

データマイニング

第11回 サポートベクターマシン (+ XGBOOST)

2023年春学期

宮津和弘

本日の講義・演習

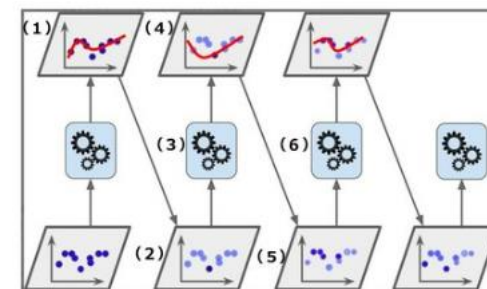
| 日付 | 講義・演習内容 |
|-----------------|-------------------------|
| 04/14/23 | (1) イントロダクション |
| 04/21/23 | (2) ビジネスシミュレーション |
| 04/28/23 | (3) ID-POSデータ分析 |
| 05/12/23 | (4) 対応分析 |
| 05/19/23 | (5) クラスター分析 |
| 05/26/23 | (6) 自己組織化マップ |
| 06/02/23 | (7) 線形判別分析 |
| 06/09/23 | (8) 非線形判別分析 |
| 06/16/23 | (9) ツリーモデル |
| 06/23/23 | (10) 集団学習 |
| 06/30/23 | 休講（※黒門祭のため） |
| 07/04/23 | (11) サポートベクターマシン |
| 07/14/23 | (12) 共分散構造分析 |
| 07/21/23 | (13) テキスト分析 |
| 07/28/23 | (14) まとめ（ポートフォリオ） |



本日の演習概要とポイント

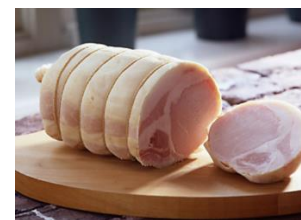
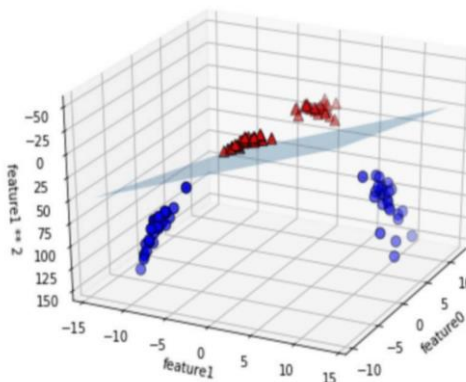
XGBoost

→ タイタニック号事故データ



SVM

→ SPAMメールデータ



機械学習の手法

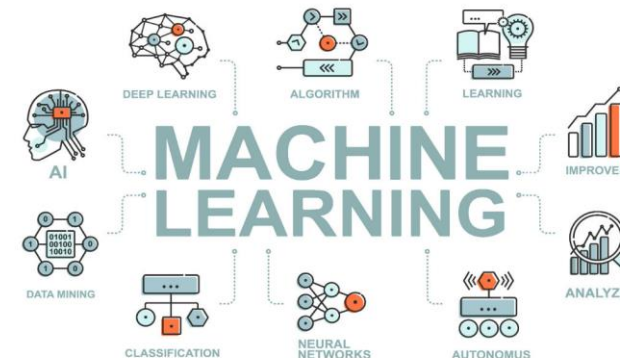
教師データあり

- ✓ 線形回帰
- ✓ ロジスティック回帰
- サポートベクターマシン (SVM)
- ✓ 分類木
- ✓ 回帰木
- ✓ ランダムフォレスト
- 勾配ブースティング木 (XGBoost)
- ✓ ニューラルネットワーク
- 畳み込みニューラルネットワーク
- 再起型ニューラルネットワーク
- ✓ ナイーブベイズ
- k近傍法ブースティング
- ✓ バギング

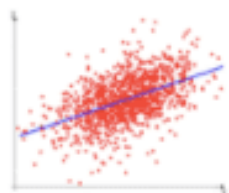
教師データなし

- ✓ 階層型クラスタリング(ワード法など)
- ✓ 非階層型クラスタリング (k-meansなど)
- トピックモデル (LDAなど)
- ✓ 自己組織化マップ
- ✓ アソシエーション分析 (*)
- ✓ 協調フィルタリング (*)
- ✓ ベイジアンネットワーク (*)

* データサイエンス演習 1



機械学習の手法 2 – 分類と予測 –



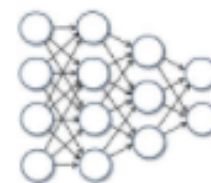
回帰

- ・ 線形回帰
- ・ ロジスティック回帰
- ・ サポートベクターマシン (SVM)



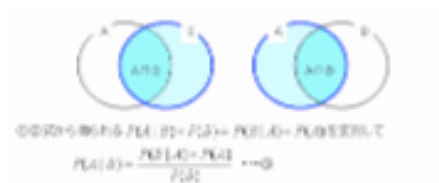
木

- ・ 決定木
- ・ 回帰木
- ・ ランダムフォレスト
- ・ XGBoost



ニューラルネット

- ・ 単純パーセプトロン
- ・ DNN
- ・ CNN
- ・ RNN



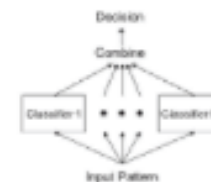
ベイズ (事後確率)

