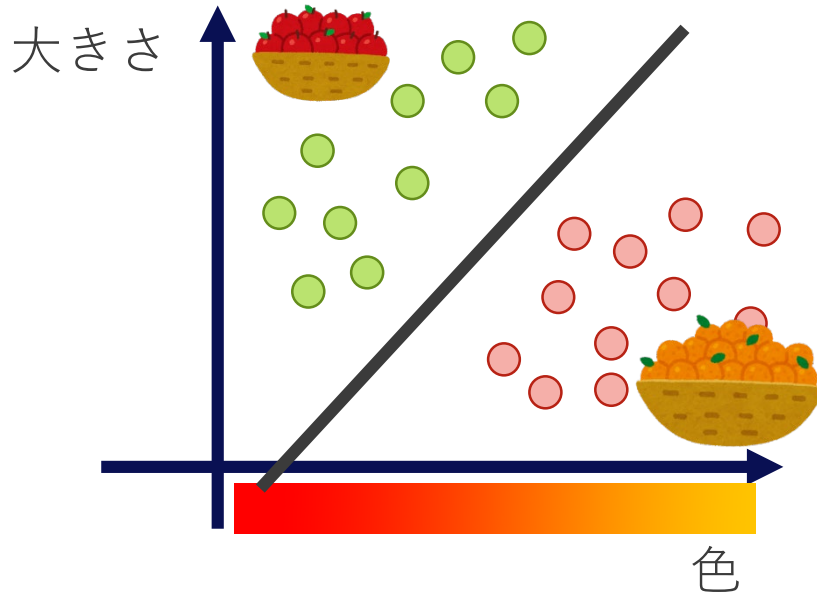


教師あり学習と教師なし学習と確率

教師あり学習（識別問題）と教師なし学習（クラスタリング）

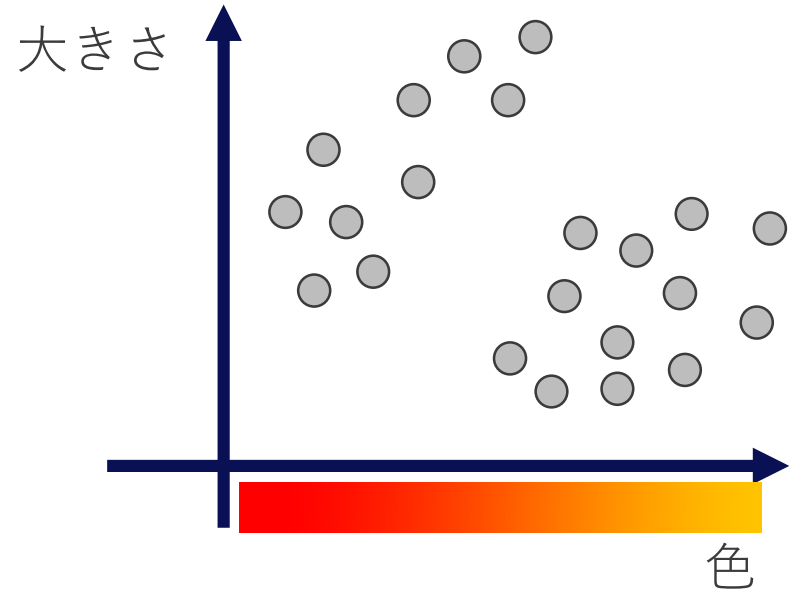
識別問題（第二回講義）：
データをクラスに識別する問題

与えられた教師データからそれらを分離する境界を計算



クラスタリング：
データを複数の集合(クラスタ)に分離する問題

どちらがリンゴかミカンか教えなくとも二つの集団があることがわかる



確率の復習

(離散)確率変数 X , 実現値 x , 確率 $P(X = x)$

同時確率 $P(X = x, Y = y)$

$$P(X = 0, Y = 0) = 40/100$$

周辺確率 $P(Y = y) = \sum_x P(x, y)$

$$P(Y = 0) = 40/100 + 20/100$$

条件付確率 $P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$

$$P(X = 0|Y = 0) = \frac{40/100}{30/50} = 2/3$$

ベイズの定理

$$P(Y = y|X = x) = \frac{P(X = x|Y = y) P(Y = y)}{P(X = x)}$$

$$P(Y = 0|X = 0) = \frac{2/3 \cdot 3/5}{1/2} = 1/5$$

| $X \setminus Y$ | 0 | 1 |
|-----------------|----|----|
| 0 | 40 | 10 |
| 1 | 20 | 30 |

度数分布表

ナイーブベイズ識別器（教師あり学習）（1/3）

スパムメールフィルタの問題：特定の単語がメールに含まれているかどうかでスパムメールを識別したい

スパムメール中の単語の出現確率

$$\frac{\text{単語がスパムメールに出現した回数}}{\text{全スパムメールの数}}$$

$P(\text{今だけ} = 1 | \text{スパム} = 1)$

$P(\text{【重要】} = 1 | \text{スパム} = 1)$

.....

非スパムメール中の単語の出現確率

$$\frac{\text{単語が非スパムメールに出現した回数}}{\text{全非スパムメールの数}}$$

$P(\text{今だけ} = 1 | \text{スパム} = 0)$

$P(\text{【重要】} = 1 | \text{スパム} = 0)$

.....

スパムメールと非スパムメールの確率

$$P(\text{スパム} = 1) = \frac{\text{スパムメールの数}}{\text{全メールの数}}$$

$$P(\text{スパム} = 0) = \frac{\text{非スパムメールの数}}{\text{全メールの数}}$$

ナイーブベイズ識別器（教師あり学習）(2/3)

今、あるメールが来た時に知りたい確率

$P(\text{スパム} = 1 | \text{今だけ} = 1, \text{【重要】} = 1)$

ナイーブベイズ識別器（教師あり学習）（2/3）

今、あるメールが来た時に知りたい確率

$P(\text{スパム} = 1 | \text{今だけ} = 1, \text{【重要】} = 1)$



ベイズの定理

$$P(Y = y | X = x) = \frac{P(X = x | Y = y) P(Y = y)}{P(X = x)}$$

$P(\text{今だけ} = 1, \text{【重要】} = 1 | \text{スパム} = 1) P(\text{スパム} = 1)$

$P(\text{今だけ} = 1, \text{【重要】} = 1)$

ナイーブベイズ識別器（教師あり学習）（2/3）

今、あるメールが来た時に知りたい確率

$$P(\text{スパム} = 1 | \text{今だけ} = 1, \text{【重要】} = 1)$$

ベイズの定理

$$P(Y = y | X = x) = \frac{P(X = x | Y = y) P(Y = y)}{P(X = x)}$$

$$= \frac{P(\text{今だけ} = 1, \text{【重要】} = 1 | \text{スパム} = 1) P(\text{スパム} = 1)}{P(\text{今だけ} = 1, \text{【重要】} = 1)}$$

