Python ではじめる機械学習

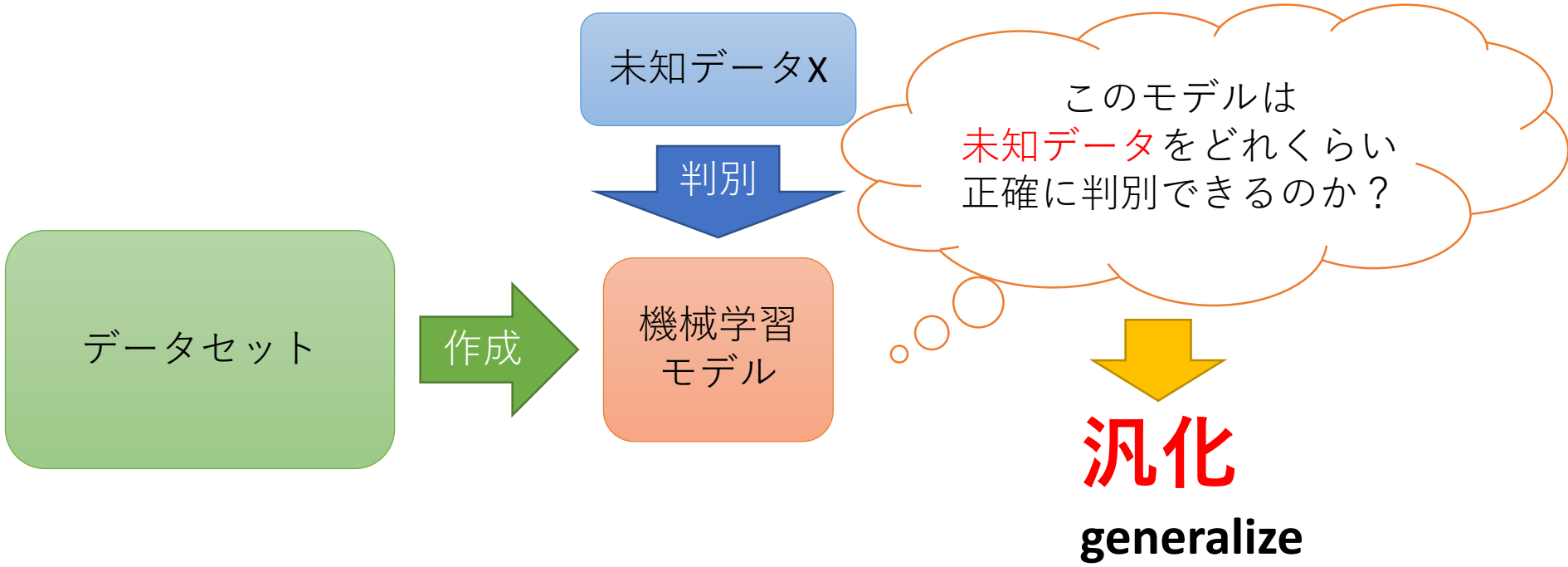
5 モデルの評価と改良

5.1 交差検証

15T4057F　藤井　真

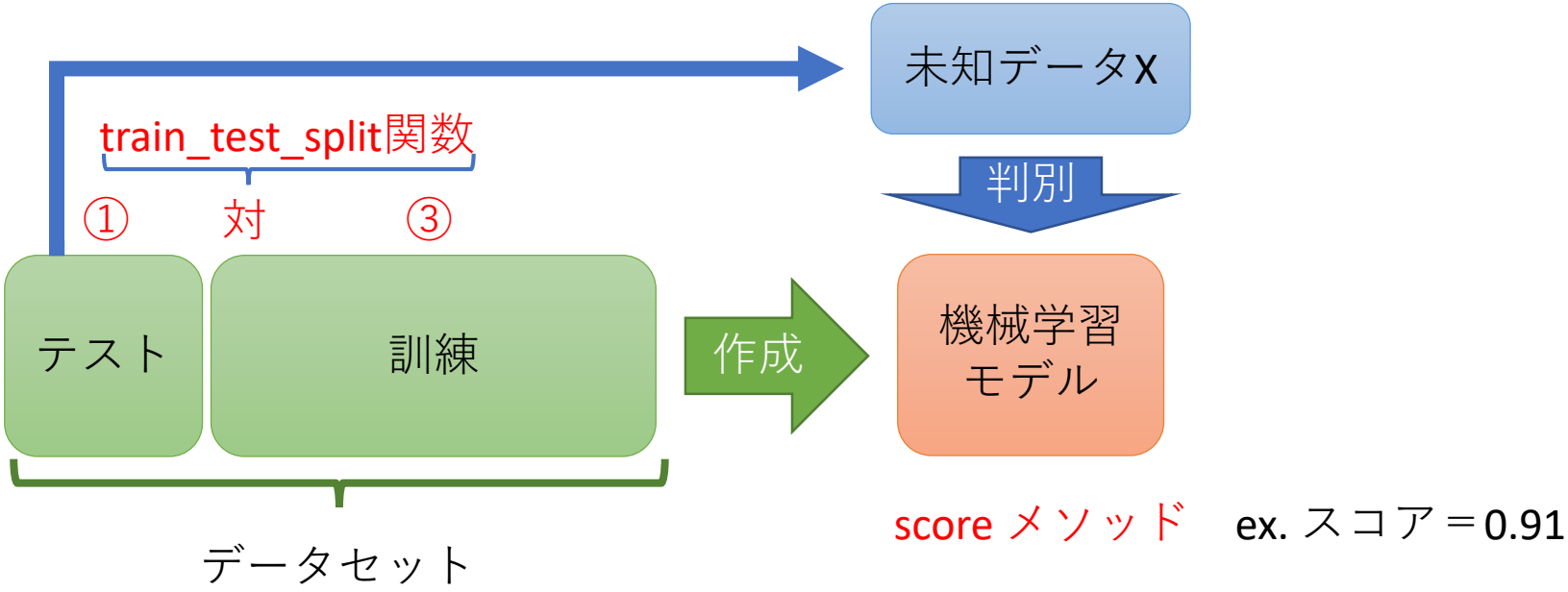
5 モデルの評価と改良

- モデルの評価とは
  「1.7.2 成功度合いの測定」
  「2.2 汎化・過剰適合・適合不足」でも学んだが
  どれだけ未見（未知）のデータに対して正解するか



5 モデルの評価と改良