- ・ ナイーブベイズ



クラスタリング

- ・ k-means
- ・ k-means++



アンサンブル学習

- ・ Boosting
- ・ Adaboost

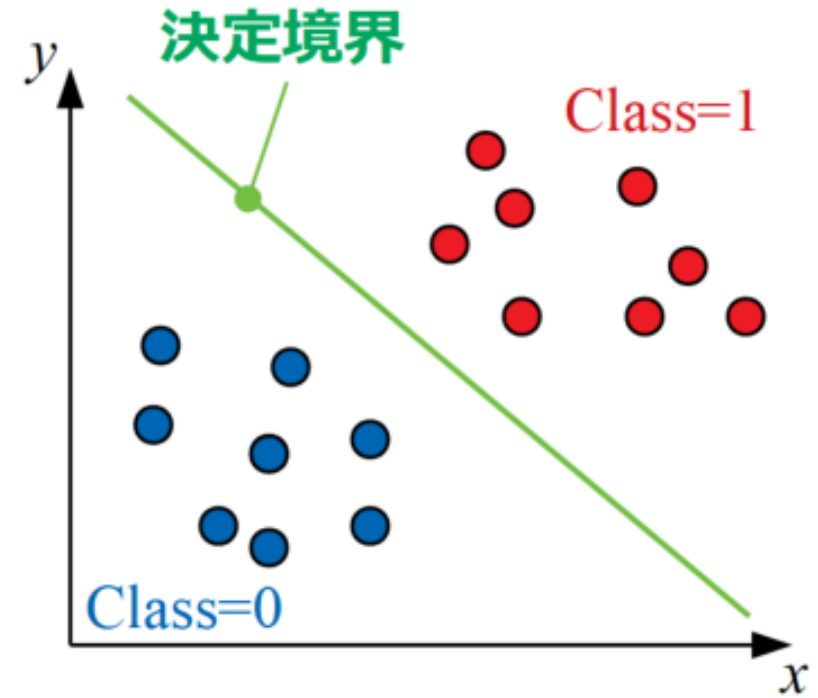


SVM

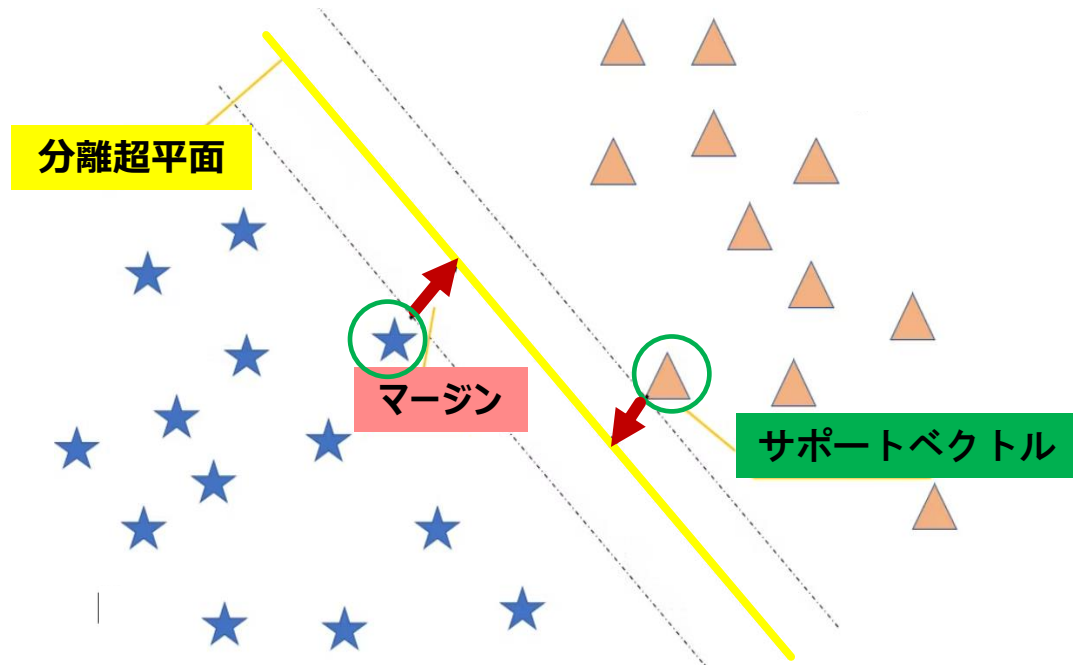


サポートベクトルマシン (SVM)

サポートベクトルマシン(SVM)とは、線形判別関数を用いて2クラスのパターン識別器を利用した教師あり学習による機械学習の手法であり、**判別境界マージンを最大化**するように識別器が構成される。また、カーネル法を利用することで、境界線が非線形となる元データに対して、**カーネルにより境界線が線形**となるような特徴空間に移動して判別する。