仮定：条件付き独立

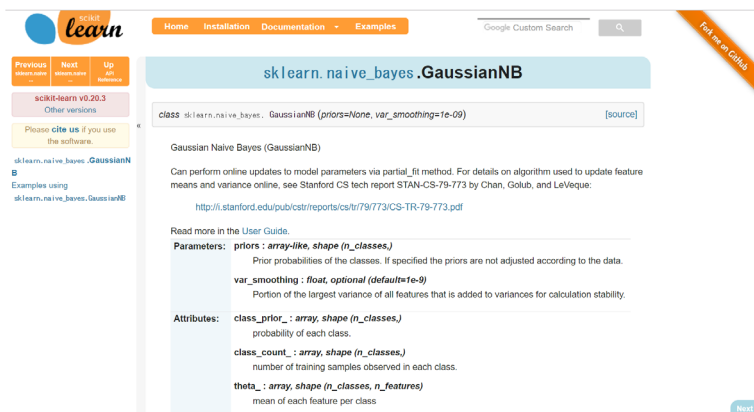
$$P(X = x, Y = y | C = c) = P(X = x | C = c) P(Y = y | C = c)$$

$$= \frac{P(\text{今だけ} = 1, | \text{スパム} = 1) P(\text{【重要】} = 1, | \text{スパム} = 1) P(\text{スパム} = 1)}{P(\text{今だけ} = 1, \text{【重要】} = 1)}$$

この計算をすれば新たに来たメールのスパム判定ができる

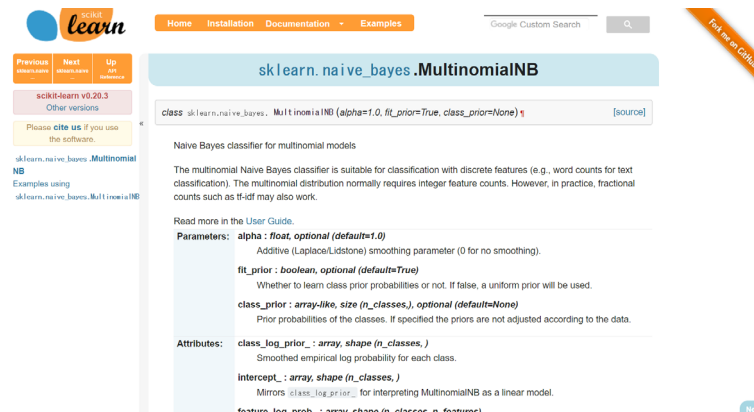
ナイーブベイズ識別器（教師あり学習）（3/3）

ナイーブベイズ識別器の特徴の一つ：確率分布を用いているので様々な分布のものをデータの形式に合わせて選択できる



The screenshot shows the sklearn documentation page for `sklearn.naive_bayes.GaussianNB`. The page title is "sklearn.naive_bayes.GaussianNB". The code snippet is `class sklearn.naive_bayes.GaussianNB (priors=None, var_smoothing=1e-09)`. The description states: "Gaussian Naive Bayes (GaussianNB) Can perform online updates to model parameters via `partial_fit` method. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque: <http://l.stanford.edu/pub/cstr/reports/cstr/79/773/CS-TR-79-773.pdf>". The parameters section lists: `priors`: array-like, shape (n_classes,); `var_smoothing`: float, optional (default=1e-9). The attributes section lists: `class_prior_`: array, shape (n_classes,); `class_count_`: array, shape (n_classes,); `theta_`: array, shape (n_classes, n_features).

ガウス分布
（連続値データ）



The screenshot shows the sklearn documentation page for `sklearn.naive_bayes.MultinomialNB`. The page title is "sklearn.naive_bayes.MultinomialNB". The code snippet is `class sklearn.naive_bayes.MultinomialNB (alpha=1.0, fit_prior=True, class_prior=None)`. The description states: "Naive Bayes classifier for multinomial models The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as (f-df) may also work." The parameters section lists: `alpha`: float, optional (default=1.0); `fit_prior`: boolean, optional (default=True); `class_prior`: array-like, size (n_classes,), optional (default=None). The attributes section lists: `class_log_prior_`: array, shape (n_classes,); `feature_log_prob_`: array, shape (n_classes, n_features).

