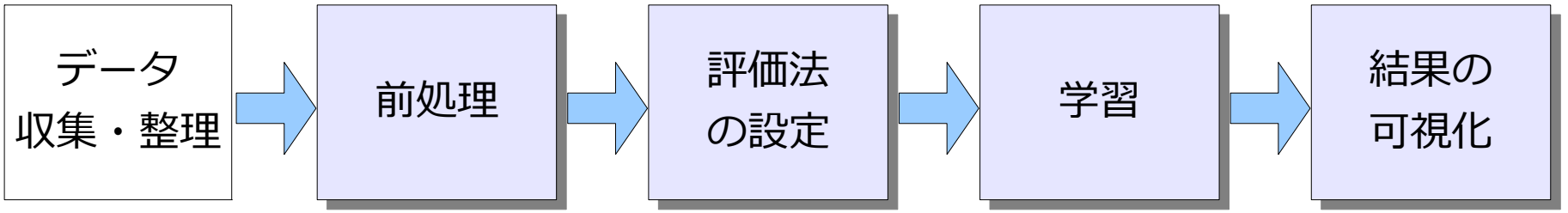
2. 機械学習の基本的な手順



：ツールによる支援が可能

2.1 Weka を用いた機械学習

- Weka とは
  - Waikato Environment for Knowledge Analysis
  - 機械学習のアルゴリズムを実装した Java ライブラリ
  - データファイルを直接操作できる GUI を持つ
  - ライセンスは GNU GPL
    - プログラムの実行・改変・再配布が自由
    - ただし二次的著作物に対しても GNU GPL が適用される

2.1.1 データ収集・整理

- Weka のデータ形式 ARFF フォーマット

```
% 1. Title: Iris Plants Database
@RELATION iris

@ATTRIBUTE sepallength    REAL
@ATTRIBUTE sepalwidth     REAL
@ATTRIBUTE petallength    REAL
@ATTRIBUTE petalwidth     REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1, 3.5, 1.4, 0.2, Iris-setosa
4.9, 3.0, 1.4, 0.2, Iris-setosa
…
7.0, 3.2, 4.7, 1.4, Iris-versicolor
6.4, 3.2, 4.5, 1.5, Iris-versicolor
…
6.3, 3.3, 6.0, 2.5, Iris-virginica
5.8, 2.7, 5.1, 1.9, Iris-virginica
…
```

データセット名

特徴名と型

萼・花びらの長さ・幅

アヤメの種類

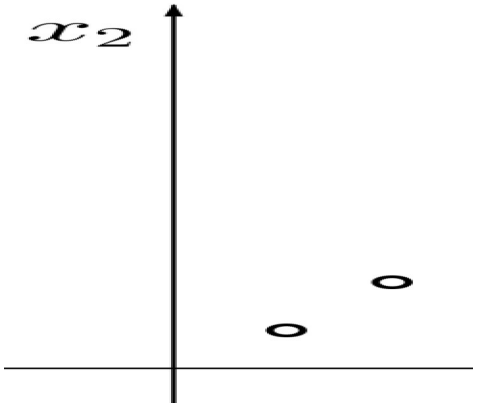
これ以降、1 行に 1 事例
（Excel の CSV 形式と同じ）

2.1.2 前処理

- 分析
  - 主成分分析 (次元削減)
    - データの散らばりをできるだけ保存する低次元空間へ写像
    - データの可視化に有効
- データの標準化
  - すべての次元を平均 0 、分散 1 にそろえる
  - 各次元に対して平均値を引き、標準偏差で割る