分離超平面とマージン



■ 分離超平面

→ n 次元データを $n-1$ 次元の平面で分離するとき
その平面を分離超平面と呼ぶ

■ サポートベクトル

→ 分離超平面に最も近接したデータ

■ マージン

→ 分離超平面とサポートベクトルとの距離

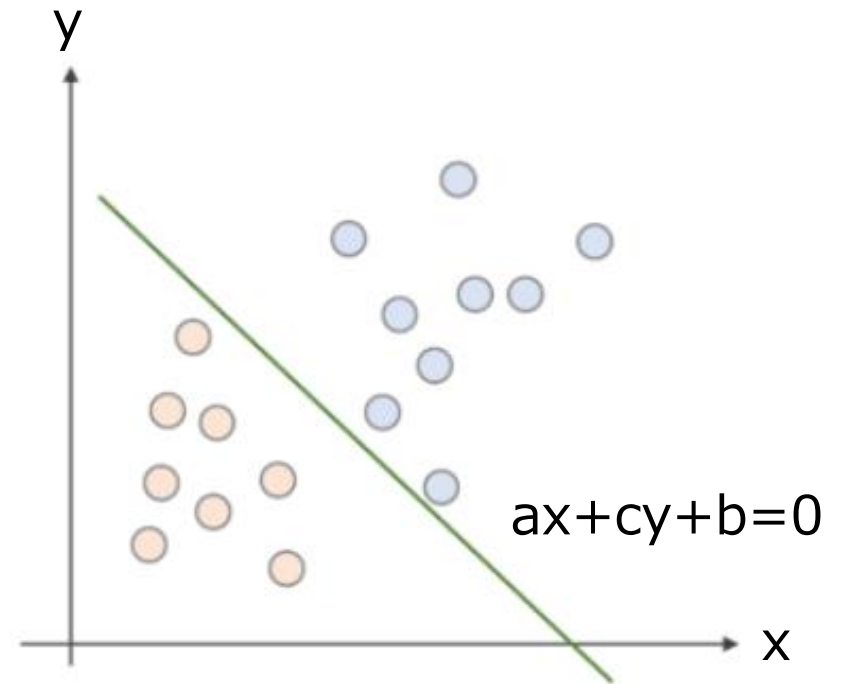
判別超平面

サポートベクターマシンでは、二次元データを直線($ax+cy+b=0$)で二つに分割する場合、マージンを最大化するように(a,b,c)を決定する！

一般的に、SVMではn次元ベクトル X と重みパラメータ W で線形結合した超平面を表す

$$W^t X + b = 0$$

において、パラメータ W をマージンが最大となるように決定する！



マージンの算出

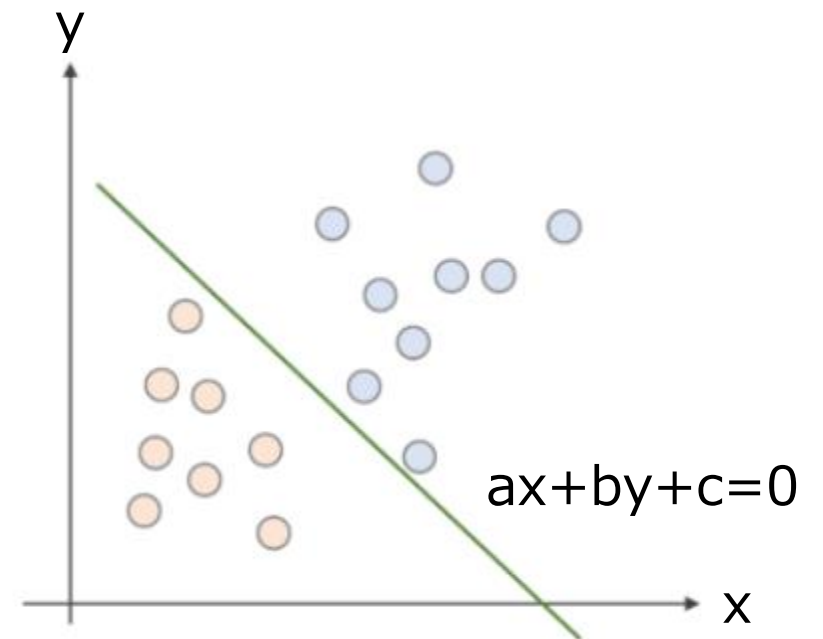
点 (x_0, y_0) と直線： $ax+by+c=0$ の距離

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

n次元の点の X_0 と超平面： $W^t X + b = 0$ の距離

$$d = \frac{|w_1x_1 + w_2x_2 + \cdots + w_nx_n + b|}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}}$$

$$= \frac{|W^t X_0 + b|}{\|w\|}$$



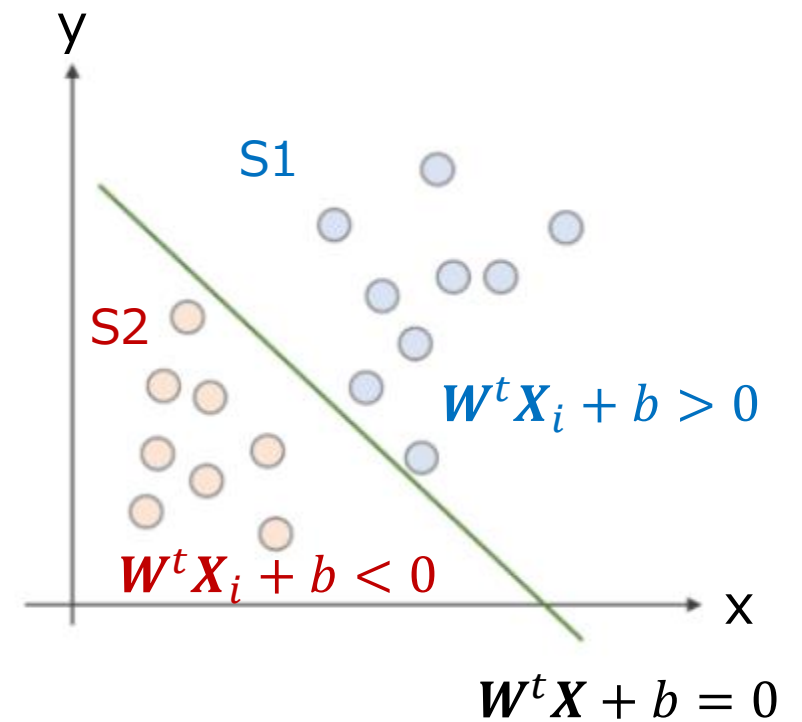
最適解に対する定式化

分離超平面によって、入力データがS1とS2に分離される場合、S1に+1およびS2に-1となるラベル(t_i)を与える。

$$t_i = \begin{cases} +1 & \text{S1: } \mathbf{W}^t \mathbf{X}_i + b > 0 \\ -1 & \text{S2: } \mathbf{W}^t \mathbf{X}_i + b < 0 \end{cases}$$

これにより、与えられたデータに対して以下満たす w, b, M を求めることに帰着する。

$$\max_{\mathbf{w}, b, M} \frac{t_i |\mathbf{W}^t \mathbf{X}_i + b|}{\|\mathbf{w}\|} \geq M$$

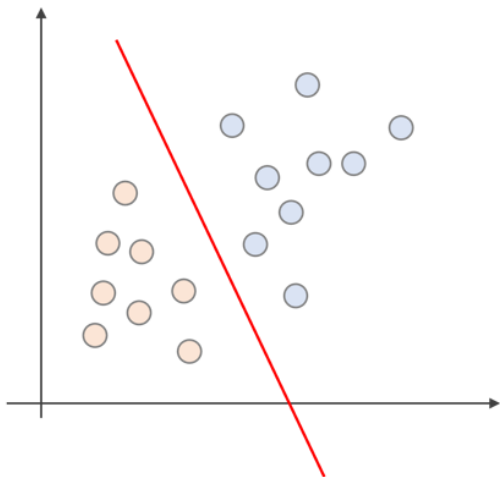


ハードマージンの定式化

ハードマージンとはサンプルが分離超平面によって完全に2つに分けられるときのみ

$$\max_{\mathbf{w}, b, M} \frac{t_i |\mathbf{W}^t \mathbf{X}_i + b|}{\|\mathbf{w}\|} \geq M$$