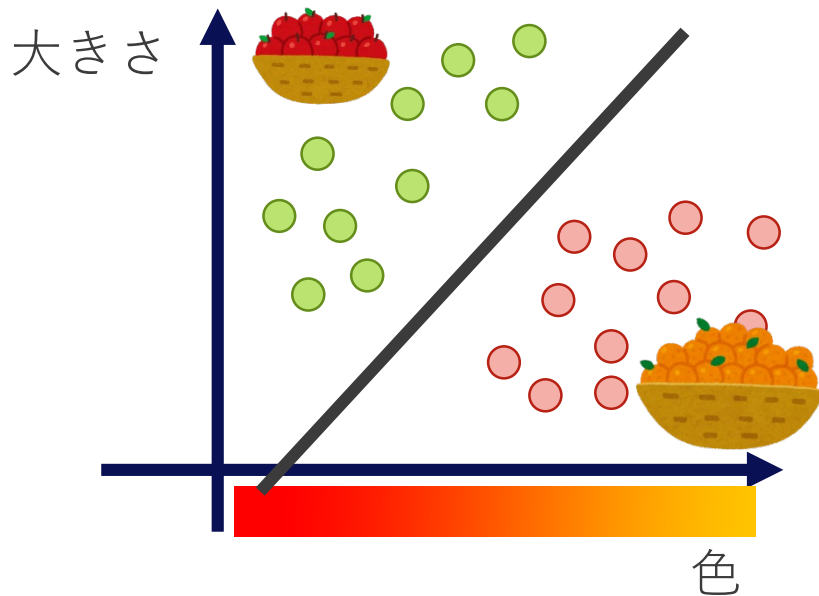
多項分布
（カテゴリーデータ）

教師あり学習（識別問題）と教師なし学習（クラスタリング）

識別問題：

データをクラスに識別する問題

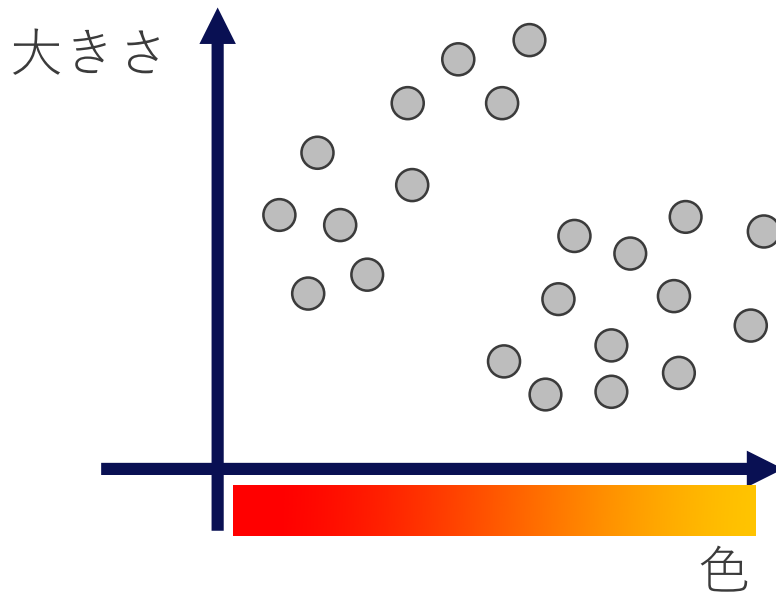
与えられた教師データからそれらを分離する境界を計算



クラスタリング：

データを複数の集合(クラスタ)に分離する問題

どちらがリンゴかミカンか教えなくとも二つの集団があることがわかる

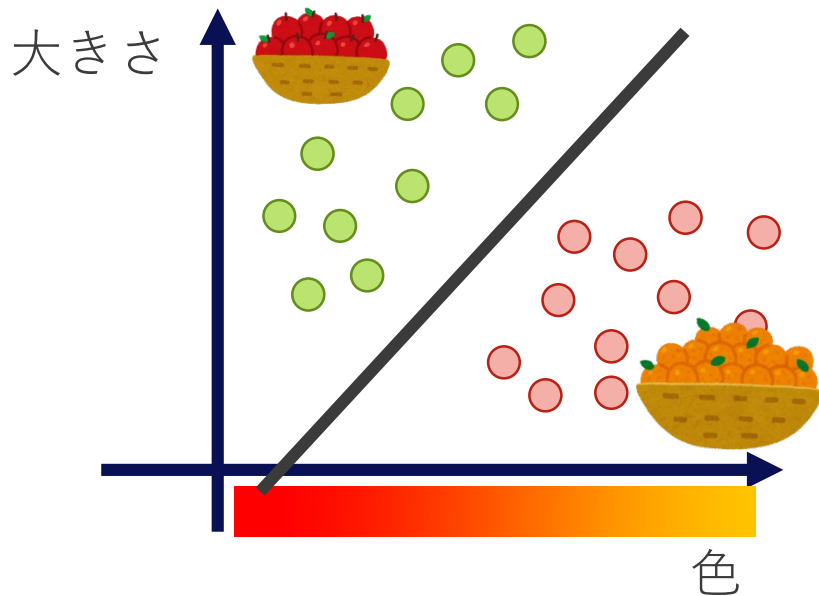


教師あり学習（識別問題）と教師なし学習（クラスタリング）

識別問題：

データをクラスに識別する問題

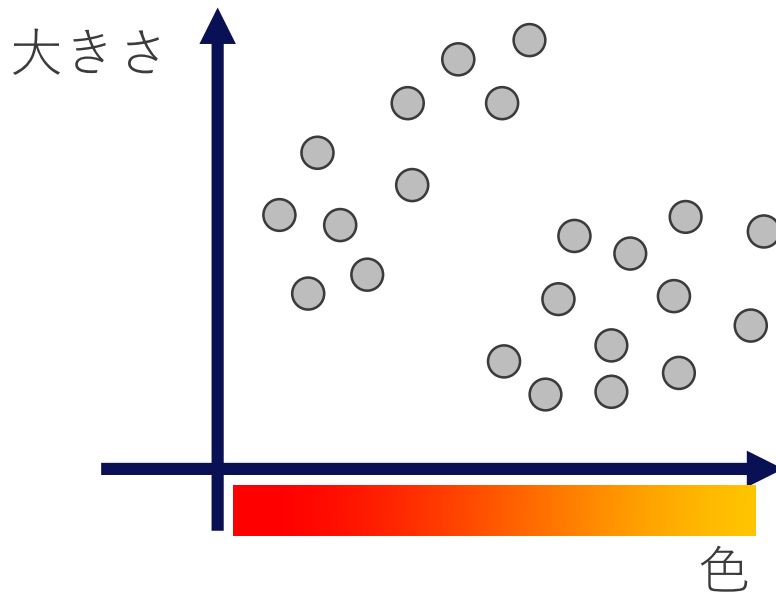
与えられた教師データからそれらを分離する境界を計算



クラスタリング：

データを複数の集合(クラスタ)に分離する問題

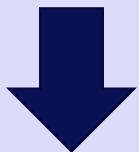
どちらがリンゴかミカンか教えなくとも二つの集団があることがわかる



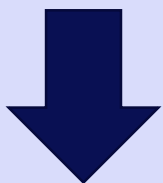
ナイーブベイズクラスタリング

ナイーブベイズ識別器

クラス分けされた文書



$P(X = x|C = c)$ を計算する



c : クラス
(スパムかどうか)
 x : 単語の有無

文書のクラスの確率を計算できる

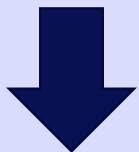
$P(C = c|X = x)$

ナイーブベイズクラスタリング

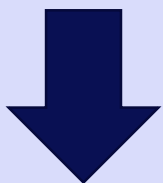
文書のクラスをランダムに割り当てる

ナイーブベイズ識別器

クラス分けされた文書



$P(X = x|C = c)$ を計算する



c : クラス
(スパムかどうか)
 x : 単語の有無

文書のクラスの確率を計算できる

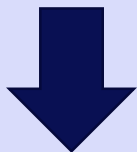
$P(C = c|X = x)$

ナイーブベイズクラスタリング

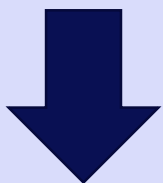
