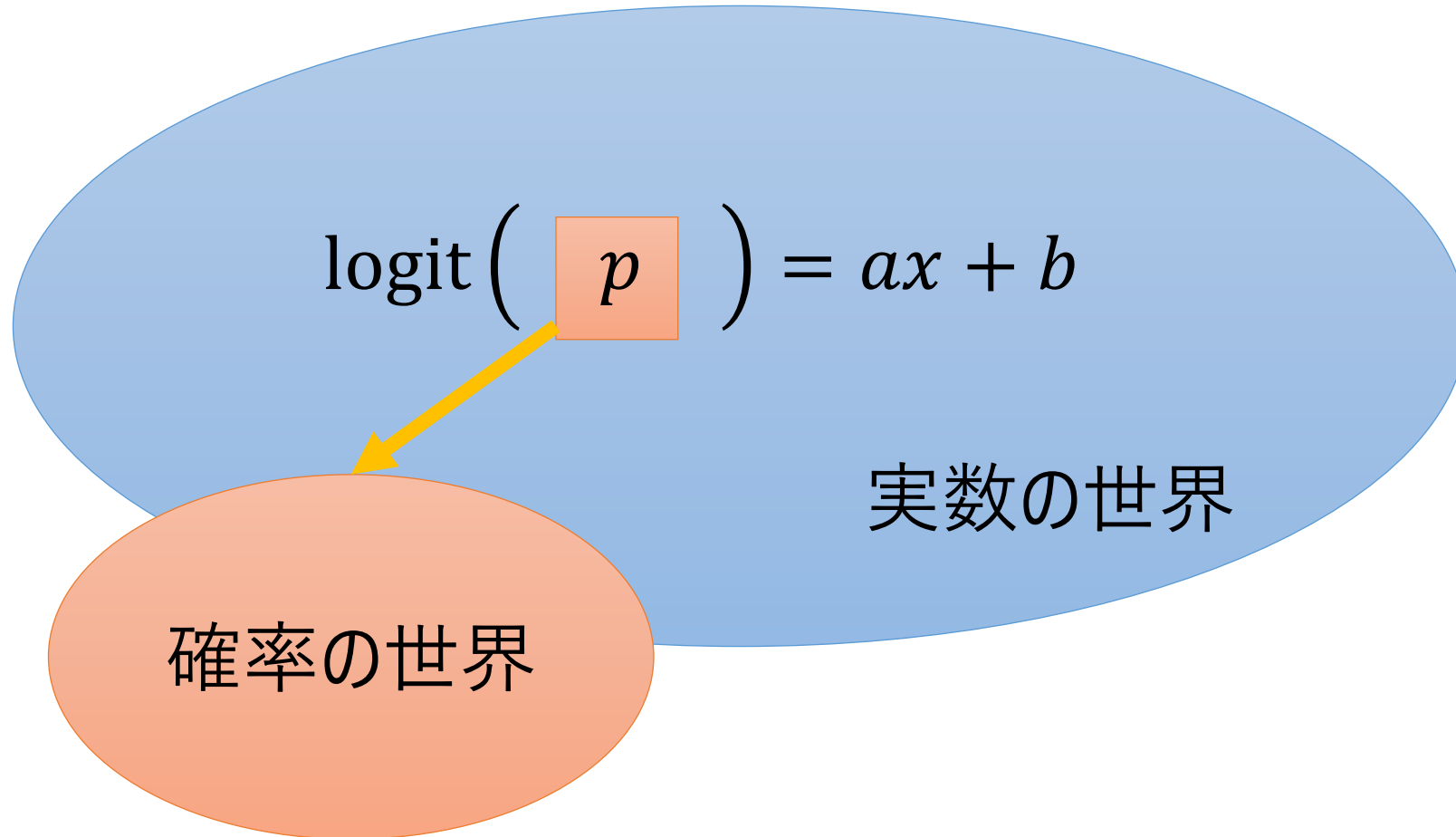


確率 p

実数
 $\text{logit}(p)$
 $[-\infty, \infty]$



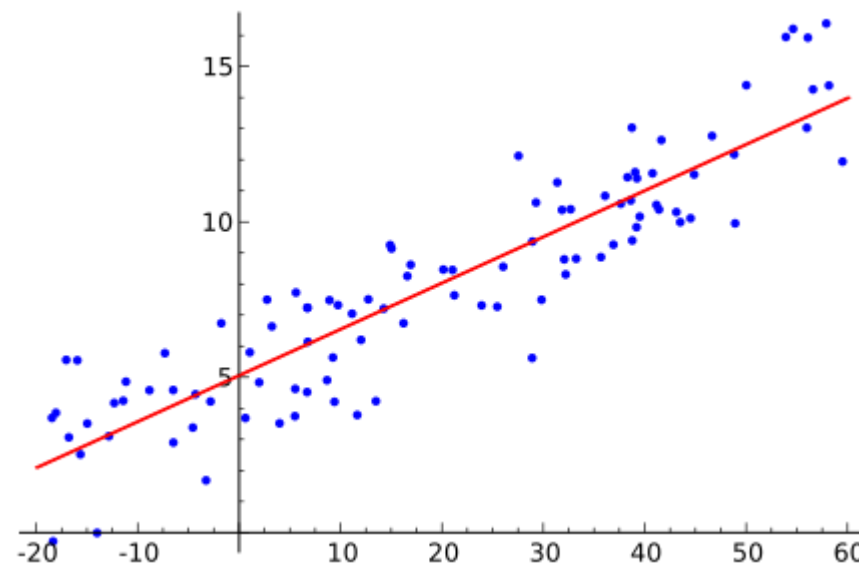
連結関数：異なる世界をつなぐ



予測における確率の効用



確率の世界でできることは、
YESかNOかでYESである確率が0.6、
写真のオブジェクトが犬である確率が0.6、
などクラス分類問題。



実数の世界でできることは、
ある値の予測や再現（回帰）



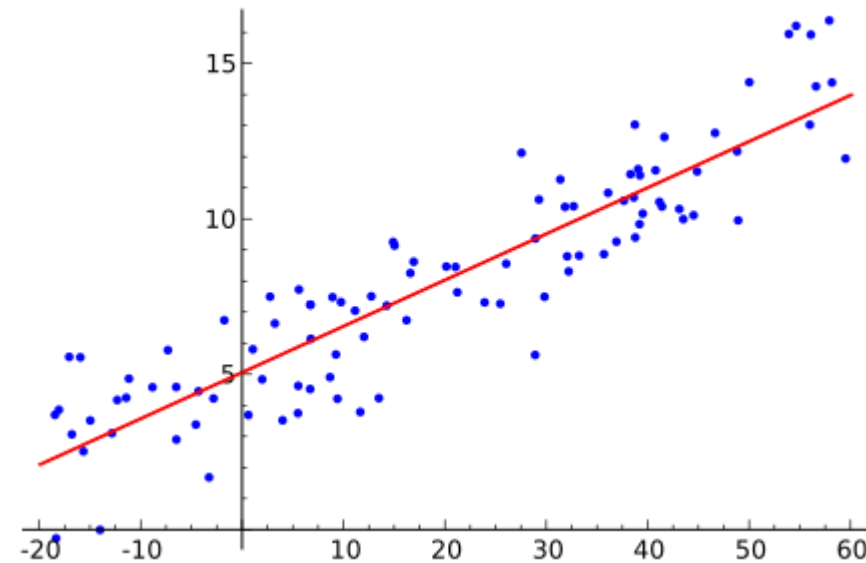
連結関数

目的変数確率 p
を使ってクラス分類

連結
関数

実数の世界で説明変数 x から
の対応関係をモデル化
($y=ax+b$ のパラメータ a, b 決め)

0  1



ロジスティック回帰

連結関数としてロジットを使用する一般化線形モデル (GLM) の一種

左辺：目的変数 p にロジットという皮をかぶせ
(右辺のモデルと左辺を関数で連結させ) ることで、実数を確率値 p と関連づける。-> p を利用することで2値問題や、多クラス分類に利用できる。

$$\text{logit}(p) = \ln \left(\frac{p}{1-p} \right) = \underline{ax + b}$$

線形モデル

右辺：線形モデルの値をそのまま目的変数にすれば、 $-\infty$ から $+\infty$ 。
->実数値を予測するエンジン（線形回帰）としてしか使えない。



対数と指数の関係を使って目的変数 p の式を得る

$$e^y = L \leftrightarrow y = \log_e L$$

logit連結関数

$$\log_e \left(\frac{p}{1-p} \right) = ax + b$$

式変換

$$p = \frac{1}{1 + e^{-(ax+b)}}$$

確率

p を予測するための
実数の世界でのモデル

説明変数 x がある値をとったときに、目的変数 p （あるクラスに属している確率）が得られる。



ロジスティック回帰の実装

目的変数 p (クラスに属する確率)