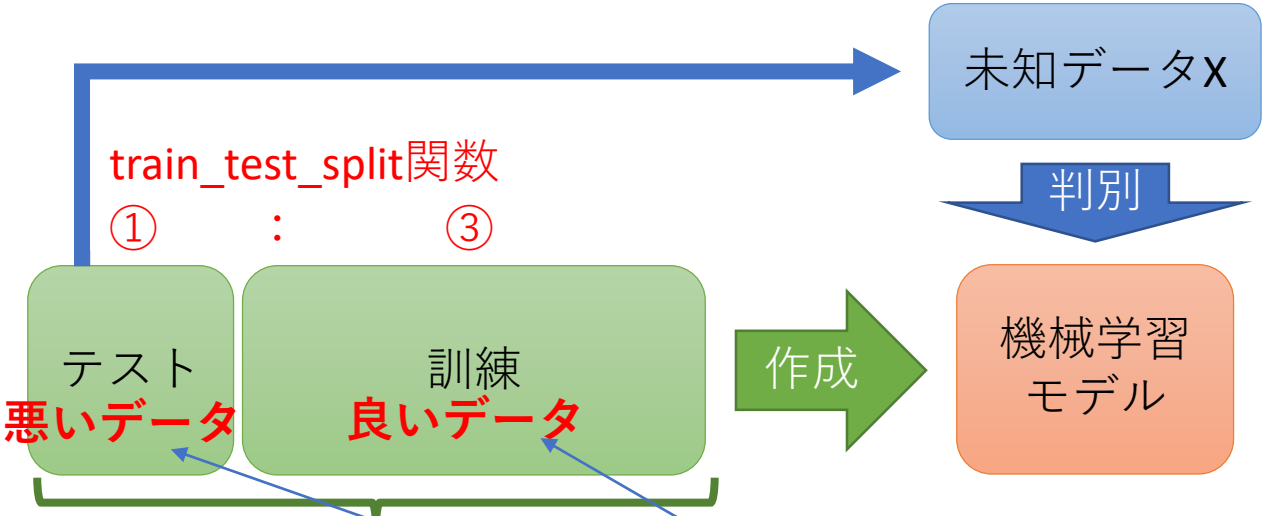
- これまでのモデル評価



⇒train_test_split関数はランダムにデータを分けられる

5 モデルの評価と改良

- これまでのモデル評価



score メソッド　ex. スコア＝0.91

train_test_split関数はランダムにデータを分けられる

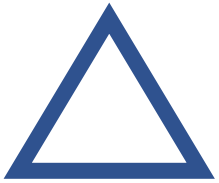
しかし、ランダムにデータを分けた結果が偏っていたら
⇒正しいモデル評価にならない