文書のクラスをランダムに割り当てる

ナイーブベイズ識別器

クラス分けされた文書



$P(X = x|C = c)$ を計算する

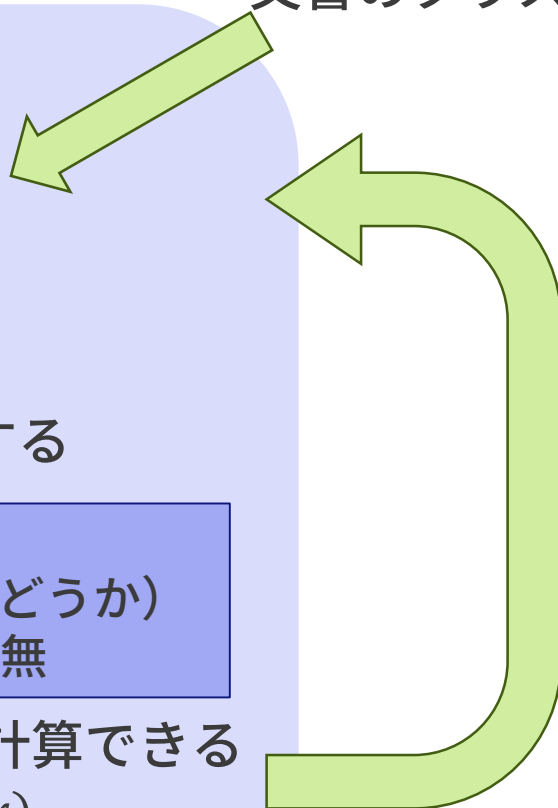


文書のクラスの確率を計算できる

$$P(C = c|X = x)$$

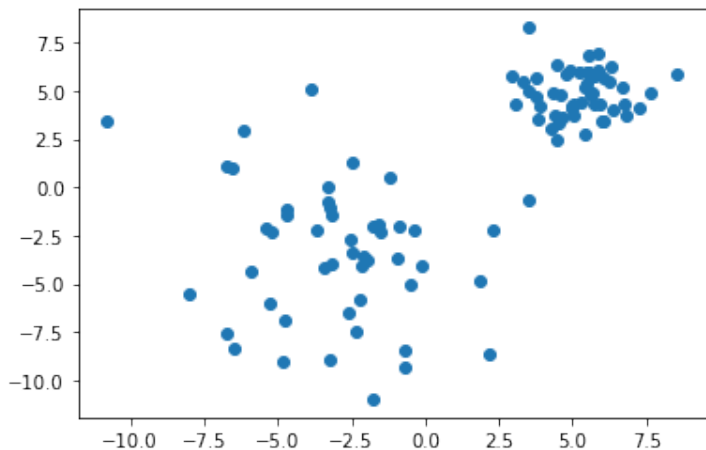
c : クラス
(スパムかどうか)
 x : 単語の有無

再度、文書のクラス
を割り当てなおす

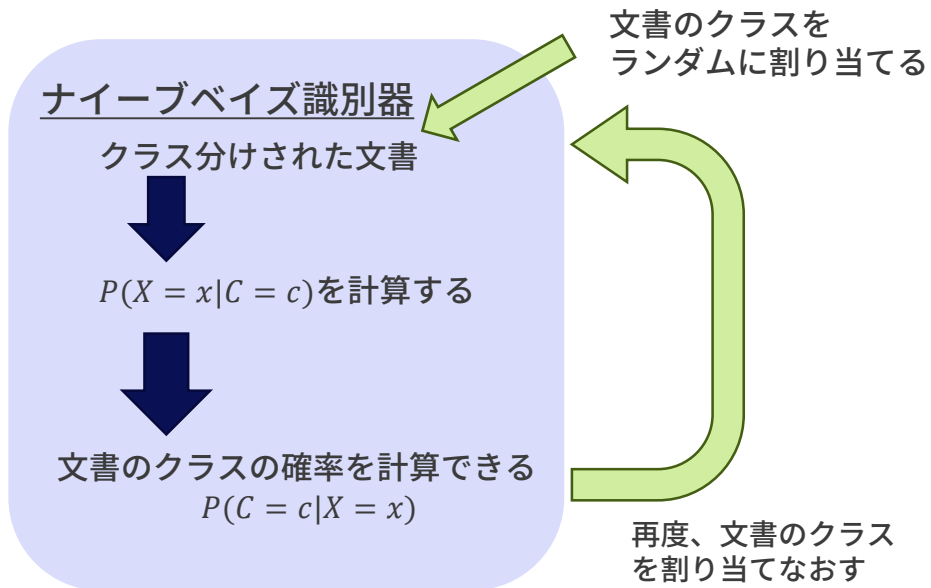
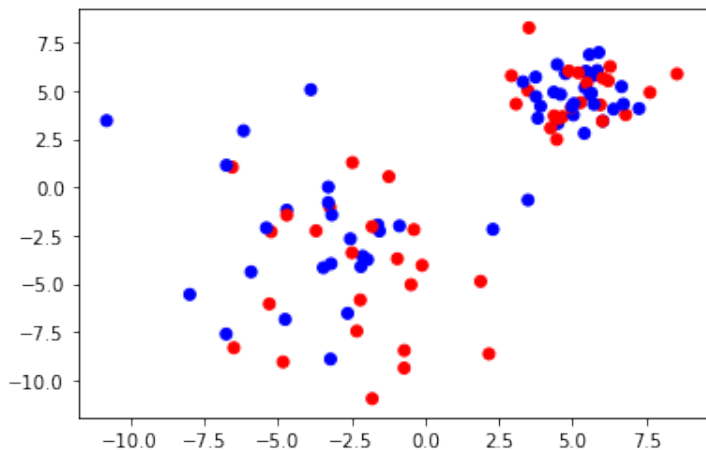


初期化 (ナイーブベイズクラスタリング)

元データ



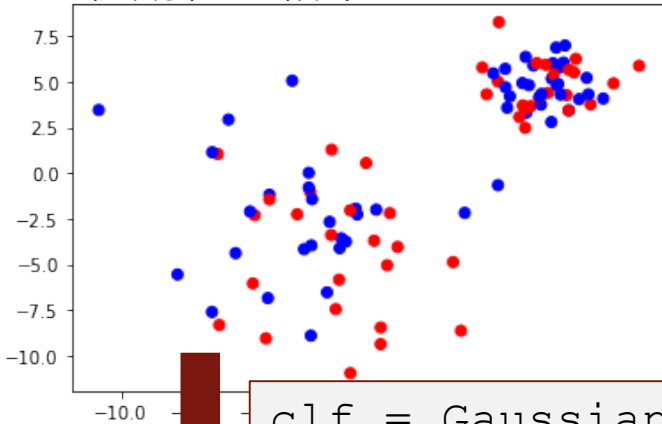
クラスをランダムに割り当てる



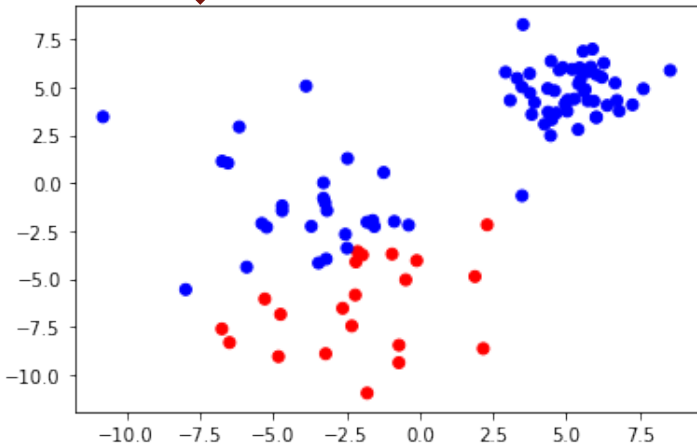
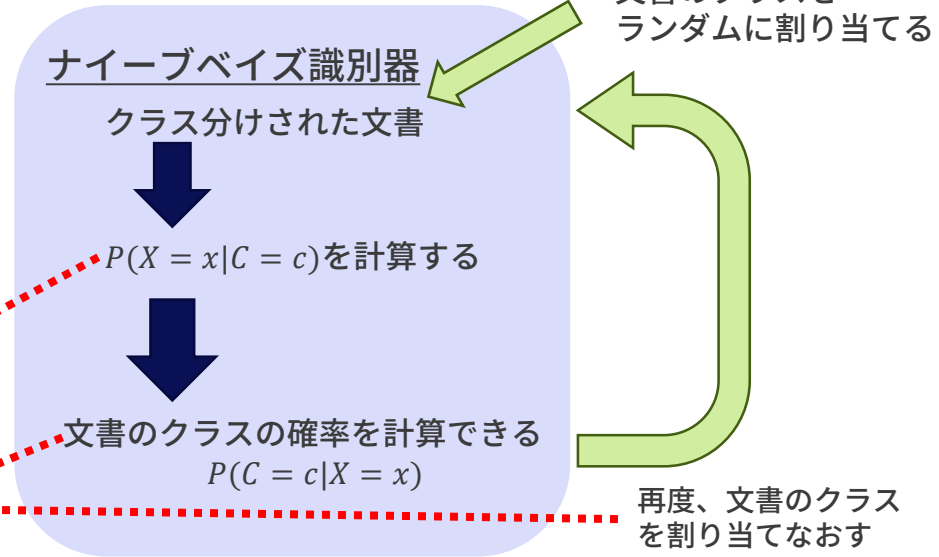
c : クラス
(スパムかどうか)
 x : 単語の有無

繰り返し1回目 (ナイーブベイズクラスタリング)

初期化の結果



```
clf = GaussianNB()  
clf.fit(X, Y)  
newY=clf.predict(X)
```



sklearn.naive_bayes.GaussianNB

```
class sklearn.naive_bayes.GaussianNB (priors=None, var_smoothing=1e-09)
```

Gaussian Naive Bayes (GaussianNB)

Can perform online updates to model parameters via partial_fit method. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque:
<http://l.stanford.edu/pub/cstr/reports/cs/tr/79/773/CS-TR-79-773.pdf>

Read more in the User Guide.