2.1.2 前処理

- 主成分分析の考え方

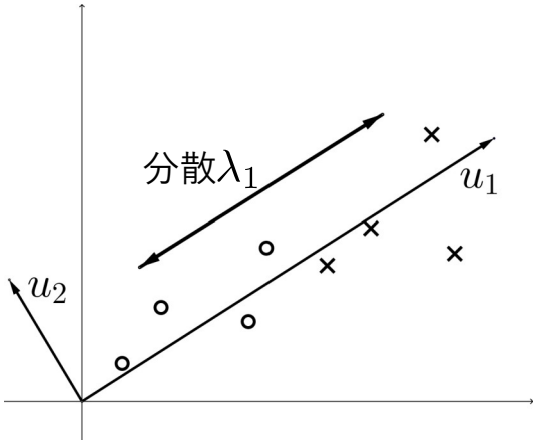


共分散行列Σの計算

$\bar{x_1}, \bar{x_2}$:平均値、N : データ数

対角成分は分散、
非対角成分は相関を表す

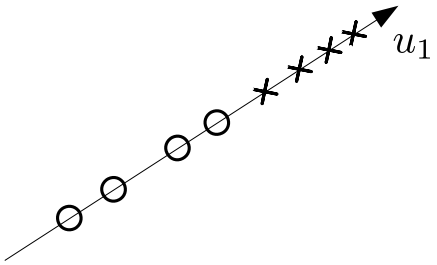
$$\Sigma = \frac{1}{N}\begin{pmatrix} \sum(x_1-\bar{x_1})^2 & \sum(x_1-\bar{x_1})(x_2-\bar{x_2}) \\ \sum(x_1-\bar{x_1})(x_2-\bar{x_2}) & \sum(x_2-\bar{x_2})^2 \end{pmatrix}$$



Σは
　半正定値(→固有値がすべて0以上の実数)
　対称行列(→固有ベクトルが実数かつ直交)
であるので、以下のように分解できる

$$\Sigma' = U^T \Sigma U = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

λは固有値の大きい順、Uは対応する
固有ベクトルU_1, U_2を並べたもの



λ_1に対応する固有ベクトルU_1で
2次元データを1次元に射影

$$u_1 = U_1^T \boldsymbol{x}$$

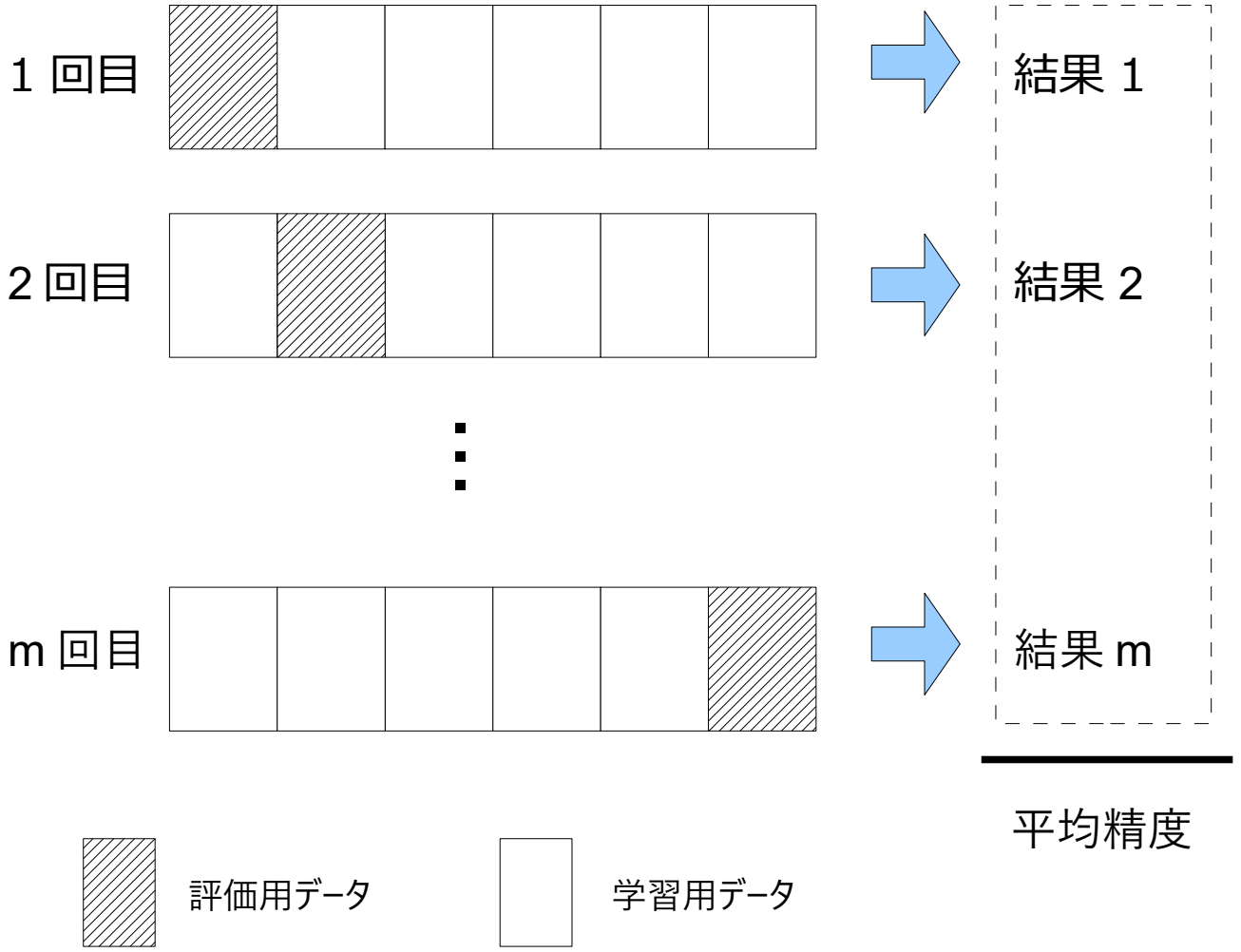
寄与率= $\dfrac{\lambda_1}{\lambda_1+\lambda_2}$