対策

5.1 交差検証
5.2 グリッドサーチ

5.1 交差検証 (cross-validation)

- モデル評価のためにデータを1回分ける

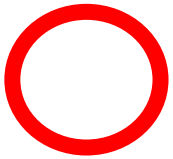


内容の偏りによる
精度のゆれがある

- データをk分割して各々をテストデータとする

1回目	テスト	訓練	訓練	訓練	訓練
2回目	訓練	テスト	訓練	訓練	訓練
3回目	訓練	訓練	テスト	訓練	訓練
4回目	訓練	訓練	訓練	テスト	訓練
5回目	訓練	訓練	訓練	訓練	テスト

5分割交差検証



内容が偏っても
各々の結果が
補正する

5.1 交差検証 (cross-validation)

- 実装 (cross_val_score関数)

```
from sklearn.model_selection import cross_val_score
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression

iris = load_iris()
logreg = LogisticRegression()

scores = cross_val_score(logreg, iris.data, iris.target)
print(scores)
```

```
[ 0.96078431  0.92156863  0.95833333]
```

デフォルトでは3分割交差検証

```
scores = cross_val_score(logreg, iris.data, iris.target, cv=5)
print(scores)
print(scores.mean())
```

cvパラメータを「5」にする

```
[ 1.          0.96666667  0.93333333  0.9         1.        ]
0.96
```

5分割交差検証の平均

5分割交差検証
各々は0.9-1.0と
分割間で差がある

5.1 交差検証 (cross-validation)

- 長所
  - データを効率的に使える。
    - 全てのデータポイントは正確に1度だけテストされる。
    - train_test_splitでは75％が訓練データだが、
      5分割交差検証なら80％が訓練、10分割なら90％が訓練
      で使用でき、モデルが頑強になる。

- 短所
  - 計算コスト
    - k分割した場合、k個のモデルが作られる。
      ⇒単純にk倍遅くなる。

5.1 交差検証 (cross-validation)

- 交差検証の制御（デフォルトから外す方法）
  - 実験結果の再現などのため、
    『あえて』デフォルトから外したいとき
    ⇒cvパラメータに
    交差検証分割器 (cross-validation splitter)を渡す。

    - サンプルとして
      - 3クラス分類のirisデータセット
      - ラベルが[0,0,0,...,0,1,1,1,…,1,2,2,2,…,2,2]
        と並んでいるものを、
        『あえて』シャッフルしないで3分割する。

5.1 交差検証 (cross-validation)

- 交差検証の制御

```
from sklearn.datasets import load_iris
iris = load_iris()
print("Iris labels:\n{}".format(iris.target))
```

0,1,2が並んだまま

```
Iris labels:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

交差検証分割器のインポート

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=3)
scores = cross_val_score(logreg, iris.data, iris.target, cv=kfold)
print(scores)
```

交差検証分割器の生成（シャッフルしない）

cvパラメータに交差検証分割器を渡す

```
[ 0.  0.  0.]
```

```
kfold = KFold(n_splits=3, shuffle=True, random_state=0)
scores = cross_val_score(logreg, iris.data, iris.target, cv=kfold)
print(scores)
```

シャッフルする分割器の生成

```
[ 0.9   0.96  0.96]
```

5.1 交差検証 (cross-validation)