Parameters: **priors** : array-like, shape (n_classes,)
Prior probabilities of the classes. If specified the priors are not adjusted according to the data.

var_smoothing : float, optional (default=1e-9)
Portion of the largest variance of all features that is added to variances for calculation stability.

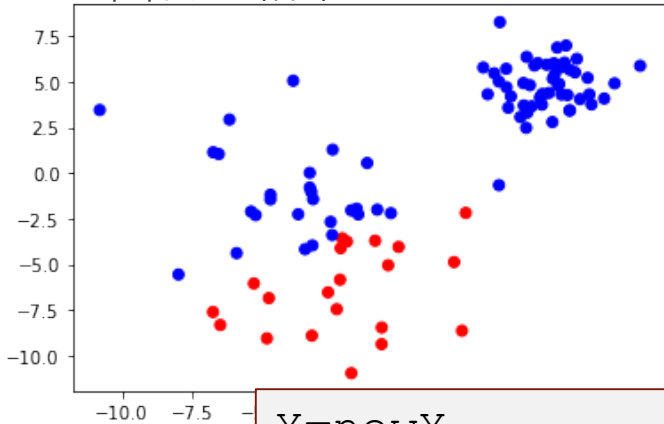
Attributes: **class_prior_** : array, shape (n_classes,)
probability of each class.

class_count_ : array, shape (n_classes,)
number of training samples observed in each class.

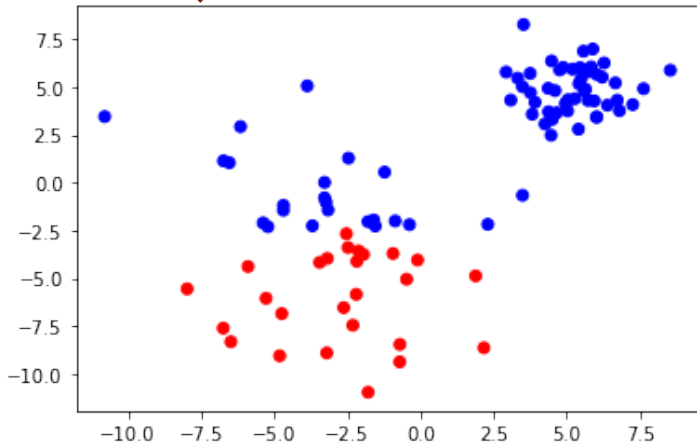
theta_ : array, shape (n_classes, n_features)
mean of each feature per class

繰り返し2回目 (ナイーブベイズクラスタリング)

1回目の結果



```
Y=newY
clf.fit(X, Y)
newY=clf.predict(X)
```



ナイーブベイズ識別器

クラス分けされた文書

$P(X = x|C = c)$ を計算する

文書のクラスの確率を計算できる
 $P(C = c|X = x)$

文書のクラスを
ランダムに割り当てる

再度、文書のクラス
を割り当てなおす

sklearn.naive_bayes.GaussianNB

```
class sklearn.naive_bayes.GaussianNB (priors=None, var_smoothing=1e-09)
```

Gaussian Naive Bayes (GaussianNB)

Can perform online updates to model parameters via `partial_fit` method. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque: <http://l.stanford.edu/pub/cstr/reports/cs/tr/79/773/CS-TR-79-773.pdf>

Read more in the User Guide.

Parameters: **priors** : array-like, shape (n_classes,)
Prior probabilities of the classes. If specified the priors are not adjusted according to the data.

var_smoothing : float, optional (default=1e-9)
Portion of the largest variance of all features that is added to variances for calculation stability.

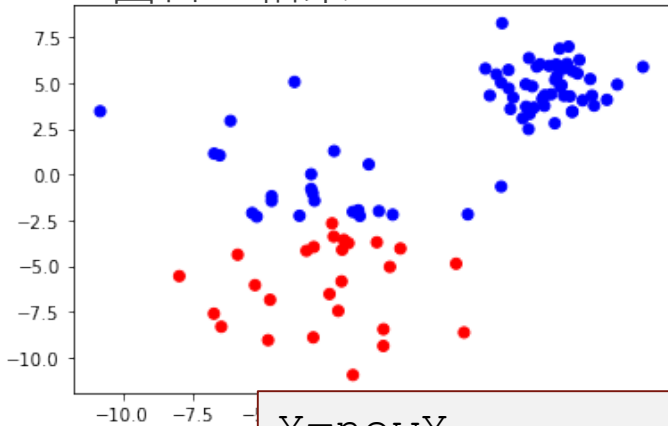
Attributes: **class_prior_** : array, shape (n_classes,)
probability of each class.

class_count_ : array, shape (n_classes,)
number of training samples observed in each class.

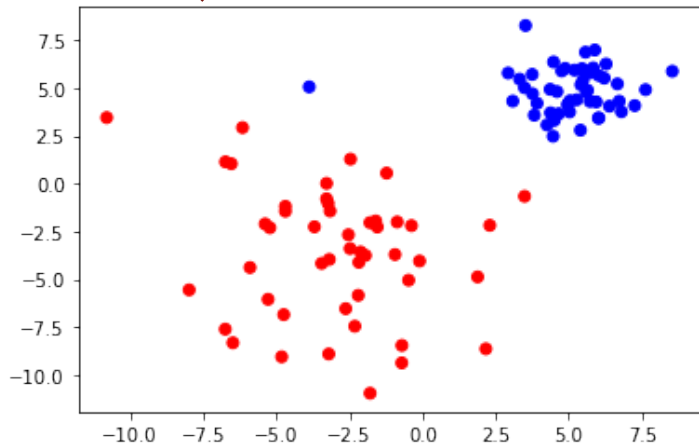
theta_ : array, shape (n_classes, n_features)
mean of each feature per class

繰り返し3回目 (ナイーブベイズクラスタリング)