2.1.3 評価基準の設定

- 分割学習法
  - データの半分を学習用、残りの半分を評価用とする
  - ハイパーパラメータを調整する場合は、学習用・検証用・評価用に分ける
- 交差確認法
  - データを m 個の集合に分割し、 m-1 個の集合で学習、残りの 1 個の集合で評価を行う
  - 評価する集合を入れ替え、合計 m 回評価を行う
  - 分割数をデータ数とする場合を一つ抜き法とよぶ

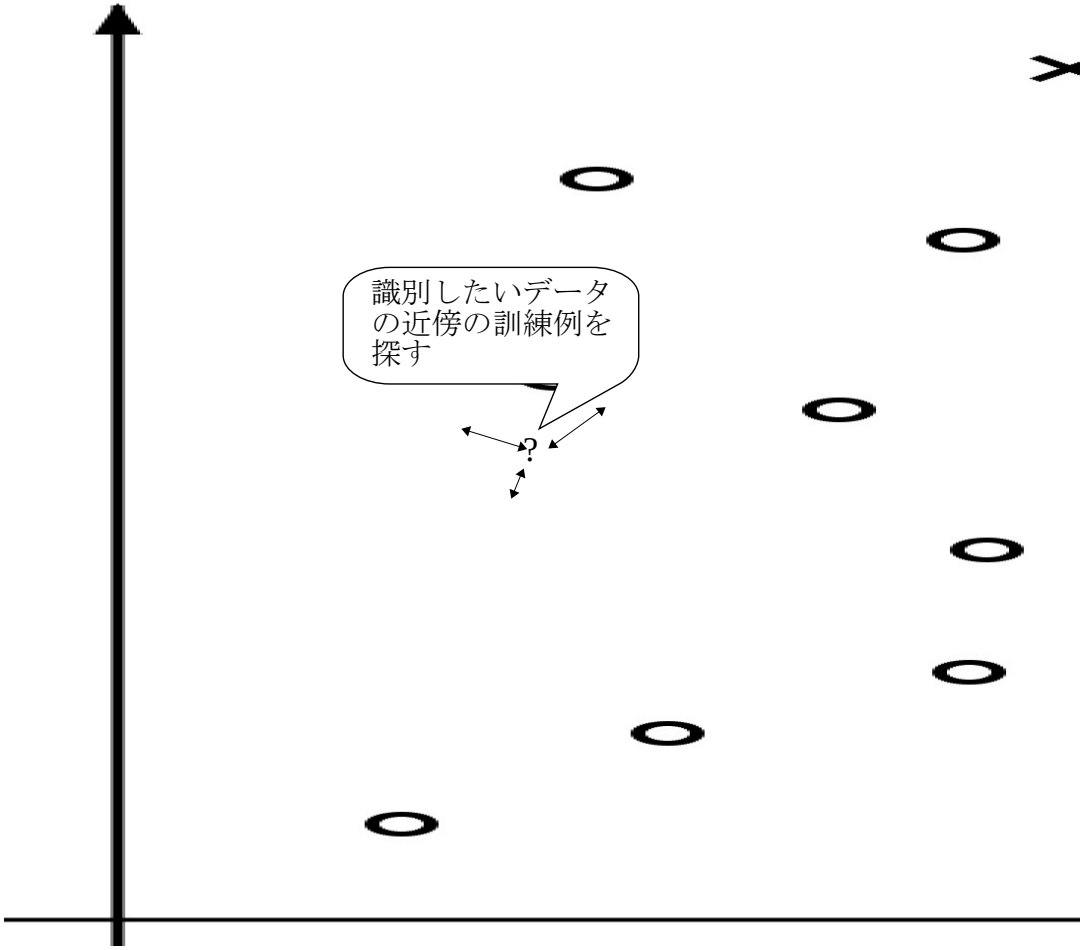
2.1.3 評価基準の設定

- 交差確認法



2.1.4 学習 k-NN 法

- 識別したいデータの近傍の k 個の学習データを探し、属するクラスの多数決で識別



2.1.5 結果の可視化

- 混同行列

	予測＋	予測－
正解＋	true positive(TP)	false negative(FN)
正解－	falsepositive(FP)	true negative(TN)