※ 等号が成立するのはサンプルがサポートベクターのときのみ

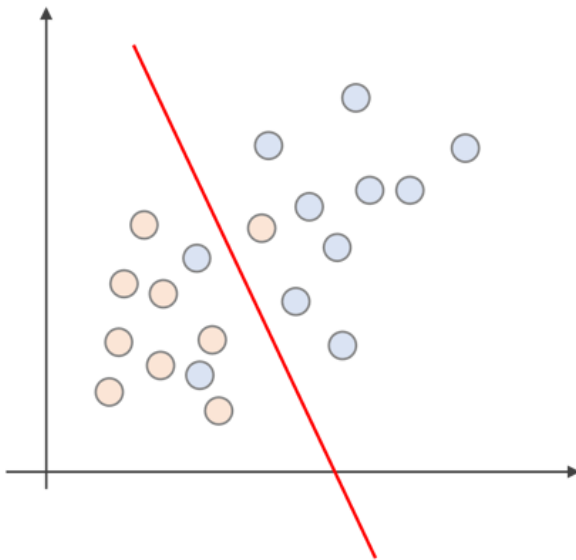


$$\min_{\mathbf{w}, b, M} \frac{1}{2} \|\mathbf{W}\|^2$$

$$t_i |\mathbf{W}^t \mathbf{X}_i + b| \geq 1 \quad (i = 1, 2, \dots, N)$$

ソフトマージンの定式化

ソフトマージンでは、サンプルが分離超平面によって完全に分離されるのではなく、一部マージンの内側に入り込むことを許容する




$$t_i |W^t X_i + b| \geq 1 - \xi_i$$

$$\xi_i = \max\{0, M - \frac{t_i |W^t X_i + b|}{\|w\|}\}$$

これにより、データがマージンの内側にあることが許容され、ハードマージンの際の制約が緩和される

ソフトマージン最適化問題

ソフトマージンでは、サンプルが分離超平面によって完全に分離されるのではなく、一部マージンの内側に入り込むことを許容する

$$\min_{\mathbf{W}, \xi} \left\{ \frac{1}{2} \|\mathbf{W}\|^2 + C \sum_{i=1}^N \xi_i \right\}$$


$$\text{制約条件} \quad t_i |\mathbf{W}^t \mathbf{X}_i + b| \geq 1 - \xi_i$$

$$\xi_i = \max\left\{0, M - \frac{t_i |\mathbf{W}^t \mathbf{X}_i + b|}{\|\mathbf{W}\|}\right\}$$

$\|\mathbf{W}\|^2$ を小さくすると $\sum_{i=1}^N \xi_i$ が大きくなり、
これらはトレードオフの関係にある

※ Cハイパーパラメータと呼ばれ、調整しながらモデル構築する

ラグランジュの未定乗数法による解法

ソフトマージンの問題をラグランジュの未定乗数法によって解を算出する

$$\min_{\mathbf{W}, \xi} \left\{ \frac{1}{2} \|\mathbf{W}\|^2 + C \sum_{i=1}^N \xi_i \right\} \quad \text{制約条件 (N個の不等式)}$$
$$t_i |\mathbf{W}^T \mathbf{X}_i + b| \geq 1 - \xi_i \quad (i = 1, 2, \dots, N)$$

ラグランジュ関数を以下の様に定義して

$$L(W, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{W}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i \{t_i (\mathbf{W}^T \mathbf{X}_i + b) - 1 + \xi_i\} - \sum_{i=1}^N \beta_i \xi_i$$

※ 右記で α を求める問題に帰着

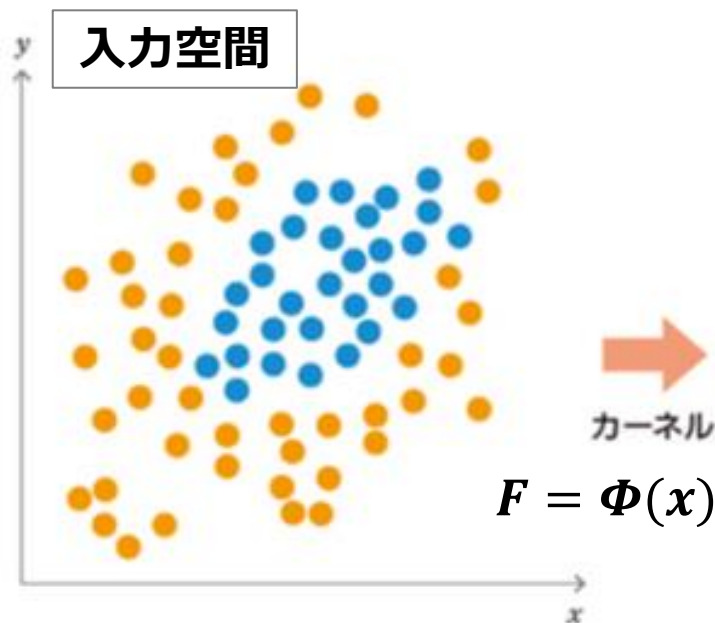


$$\max \left\{ \tilde{L}(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \mathbf{X}_i^T \mathbf{X}_j \right\}$$
$$\sum_{i=1}^N \alpha_i t_i = 0, \quad 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N$$

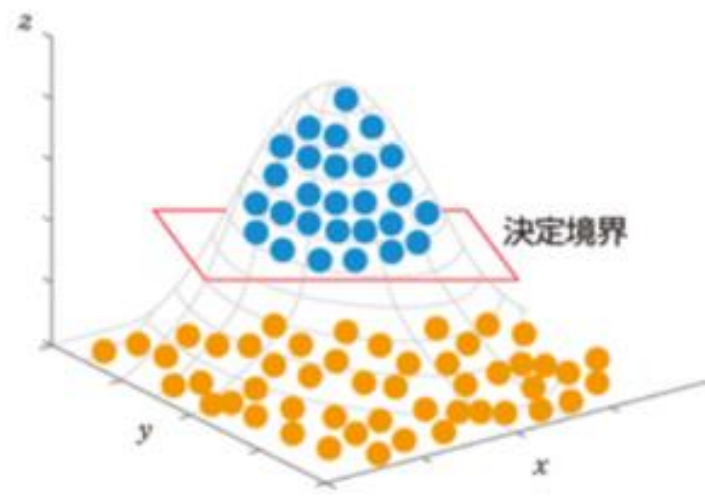
カーネル法

カーネル法は、元の情報空間では非線形判別でないと対応できないものを、カーネル関数を用いて入力空間を特徴空間に写像して線形判別を可能とする。

2次元では線形判別ができない



特徴空間



3次元では線形判別ができる

高次元特徴空間への写像例

写像の例)