- 交差検証の種類

  本書では以下の4種類

  - 層化k分割交差検証
  - 1つ抜き交差検証
  - シャッフル分割交差検証
  - グループ付き交差検証

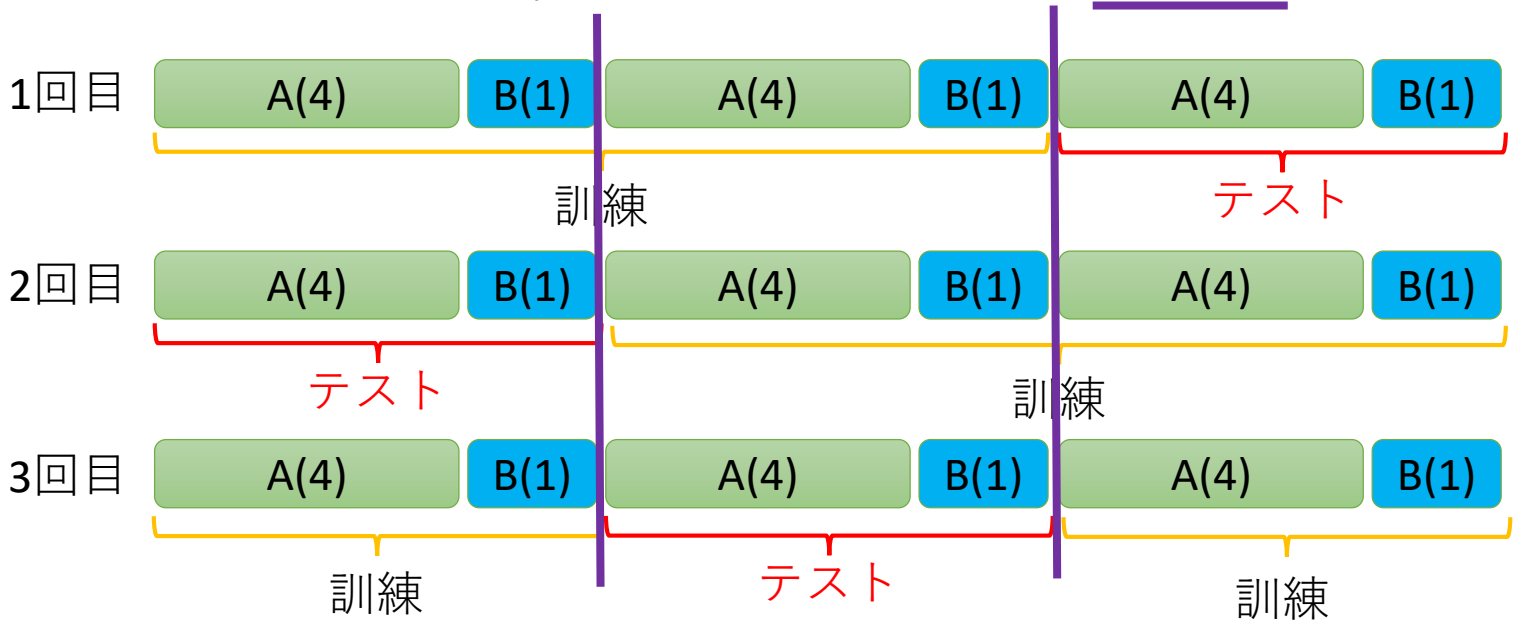
5.1 交差検証 (cross-validation)

層化k分割交差検証
1つ抜き交差検証
シャッフル分割交差検証
グループ付き交差検証

- 層化k分割交差検証
  (stratified k-fold cross-validation)

クラスの偏りを無くすため、最初から分割1単位のクラス割合を**全体の割合と同じにした**もの。

- イメージ（A,Bの割合が4:1の層化3分割交差検証）



5.1 交差検証 (cross-validation)

層化k分割交差検証
1つ抜き交差検証
シャッフル分割交差検証
グループ付き交差検証

- 1つ抜き交差検証
  (leave-one-out)

データ数100のデータならば**100分割交差検証を行う。**
⇒1サンプルだけをテストとし、残り全てで訓練する。
⇒少データで有用、多大データでは終わらない。

- 実装

```
from sklearn.model_selection import LeaveOneOut
loo = LeaveOneOut()
scores = cross_val_score(logreg, iris.data, iris.target, cv=loo)
print("Number of cv iterations: ", len(scores))
print("Mean accuracy: {:.2f}".format(scores.mean()))
```

cvパラメータにLOOインスタンスを渡す

```
Number of cv iterations:  150
Mean accuracy: 0.95
```

150回行われた

150回の平均スコア

5.1 交差検証 (cross-validation)

層化k分割交差検証
1つ抜き交差検証
シャッフル分割交差検証
グループ付き交差検証

- シャッフル分割交差検証
  (shuffle-split cross-validation)