- 正解率 $Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$

  正解の割合
  クラスの出現率に
  偏りがある場合は不適

- 精度 $Precision = \frac{TP}{TP + FP}$

  正例の判定が
  正しい割合

- 再現率 $Recall = \frac{TP}{TP + FN}$

  正しく判定された
  正例の割合

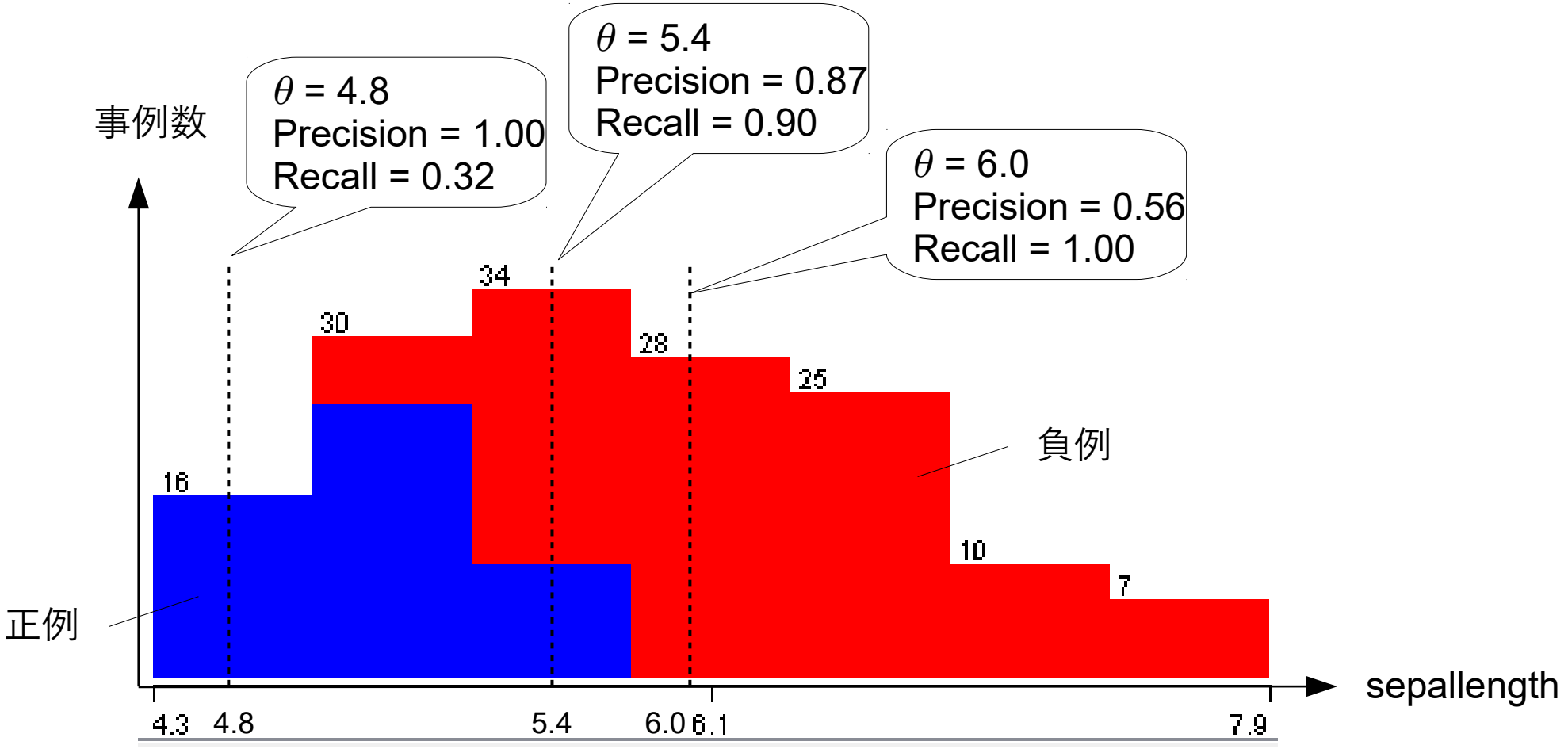
  トレードオフ

- F 値 $F\text{-}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