入力空間

特徴空間

$$X = (x_1, x_2) \longrightarrow \psi(X) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

入力空間におけるベクトル X, Y に対して、写像変換後の $\psi(X), \psi(Y)$ の内積は写像変換さえ分かっているならば、元データの内積を計算するだけで求めることができる！

$$\begin{aligned}\psi(X) &= \psi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \\ \psi(Y) &= \psi(y_1, y_2) = (y_1^2, \sqrt{2}y_1y_2, y_2^2)\end{aligned}$$



$$\begin{aligned}\psi(X)^T \psi(Y) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T (y_1^2, \sqrt{2}y_1y_2, y_2^2) \\ &= x_1^2y_1^2 + 2x_1y_1x_2y_2 + x_2^2y_2^2 \\ &= (x_1y_1 + x_2y_2)^2 \\ &= ((x_1, x_2)^T (y_1, y_2))^2 \\ &= (X^T Y)^2\end{aligned}$$

代表的な 3 つのカーネル関数

入力空間におけるベクトル X, Y に対して、写像変換後の $\phi(X), \phi(Y)$ の内積は写像変換さえ分かっているならば、元データの内積を計算するだけで求めることができる！

- ・ 多項式カーネル $K(x_i, x_j) = (x_i^T x_j + c)^d$

- ・ ガウスカーネル $K(x_i, x_j) = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$

- ・ シグモイドカーネル $K(x_i, x_j) = \tanh(bx_i^T x_j + c)$

SPAMデータの読み込み

① Rstudio起動する

② `> library(kernlab)` ※コマンドラインから Rコマンダー を起動する

③ 演習ファイル “spam.csv” を読み込む

- Rstudio `> Dataset<-read.csv(“spam.csv”)`

又は

- Rコマンダー (データ) → (データインポート) → (テキストファイルまたはクリップボード...) →

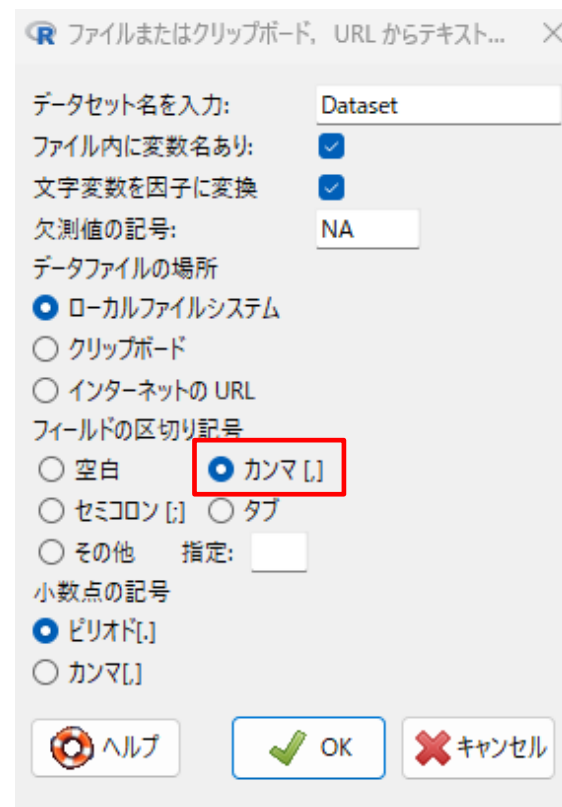
✓ OKを選択して、spam.csv を指定する

又は

- Rstudio `> data(spam)`

`Rstudio > Dataset <- spam`

④ 演習データが Dataset に読み込まれる



SPAMデータの特徴量

実際のSMSメッセージを57項目の特徴量で定量化したデータを入力とする

> head(spam)

| | make | address | all | num3d | our | over | remove | internet | order | mail | receive | will | people | report | addresses | free | business | email | you | credit | your | font | num000 |
|---|------|---------|------|-------|------|------|--------|----------|-------|------|---------|------|--------|--------|-----------|------|----------|-------|------|--------|------|------|--------|
| 1 | 0.00 | 0.64 | 0.64 | 0 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.64 | 0.00 | 0.00 | 0.00 | 0.32 | 0.00 | 1.29 | 1.93 | 0.00 | 0.96 | 0 | 0.00 |
| 2 | 0.21 | 0.28 | 0.50 | 0 | 0.14 | 0.28 | 0.21 | 0.07 | 0.00 | 0.94 | 0.21 | 0.79 | 0.65 | 0.21 | 0.14 | 0.14 | 0.07 | 0.28 | 3.47 | 0.00 | 1.59 | 0 | 0.43 |
| 3 | 0.06 | 0.00 | 0.71 | 0 | 1.23 | 0.19 | 0.19 | 0.12 | 0.64 | 0.25 | 0.38 | 0.45 | 0.12 | 0.00 | 1.75 | 0.06 | 0.06 | 1.03 | 1.36 | 0.32 | 0.51 | 0 | 1.16 |
| 4 | 0.00 | 0.00 | 0.00 | 0 | 0.63 | 0.00 | 0.31 | 0.63 | 0.31 | 0.63 | 0.31 | 0.31 | 0.31 | 0.00 | 0.00 | 0.31 | 0.00 | 0.00 | 3.18 | 0.00 | 0.31 | 0 | 0.00 |
| 5 | 0.00 | 0.00 | 0.00 | 0 | 0.63 | 0.00 | 0.31 | 0.63 | 0.31 | 0.63 | 0.31 | 0.31 | 0.31 | 0.00 | 0.00 | 0.31 | 0.00 | 0.00 | 3.18 | 0.00 | 0.31 | 0 | 0.00 |
| 6 | 0.00 | 0.00 | 0.00 | 0 | 1.85 | 0.00 | 0.00 | 1.85 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0 | 0.00 |