2回目の結果



```
Y=newY
clf.fit(X, Y)
newY=clf.predict(X)
```



ナイーブベイズ識別器

クラス分けされた文書

$P(X = x|C = c)$ を計算する

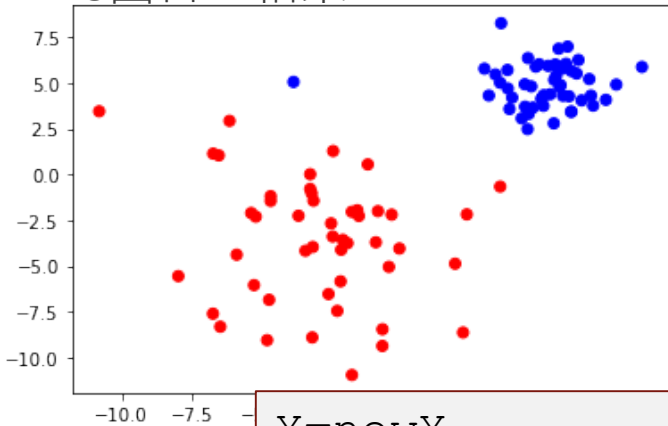
文書のクラスの確率を計算できる
 $P(C = c|X = x)$

文書のクラスを
ランダムに割り当てる

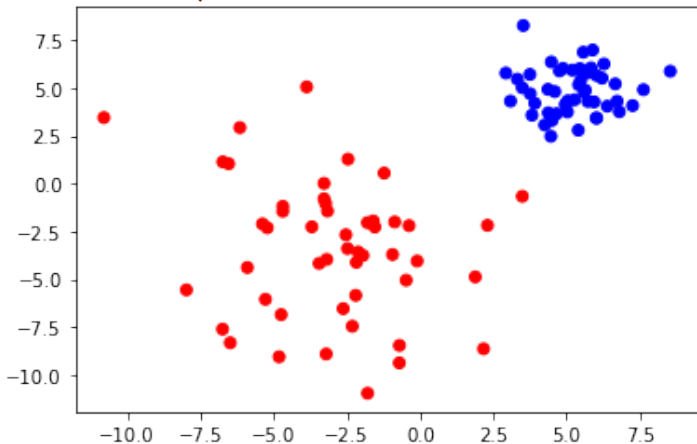
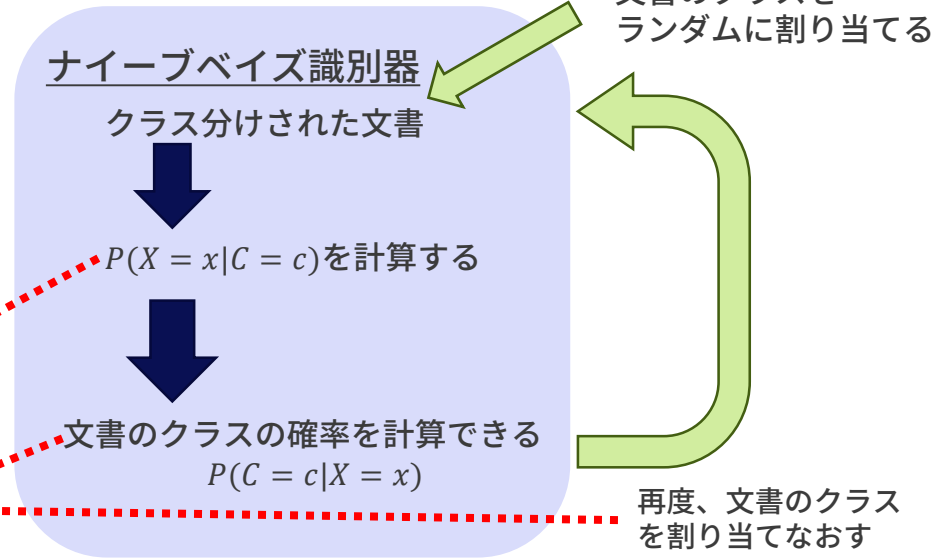
再度、文書のクラス
を割り当てなおす

繰り返し4回目 (ナイーブベイズクラスタリング)

3回目の結果



```
Y=newY
clf.fit(X, Y)
newY=clf.predict(X)
```



sklearn.naive_bayes.GaussianNB

```
class sklearn.naive_bayes.GaussianNB (priors=None, var_smoothing=1e-09)
```

Gaussian Naive Bayes (GaussianNB)

Can perform online updates to model parameters via partial_fit method. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque: <http://l.stanford.edu/pub/cstr/reports/cs/tr/79/773/CS-TR-79-773.pdf>

Read more in the User Guide.

Parameters: **priors** : array-like, shape (n_classes,)
Prior probabilities of the classes. If specified the priors are not adjusted according to the data.

var_smoothing : float, optional (default=1e-9)
Portion of the largest variance of all features that is added to variances for calculation stability.

Attributes: **class_prior_** : array, shape (n_classes,)
probability of each class.

class_count_ : array, shape (n_classes,)
number of training samples observed in each class.

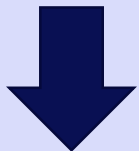
theta_ : array, shape (n_classes, n_features)
mean of each feature per class