交差検証を行う訓練・テストの割合や個数、
分割数などを細かく設定できる。

- 実装（irisのテストを23個、訓練を70％、18分割）

```
from sklearn.model_selection import ShuffleSplit
shuffle_split = ShuffleSplit(test_size=23, train_size=.7, n_splits=18)
scores = cross_val_score(logreg, iris.data, iris.target, cv=shuffle_split)
print("Cross-validation scores:\n{}".format(scores))
```

```
Cross-validation scores:
[ 0.95652174  0.95652174  1.          1.          0.82608696  0.91304348
  0.95652174  0.91304348  0.95652174  0.95652174  0.82608696  0.95652174
  1.          0.91304348  0.95652174  0.95652174  0.95652174  0.91304348]
```

cvパラメータにインスタンスを渡す

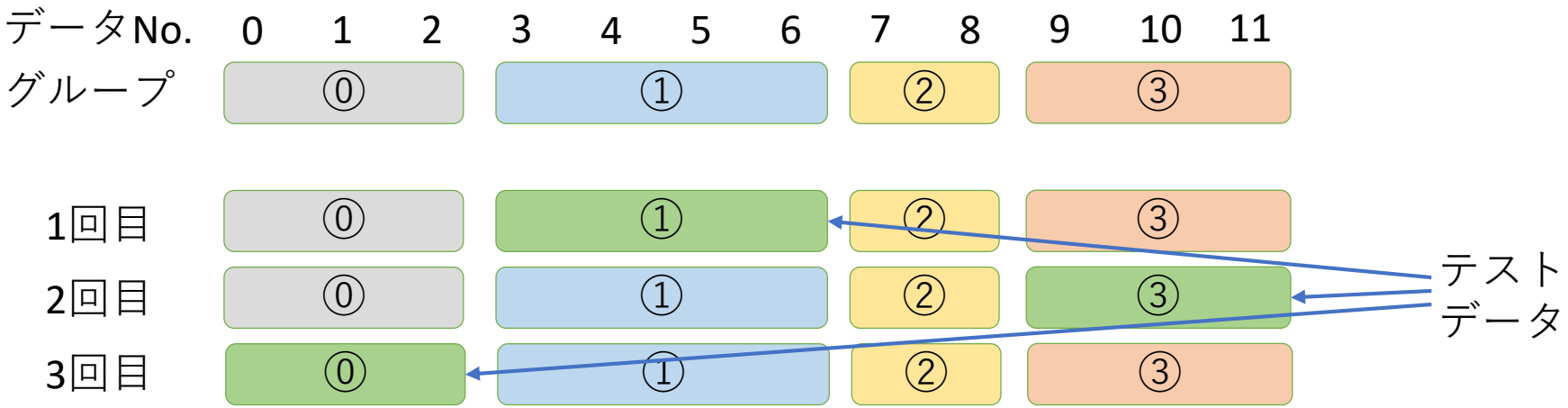
5.1 交差検証 (cross-validation)

層化k分割交差検証
1つ抜き交差検証
シャッフル分割交差検証
グループ付き交差検証

- グループ付き交差検証

データセットの中に密接に関係するグループがある場合に用いられる交差検証の設定。(GroupKFold)

- 12データ、グループ⓪*3, ①*4, ②*2, ③*3



⇒交差検証分割の際、グループ単位で利用される。

5.1 交差検証 (cross-validation)

- グループ付き交差検証

層化k分割交差検証
1つ抜き交差検証
シャッフル分割交差検証
グループ付き交差検証

データセットの中に密接に関係するグループがある場合に用いられる交差検証の設定。(GroupKFold)

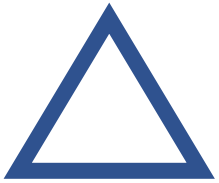
- 実装（12データ、グループ⓪*3, ①*4, ②*2, ③*3）

```
from sklearn.datasets.samples_generator import make_blobs
from sklearn.model_selection import GroupKFold
X, y = make_blobs(n_samples=12, random_state=0)
groups = [0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 3, 3]
scores = cross_val_score(logreg, X, y, groups, cv=GroupKFold(n_splits=3))
print("Cross-validation scores:\n{}".format(scores))
```

```
Cross-validation scores:
[ 0.75        0.8         0.66666667]
```

5.1 交差検証 (cross-validation)

- モデル評価のためにデータを1回分ける

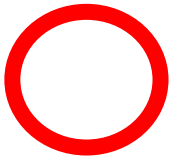


内容の偏りによる
精度のゆれがある

- データをk分割して各々をテストデータとする

1回目	テスト	訓練	訓練	訓練	訓練
2回目	訓練	テスト	訓練	訓練	訓練
3回目	訓練	訓練	テスト	訓練	訓練
4回目	訓練	訓練	訓練	テスト	訓練
5回目	訓練	訓練	訓練	訓練	テスト

5分割交差検証



内容が偏っても
各々の結果が
補正する