| | money | hp | hpl | george | num650 | lab | labs | telnet | num857 | data | num415 | num85 | technology | num1999 | parts | pm | direct | cs | meeting | original | project | re | edu | table |
|---|-------|----|-----|--------|--------|-----|------|--------|--------|------|--------|-------|------------|---------|-------|----|--------|----|---------|----------|---------|------|------|-------|
| 1 | 0.00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0 |
| 2 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.07 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0 |
| 3 | 0.06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0 | 0 | 0.06 | 0 | 0 | 0.12 | 0 | 0.06 | 0.06 | 0 |
| 4 | 0.00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0 |
| 5 | 0.00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0 |
| 6 | 0.00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0 |

| | conference | charsemicolon | charRoundbracket | charsquarebracket | charExclamation | charDollar | charHash | capitalAve | capitalLong | capitalTotal | type |
|---|------------|---------------|------------------|-------------------|-----------------|------------|----------|------------|-------------|--------------|------|
| 1 | 0 | 0.00 | 0.000 | 0 | 0.778 | 0.000 | 0.000 | 3.756 | 61 | 278 | spam |
| 2 | 0 | 0.00 | 0.132 | 0 | 0.372 | 0.180 | 0.048 | 5.114 | 101 | 1028 | spam |
| 3 | 0 | 0.01 | 0.143 | 0 | 0.276 | 0.184 | 0.010 | 9.821 | 485 | 2259 | spam |
| 4 | 0 | 0.00 | 0.137 | 0 | 0.137 | 0.000 | 0.000 | 3.537 | 40 | 191 | spam |
| 5 | 0 | 0.00 | 0.135 | 0 | 0.135 | 0.000 | 0.000 | 3.537 | 40 | 191 | spam |
| 6 | 0 | 0.00 | 0.223 | 0 | 0.000 | 0.000 | 0.000 | 3.000 | 15 | 54 | spam |

“正常メール”



VS.



“スパムメール”のラベル



SVMの実行

※ SVMには"**kernlab**"パッケージのインストールが必須 ← `> install.packages("kernlab")`

```
> library(kernlab)
> tr.num <- sample(4601,2500)
> spam.train<-Dataset[tr.num,]
> spam.test<-Dataset[-tr.num,]
> spam.svm<-ksvm(type~.,data=spam.train,cross=3)
> spam.svm
Support Vector Machine object of class "ksvm"
```

交差確認（クロスバリデーション）では、
学習データの一部を用いて学習する
※ここでは3回に分けて実施「

```
SV type: C-svc (classification)
parameter : cost C = 1
```

ハイパーパラメーター

```
Gaussian Radial Basis kernel function.
Hyperparameter : sigma = 0.0266105027057991
```

ガウスカーネルを採用してSVM

```
Number of Support Vectors : 847
```

サポートベクトルの数

```
Objective Function value : -472.4107
Training error : 0.0436
Cross validation error : 0.074402
```

学習データでの**誤班別率**

SVMの学習結果をを評価データに適用

```
> spam.pre<-predict(spam.svm,spam.test[, -58])
> spam.tab<-table(spam.test[,58],spam.pre)
> spam.tab
```

| | spam. pre | |
|---------|-----------|------|
| | nonspam | spam |
| nonspam | 1237 | 63 |
| spam | 98 | 703 |

```
> 1-sum(diag(spam.tab))/sum(spam.tab)
[1] 0.07663018
```

評価データに対する誤判別率：7.66 %



vs.



ナীবベイズ

```
> p2<- h2o.predict(tr2,spam.test.hex)
> p20<- as.data.frame(p2)
> table(p20[,1],spam.test[,58])
```

検証データの適用

| | nonspam | spam |
|---------|---------|------|
| nonspam | 1172 | 112 |
| spam | 223 | 794 |

全体の誤り率は14.6%

ニューラルネットワーク

Confusion Matrix (vertical: actual; across:

| | nonspam | spam | Error | Rate |
|---------|---------|------|----------|----------|
| nonspam | 1380 | 13 | 0.009332 | =13/1393 |
| spam | 17 | 890 | 0.018743 | =17/907 |
| Totals | 1397 | 903 | 0.013043 | =30/2300 |