説明変数 x (特徴ベクトル)

i	like(s)	a	dog	And	she
0.4	0.6	0.1	0.9	0.2	0.4

$$p = \frac{1}{1 + e^{-(ax+b)}}$$

Sigmoid関数

説明変数 x (特徴ベクトル) がある値をとったときに、目的変数 p (あるクラスに属している確率) が得られる。

ある特徴ベクトルのときの正解クラス分類を教師データとして、 p が1になるように、パラメータ(バイアス b 、重み a)をトレーニングする。



ロジスティック回帰

- 連結関数としてロジットを使用する一般化線形モデル (GLM) の一種。

式変換

$$\text{logit}(p_i) = \ln \left(\frac{p_i}{1 - p_i} \right) = \alpha + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_k x_{k,i}$$

説明変数の特徴ベクトル

モデルのパラメータ (トレーニング対象)

$$p_i = \frac{1}{1 + e^{-(\alpha + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_k x_{k,i})}}$$

<https://ja.wikipedia.org/wiki/%E3%83%AD%E3%82%B8%E3%82%B9%E3%83%86%E3%82%A3%E3%83%83%E3%82%AF%E5%9B%9E%E5%B8%B0>



評価指標：混同行列（Confusion Matrix）

[ConfusionMatrix解説動画](#)

評価指標：Confusion Matrix

	予測結果0 (陰)	予測結果 1 (陽)
実際の正解 0 (陰)	True Negative 正しく、陰とした。(真陰性)	False Positive 間違えて、陽とした。(偽陽性)
実際の正解 1 (陽)	False Negative 間違えて、陰とした。(偽陰性)	True Positive 正しく、陽とした。(真陽性)



評価指標：正解率(Accuracy)

- 正解率

- 分類したデータの総数のうち、正しく分類されたデータ数の割合
- $(TP+TN)/(TP+FN+FP+TN)$
 - 右図、太枠が分母で、色付きセルが分子
 - 0, 1 のいずれをTrue、Falseとすることによって、値が異なることに留意。

	予測結果 0	予測結果 1
実際の正解 0	True Negative	False Positive
実際の正解 1	False Negative	True Positive



評価指標：適合率(Precision)

- 適合率

- クラス1に分類されたデータのうち、実際にクラス1であるデータ数の割合
- $TP / (TP + FP)$

	予測結果 0	予測結果 1
実際の正解 0	True Negative	False Positive
実際の正解 1	False Negative	True Positive



評価指標：再現率(Recall)