  精度と再現率の
  調和平均

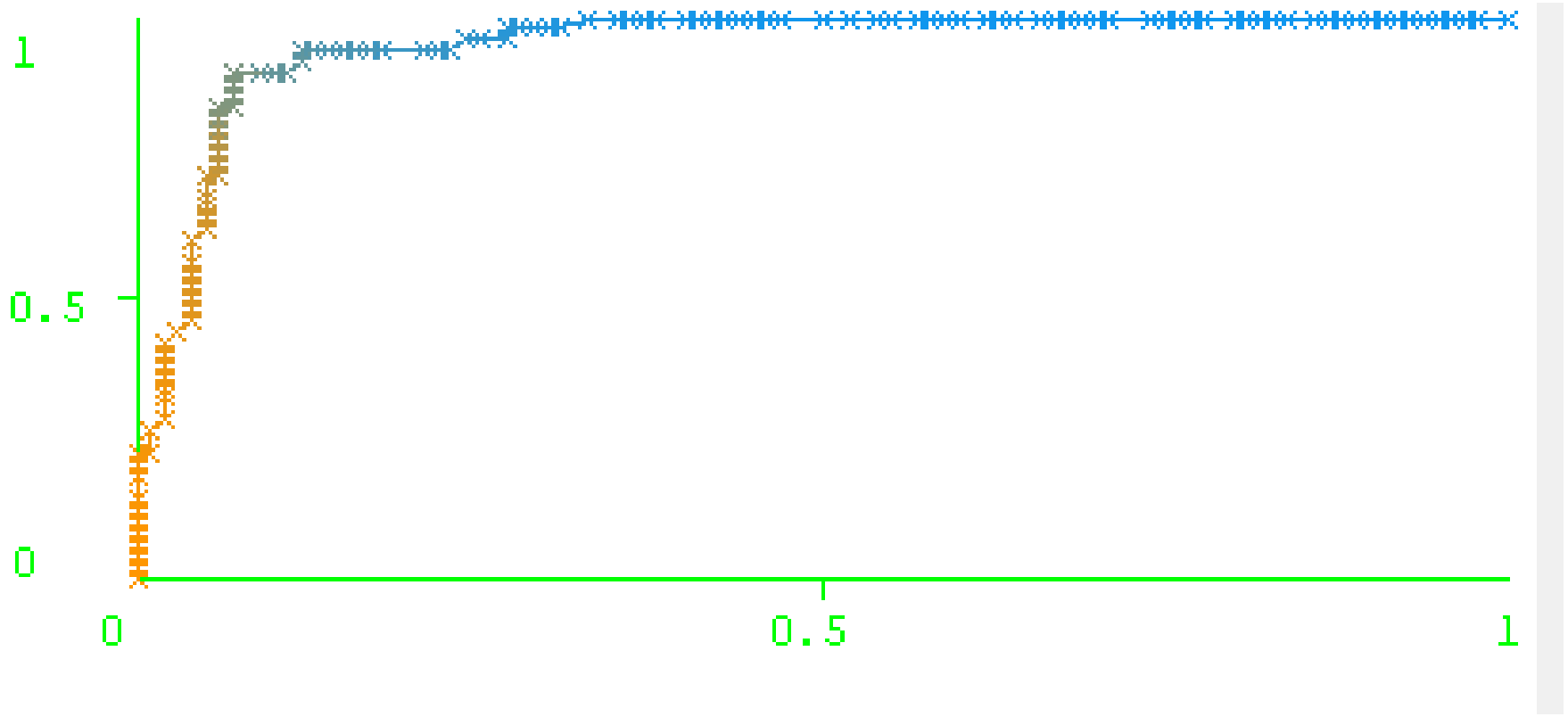
2.1.5 結果の可視化

- 識別のための閾値の設定
  - sepallength 特徴による Iris-setosa の識別



2.1.5 結果の可視化

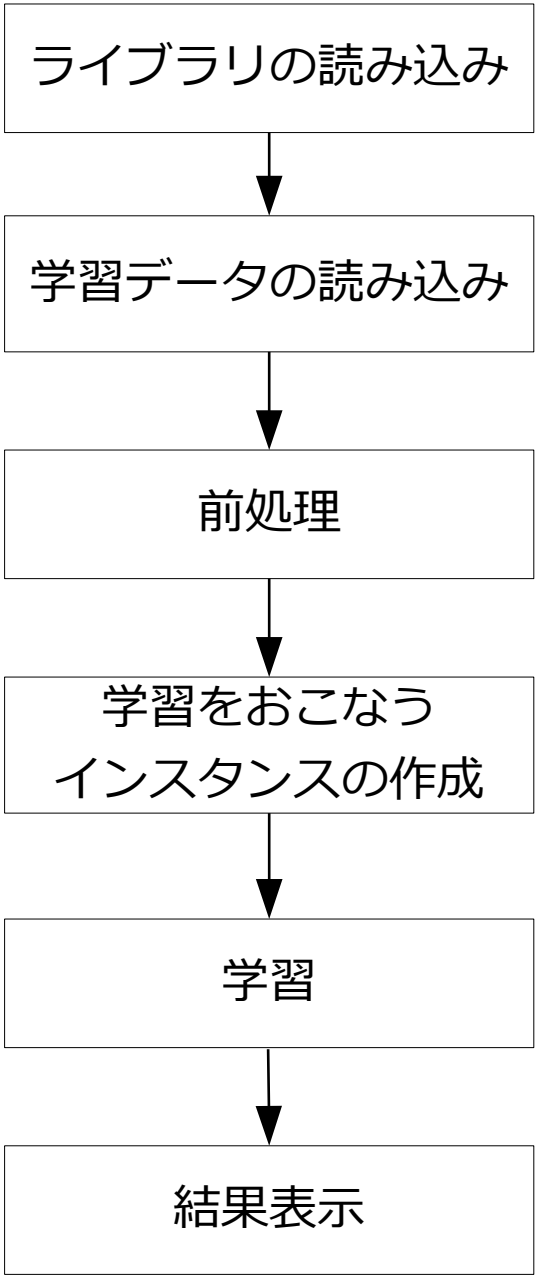
- 精度と再現率のトレードオフ
  - ROC 曲線



2.2 Python による機械学習

- Python を使うメリット
  - データ処理や機械学習のパッケージが充実
  - グラフ表示などの可視化が容易
  - Jupyter Notebook で、実行手順を記録しながらコーディングが可能

2.2 Python による機械学習



- 組み込みデータは datasets パッケージを利用
- 外部データは pandas の read_csv 等を利用

- 標準化：scale
- 主成分分析：PCA

- 学習パラメータを与えてインスタンスを作成

- fit に学習データを与えて学習
- 分割学習法では predict で予測を得る
- 交差確認法では cross_val_score を実行

- 分割学習法では confusion_matrix で混同行列を求める
- 交差確認法では、結果から平均・標準偏差などを求める