全体の誤り率は1%程度



XGBoost

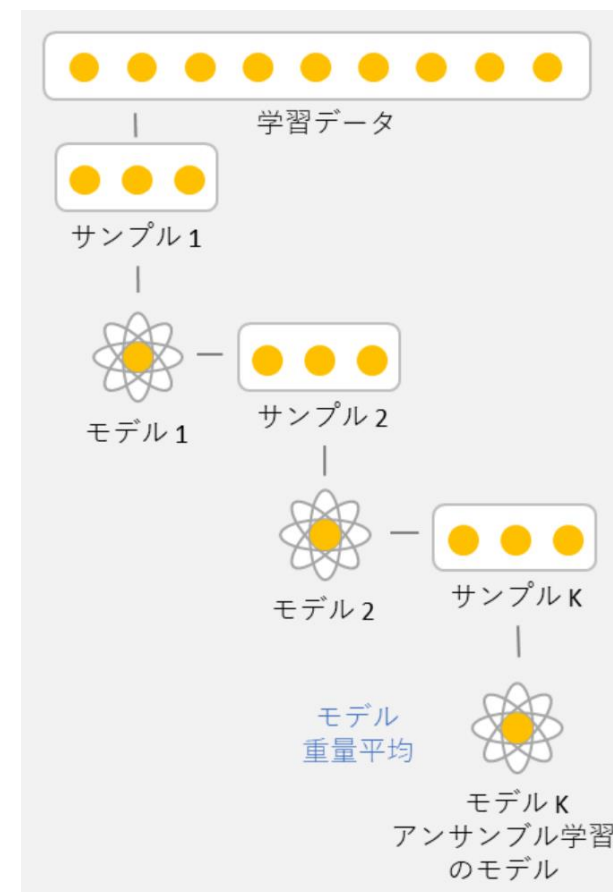


アンサンブル学習（再掲）

| | 平均投票 Max Voting | 重量平均投票 Weighted Average Voting | バギング Bagging | ブースティング Boosting | スタッキング Stacking | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|--|-----------------------------------|-----------------|---------------------|--------------------|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|
| モデルの構成 | <p>平均投票 Max Voting</p> <p>学習データ</p> <p>モデル1 モデル2 モデルK</p> <p>最大</p> <p>アンサンブル学習のモデル</p> <table><tr><td>モデル</td><td>1</td><td>2</td><td>3</td><td>融合</td></tr><tr><td>A</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>B</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>C</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table> | モデル | 1 | 2 | 3 | 融合 | A | ○ | ○ | ○ | ○ | B | ○ | ○ | ○ | ○ | C | ○ | ○ | ○ | ○ | <p>重量平均投票 Weighted Average Voting</p> <p>学習データ</p> <p>モデル1 モデル2 モデルK</p> <p>重量平均</p> <p>アンサンブル学習のモデル</p> <table><tr><td>モデル</td><td>1</td><td>2</td><td>3</td><td>融合</td></tr><tr><td>A</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>B</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>C</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table> <p>重量平均 3 1 1</p> | モデル | 1 | 2 | 3 | 融合 | A | ○ | ○ | ○ | ○ | B | ○ | ○ | ○ | ○ | C | ○ | ○ | ○ | ○ | <p>バギングアンサンブル Bagging Ensemble</p> <p>学習データ</p> <p>サンプル1 サンプル2 サンプルK</p> <p>モデル1 モデル2 モデルK</p> <p>平均</p> <p>アンサンブル学習のモデル</p> | <p>ブースティングアンサンブル Boosting Ensemble</p> <p>学習データ</p> <p>サンプル1</p> <p>モデル1</p> <p>サンプル2</p> <p>モデル2</p> <p>サンプルK</p> <p>モデルK</p> <p>重量平均</p> <p>アンサンブル学習のモデル</p> | <p>スタッキングアンサンブル Stacking Ensemble</p> <p>学習データ</p> <p>モデル1 モデル2 モデルK</p> <p>線形結合</p> <p>アンサンブル学習のモデル</p> |
| モデル | 1 | 2 | 3 | 融合 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| モデル | 1 | 2 | 3 | 融合 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 複数サンプル | × | × | ○ | ○ | × | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 複数モデル | ○ | ○ | ○ | ○ | ○ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| モデル作成方法 | 平行 | 平行 | 平行 | 階段 | 平行 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 結果の融合方法 | 平行 | 重量平均 | 平均 | 重量平均 | 線形結合 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| バイアスとバリエーションの エラー処理 | | | バリエーション | | バイアス バリエーション | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

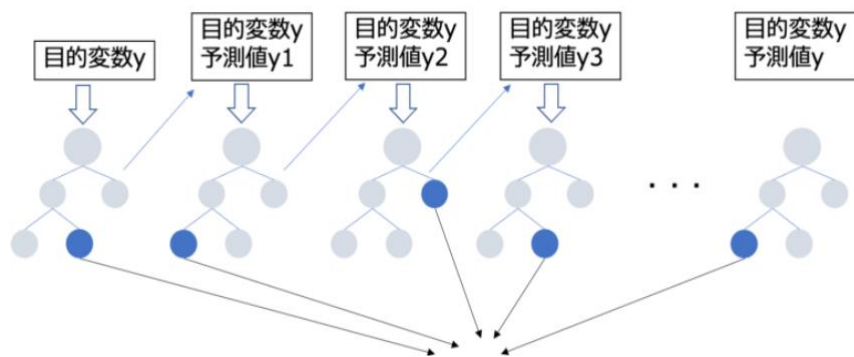
ブースティング（再掲）

- 学習データからランダム抽出したデータセットを用いて、最初のモデル(“弱学習器”)構築する
- 構築された弱学習器が誤った部分に重みをかけて、次の弱学習器を構築する
- ブースティングを用いた手法として、**XGBoost**や**勾配ブースティング**などがある
- 直列計算のため、バギングよりも時間がかかる



勾配ブースティング

勾配ブースティングでは、モデル全体の予測値と目的変数との差分を次に決定木学習に用いる。



- 決定木を構築し予想値を算出、目的変数の差分を算出する
- 予測値と目的変数の誤差が小さくなるように決定木を構築する
- 指定した決定木の本数分、上述の算出と構築を繰り返す

⇒ 各決定木で属する葉のウェイトの和が最終的な予測値とする

$$l_{mi} = (y_{mi} - h_{mi})^2, l_i = \frac{1}{M} \sum_{m=1}^M l_{mi}$$

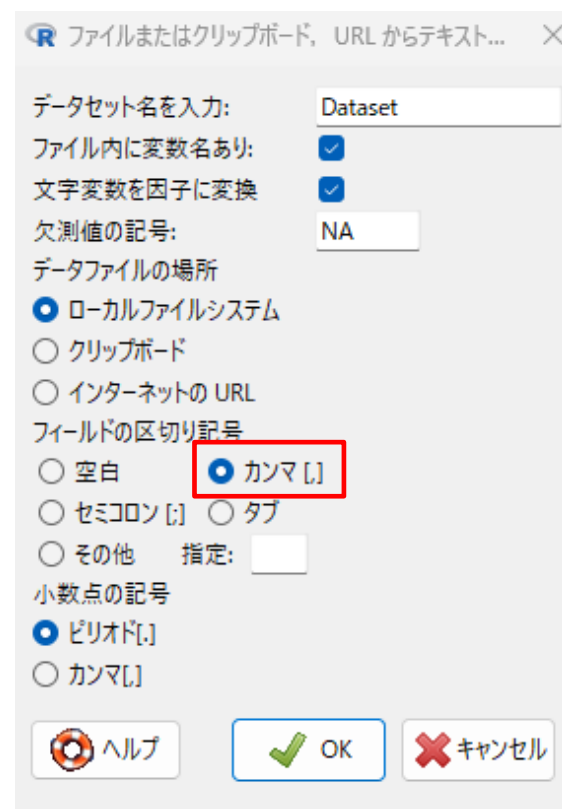
$$h_{i+1} = h_i + \eta \frac{\partial l_i}{\partial y} = h_i + 2\eta \sum_{m=1}^M (y_{mi} - h_{mi})$$