- 再現率

- 実際にクラス1であるデータのうち、クラス1に分類されたデータ数の割合
- $TP / (TP + FN)$

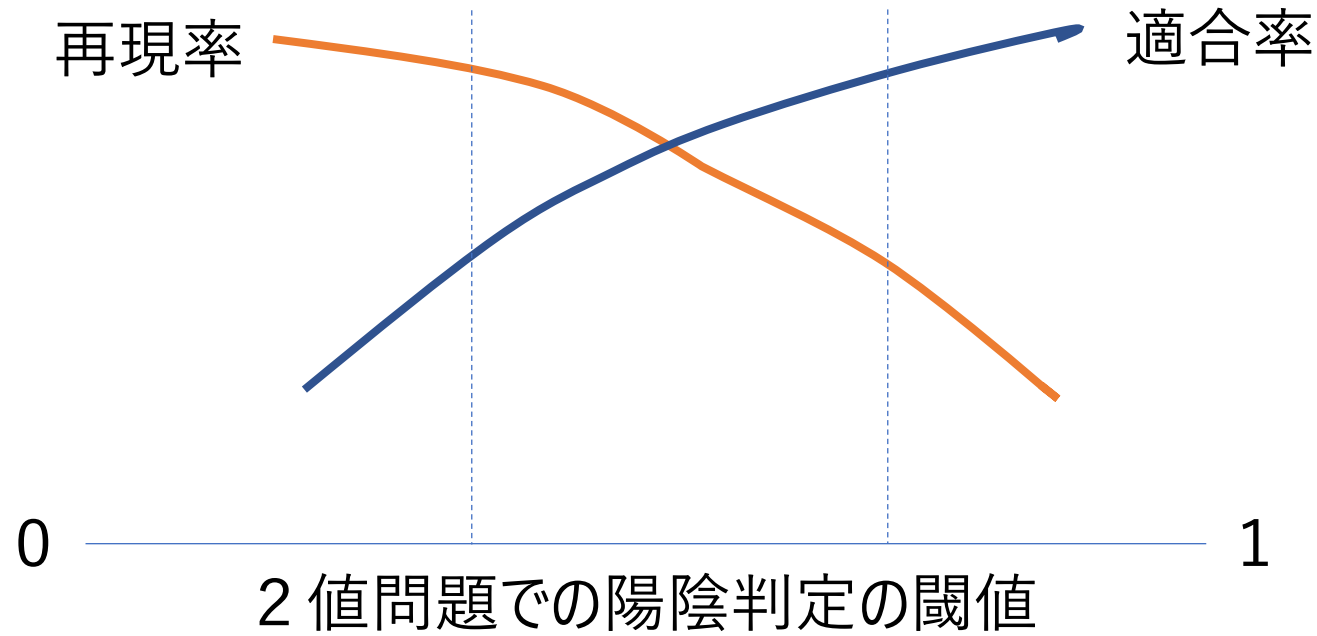
	予測結果 0	予測結果 1
実際の正解 0	True Negative	False Positive
実際の正解 1	False Negative	True Positive



適合率と再現率はトレードオフ

確率0.4以上も積極的に陽性と判定するとすると、適合率は下がるが、再現率は上がる。

安全をとって確率0.6以上が陽性と判定するとすると、適合率は上がるが、再現率は下がる



評価指標：F1スコア

$$\frac{2 * \text{適合率} * \text{再現率}}{\text{適合率} + \text{再現率}}$$

適合率と再現率のバランスをとるための指標



正則化

正則化

$$\text{ロス} = \text{ロス関数} + \text{正則化項}$$

学習に味付け済みのロスを
使うことで、過学習しにく
かったり、滑らかなモデルに
なったりする

モデルのパラメータが、
どれだけ望ましいか
(望ましくないか) の
基本尺度。モデルの
推論結果と正解ラベ
ル間の距離とか。

パラメータが極端な値をとったと
きに罰則を与える味付け

L0ノルム：0でないパラメータの数
L1ノルム：パラメータの絶対値の和
L2ノルム：パラメータの二乗和の平方根

100本ノック第6章課題50～56,58

- [「100本ノック」の6章の課題](#)を解いてみましょう。
- 「NLP、機械学習.ipynb」というノートをコピーし、冒頭の準備をやった後、各課題のセクション下のコードセルのコードを完成させ、実行ログを残してください。
- 以下に、上記の予備知識に加えて、課題を解く際に参考となることを説明します。

課題を解くための参考情報

55補足


- `confusion_matrix(y_train, lr.predict(x_train))`
 - 第1引数は正解ラベル、第2引数は予測結果ラベル

予測分類

正解分類

	b	e	m	t
b	4514	0	0	0
e	0	4225	0	0
m	0	0	723	0
t	0	0	0	1210

bをeと間違えた



56補足

- `classification_report(y_test, lr.predict(x_test))`
 - 第1引数は正解ラベル、第2引数は予測結果ラベル

	precision	recall	f1-score	support	
	1.00	1.00	1.00	569	
	1.00	1.00	1.00	563	
	1.00	1.00	1.00	76	
	1.00	1.00	1.00	126	
			1.00	1334	
	1.00	1.00	1.00	1334	
	1.00	1.00	1.00	1334	