ナイーブベイズクラスタリング

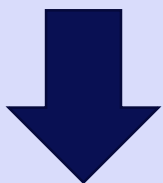
文書のクラスをランダムに割り当てる

ナイーブベイズ識別器

クラス分けされた文書



$P(X = x | C = c)$ を計算する



文書のクラスの確率を計算できる

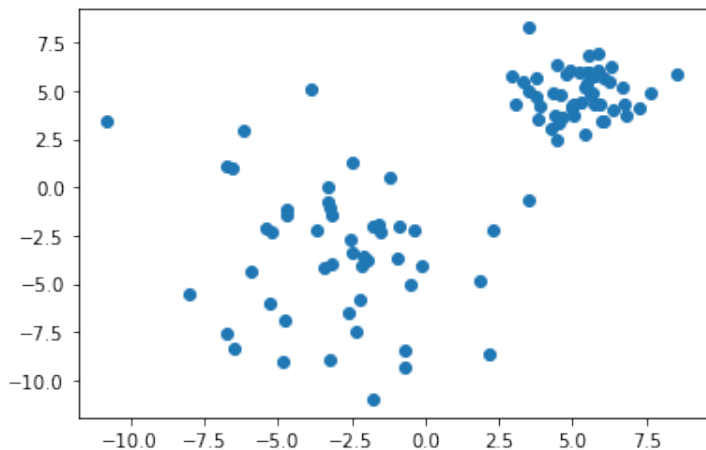
$$P(C = c | X = x)$$

c : クラス
(スパムかどうか)
 x : 単語の有無

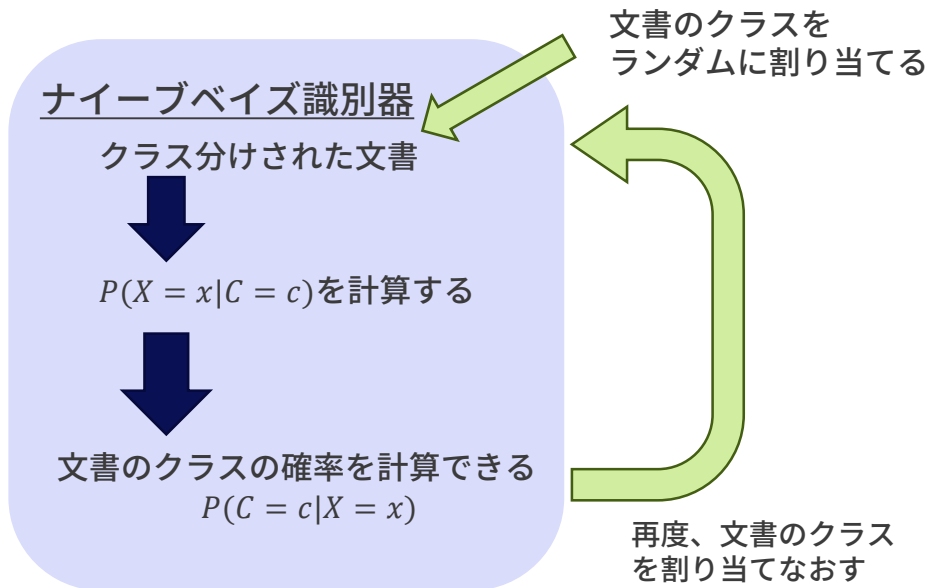
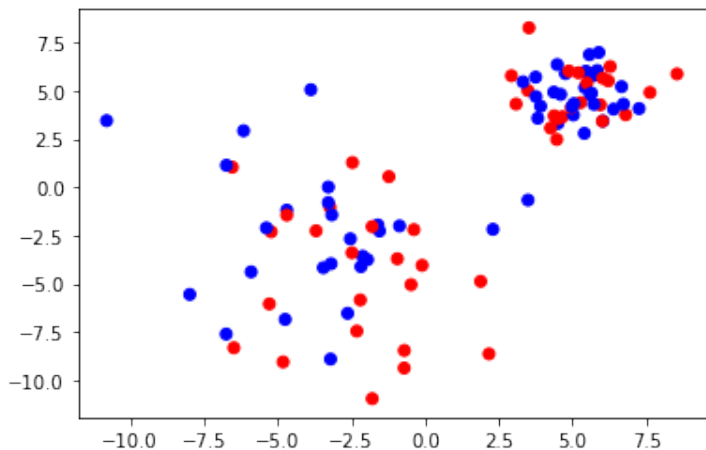
再度、文書のクラス
を**確率的に**割り当てる

初期化 (ナイーブベイズクラスタリング)

元データ



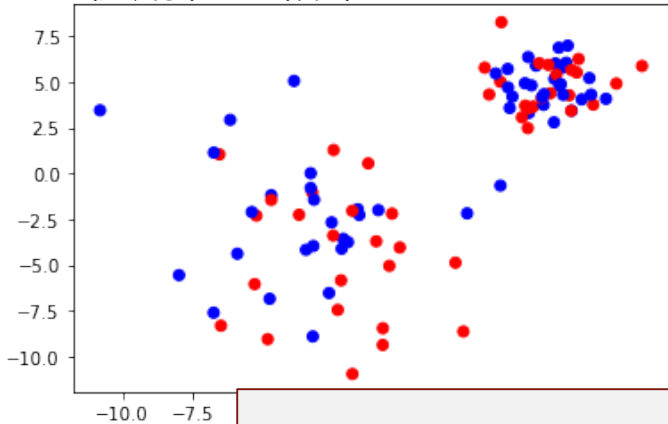
クラスをランダムに割り当てる



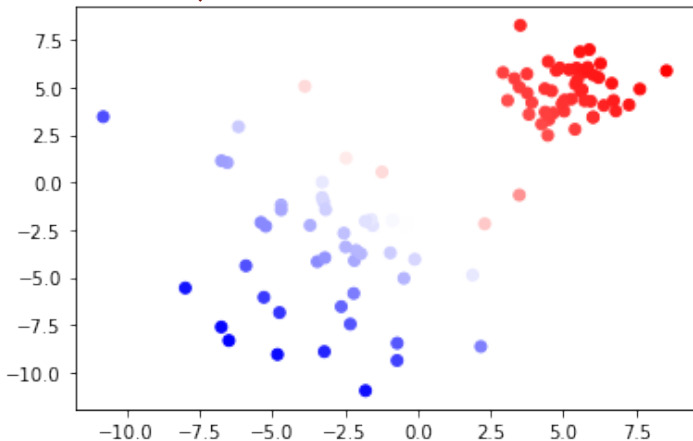
c : クラス
(スパムかどうか)
 x : 単語の有無

繰り返し1回目 (ナイーブベイズクラスタリング)

初期化の結果



```
...  
clf.fit(X, Y)  
newY=clf.predict_proba(X)
```



ナイーブベイズ識別器

クラス分けされた文書

$P(X = x|C = c)$ を計算する

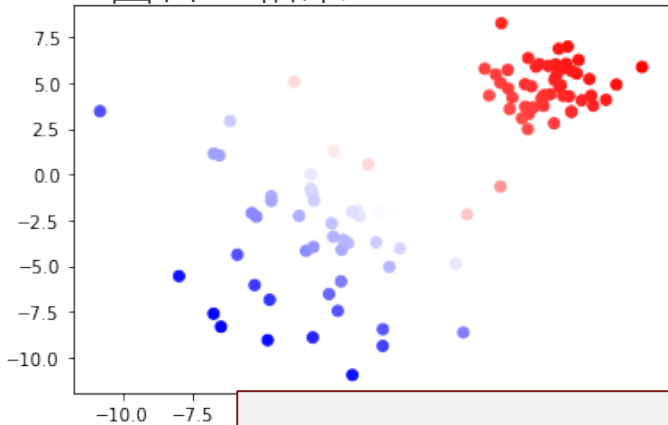
文書のクラスの確率を計算できる
 $P(C = c|X = x)$

文書のクラスを
ランダムに割り当てる

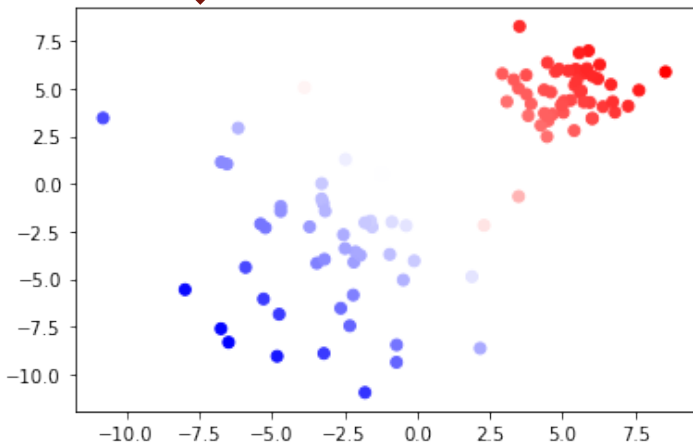
再度、文書のクラス
を確率的に割り当てる

繰り返し2回目 (ナイーブベイズクラスタリング)

1回目の結果



```
...  
clf.fit(X, Y)  
newY=clf.predict_proba(X)
```



ナイーブベイズ識別器

クラス分けされた文書

$P(X = x|C = c)$ を計算する

文書のクラスの確率を計算できる
 $P(C = c|X = x)$

文書のクラスを
ランダムに割り当てる

再度、文書のクラス
を確率的に割り当てる

sklearn.naive_bayes.GaussianNB

```
class sklearn.naive_bayes.GaussianNB (priors=None, var_smoothing=1e-09)
```

Gaussian Naive Bayes (GaussianNB)

Can perform online updates to model parameters via partial_fit method. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque:
<http://l.stanford.edu/pub/cstr/reports/cstr/79/773/CS-TR-79-773.pdf>

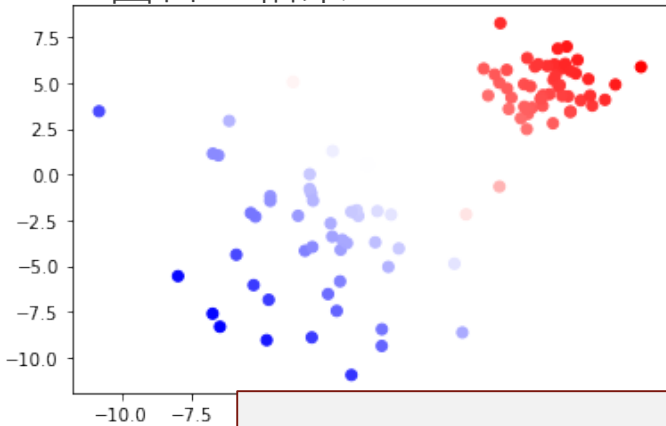
Read more in the User Guide.