test内個数

ラベルごとのPrecision等を
単純に平均したもの

accuracy

macro avg

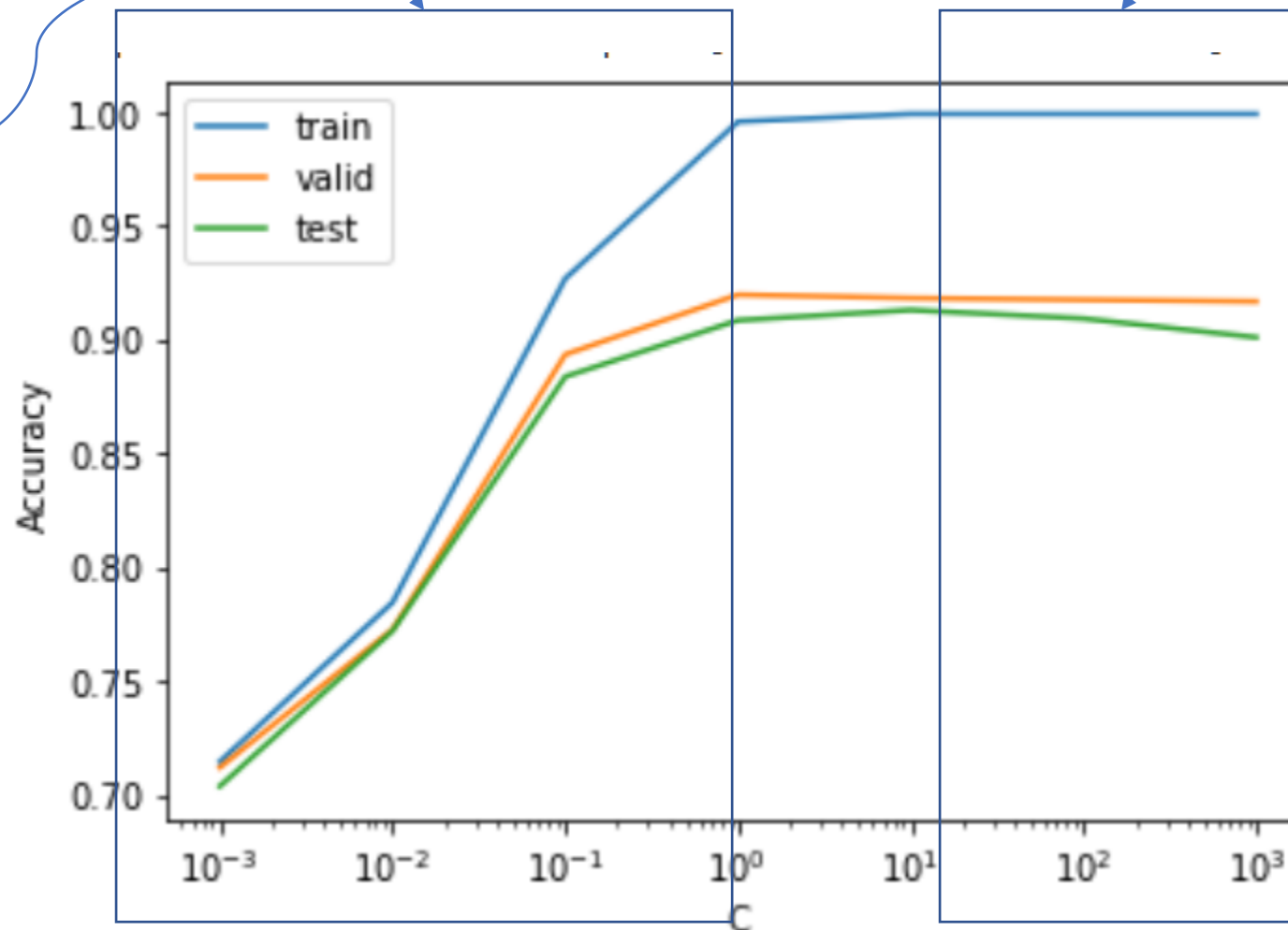
weighted avg

ラベルごとのPrecision等を
supportの数で重みづけして平均したもの

micro avgは、すべてのラベルに関して、
True Positive 等をカウントして、求める

58補足

正則化強すぎると、
うまく学習しない



正則化項が弱いと、
過学習で、trainにfit
過ぎて、客観的には
劣化している

正則化項強い

正則化項弱い

確認クイズ

- スタログの確認クイズをやってください。