Parameters: **priors** : array-like, shape (n_classes,)
Prior probabilities of the classes. If specified the priors are not adjusted according to the data.
var_smoothing : float, optional (default=1e-9)
Portion of the largest variance of all features that is added to variances for calculation stability.

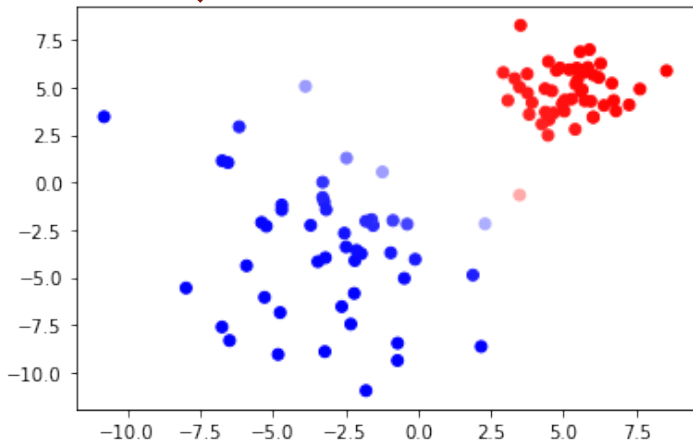
Attributes: **class_prior_** : array, shape (n_classes,)
probability of each class.
class_count_ : array, shape (n_classes,)
number of training samples observed in each class.
theta_ : array, shape (n_classes, n_features)
mean of each feature per class

繰り返し3回目 (ナイーブベイズクラスタリング)

2回目の結果



```
...  
clf.fit(X, Y)  
newY=clf.predict_proba(X)
```



ナイーブベイズ識別器

クラス分けされた文書

$P(X = x|C = c)$ を計算する

文書のクラスの確率を計算できる
 $P(C = c|X = x)$

文書のクラスを
ランダムに割り当てる

再度、文書のクラス
を確率的に割り当てる

sklearn.naive_bayes.GaussianNB

```
class sklearn.naive_bayes.GaussianNB (priors=None, var_smoothing=1e-09)
```

Gaussian Naive Bayes (GaussianNB)

Can perform online updates to model parameters via partial_fit method. For details on algorithm used to update feature means and variance online, see Stanford CS tech report STAN-CS-79-773 by Chan, Golub, and LeVeque:
<http://l.stanford.edu/pub/cstr/reports/cs/tr/79/773/CS-TR-79-773.pdf>

Read more in the User Guide.

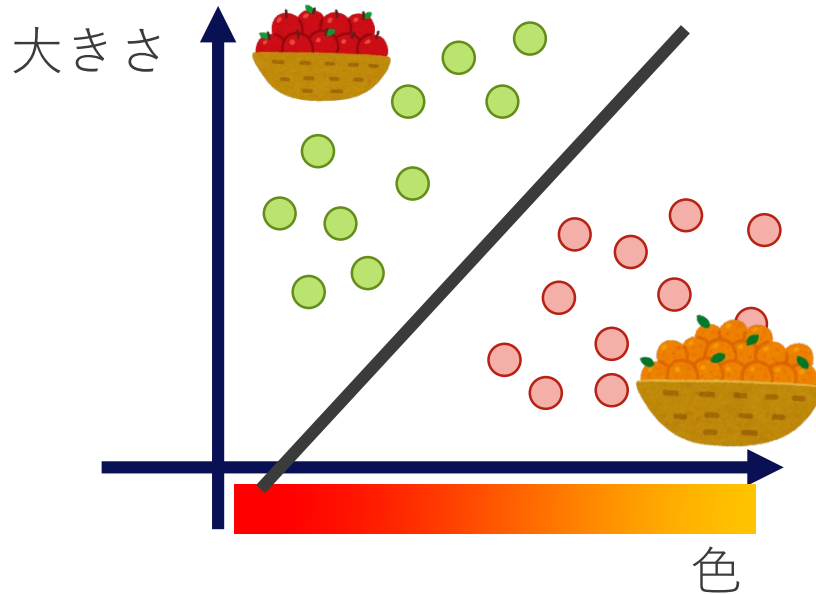
Parameters: **priors** : array-like, shape (n_classes,)
Prior probabilities of the classes. If specified the priors are not adjusted according to the data.
var_smoothing : float, optional (default=1e-9)
Portion of the largest variance of all features that is added to variances for calculation stability.

Attributes: **class_prior_** : array, shape (n_classes,)
probability of each class.
class_count_ : array, shape (n_classes,)
number of training samples observed in each class.
theta_ : array, shape (n_classes, n_features)
mean of each feature per class

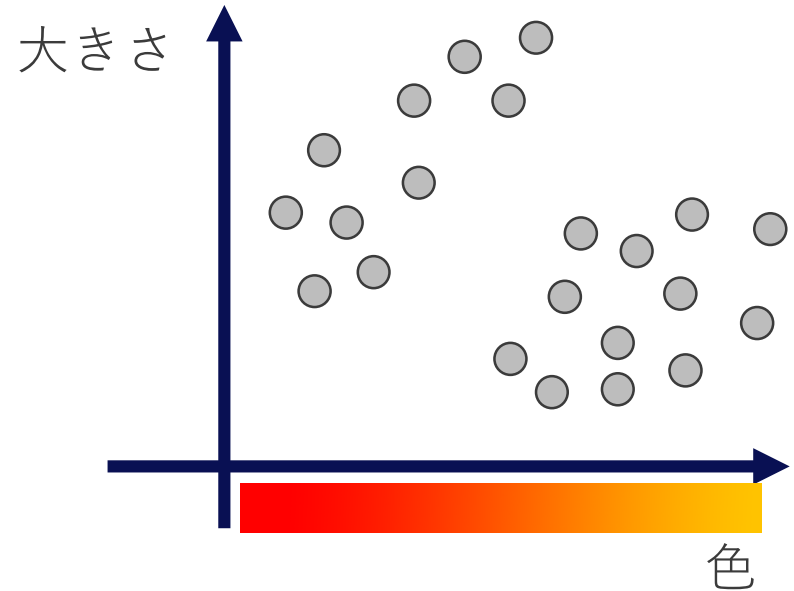
まとめ

確率を使った識別問題と クラスタリングの例を紹介した

識別問題：
データをクラスに識別する問題



クラスタリング：
データを複数の集合(クラスタ)に分離する問題



*今回紹介したクラスタリング法はHARD-EM法（前半）、EM法（後半：確率的なクラス割り当ての）と呼ばれる方法