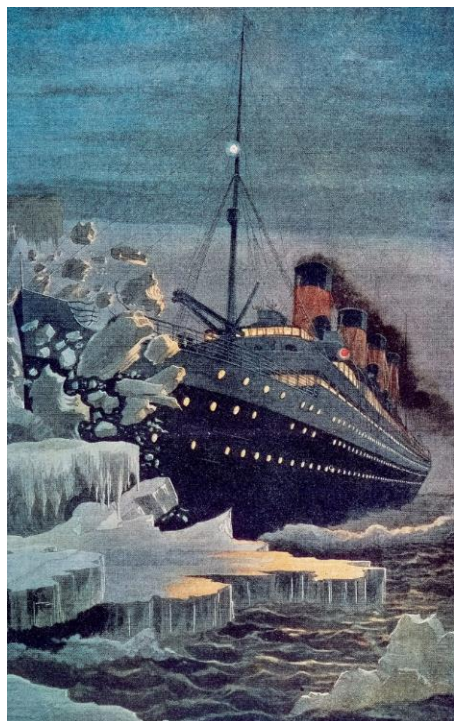
XGBoost, LightGBM, CatGBM


タイタニック号事故データの読み込み

- ① Rstudio起動する
- ② `> library(Rcmdr)` ※コマンドラインから Rコマンダー を起動する
- ③ 演習ファイル “titanic.csv” を読み込む
 - Rstudio `> Dataset<-read.csv(“titanic.csv”)`
又は
 - Rコマンダー (データ) → (データインポート) → (テキストファイルまたはクリップボード...) →
✓ OKを選択して、titanic.csv を指定する
- ④ 演習データが Dataset に読み込まれる



タイタニック号事故データ





| | 生存/非生存 | 席等級 | 兄弟/配偶者 | | | 親子 | チケット番号 | 価格 | キャビン番号 | | 寄港地 | |
|------|-------------|----------|----------|--------------|-----|-----|--------|---------|--------|------|-------|----------|
| | | | 名前・性別・年齢 | | | | | | | | | |
| 乗客番号 | PassengerId | Survived | Pclass | Name | Sex | Age | Sibsp | Parch | Ticket | Fare | Cabin | Embarked |
| | 1 | 0 | 3 | NA male 22 | 1 | 0 | NA | 7.2500 | | | | S |
| | 2 | 1 | 1 | NA female 38 | 1 | 0 | NA | 71.2833 | | | C85 | C |
| | 3 | 1 | 3 | NA female 26 | 0 | 0 | NA | 7.9250 | | | | S |
| | 4 | 1 | 1 | NA female 35 | 1 | 0 | 113803 | 53.1000 | | | C123 | S |
| | 5 | 0 | 3 | NA male 35 | 0 | 0 | 373450 | 8.0500 | | | | S |
| | 6 | 0 | 3 | NA male NA | 0 | 0 | 330877 | 8.4583 | | | | Q |
| | 7 | 0 | 1 | NA male 54 | 0 | 0 | 17463 | 51.8625 | | | E46 | S |
| | 8 | 0 | 3 | NA male 2 | 3 | 1 | 349909 | 21.0750 | | | | S |
| | 9 | 1 | 3 | NA female 27 | 0 | 2 | 347742 | 11.1333 | | | | S |
| | 10 | 1 | 2 | NA female 14 | 1 | 0 | 237736 | 30.0708 | | | | C |
| | 11 | 1 | 3 | NA female 4 | 1 | 1 | NA | 16.7000 | | | G6 | S |
| | 12 | 1 | 1 | NA female 58 | 0 | 0 | 113783 | 26.5500 | | | C103 | S |
| | 13 | 0 | 3 | NA male 20 | 0 | 0 | NA | 8.0500 | | | | S |
| | 14 | 0 | 3 | NA male 39 | 1 | 5 | 347082 | 31.2750 | | | | S |
| | 15 | 0 | 3 | NA female 14 | 0 | 0 | 350406 | 7.8542 | | | | S |
| | 16 | 1 | 2 | NA female 55 | 0 | 0 | 248706 | 16.0000 | | | | S |
| | 17 | 0 | 3 | NA male 2 | 4 | 1 | 382652 | 29.1250 | | | | Q |
| | 18 | 1 | 2 | NA male NA | 0 | 0 | 244373 | 13.0000 | | | | S |
| | 19 | 0 | 3 | NA female 31 | 1 | 0 | 345763 | 18.0000 | | | | S |
| | 20 | 1 | 3 | NA female NA | 0 | 0 | 2649 | 7.2250 | | | | C |

XGBOOSTの実行 1

※ パッケージインストール > `install.packages("xgboost")`

```
library(xgboost)
titanic$Survived<-as.factor(titanic$Survived)
sim<-10
result<-matrix(0,sim)
titanic[,4]<-as.numeric(titanic[,4])
titanic[,9]<-as.numeric(titanic[,9])
```

交差確認（クロスバリデーション）では、
学習データの一部を用いて学習する
※ここでは10回(sim=10)実施

```
for(i in 1:sim){
  train.id<-sample(nrow(titanic),400)
  train.data<-titanic[train.id,-1]
```

学習データから400レコードをランダムにサンプルして用いる

```
test.data<-titanic[-train.id,-1]
label.data.train<-as.integer(train.data$Survived)-1
label.data.predict<-as.integer(test.data$Survived)-1
```

"Survived"（目的変数）のラベリング（1 / 0）

```
train.data.xg<-train.data[,-1]
test.data.xg<-test.data[,-1]
```

XGBoostモデルへの入力データ準備

XGBOOSTの実行 2

XGBoost入力データ形成

```
xgb.data<-xgb.DMatrix(data.matrix(train.data.xg),label=label.data.train)
xgb.data.predict<-xgb.DMatrix(data.matrix(test.data.xg))
```

```
param <- list("objective" = "binary:logistic","eta" = 0.01,"min_child_weight" = 5)
model <- xgboost(param=param,data=xgb.data,nrounds=1000)
predict_xgb<-predict(model,xgb.data.predict)
```

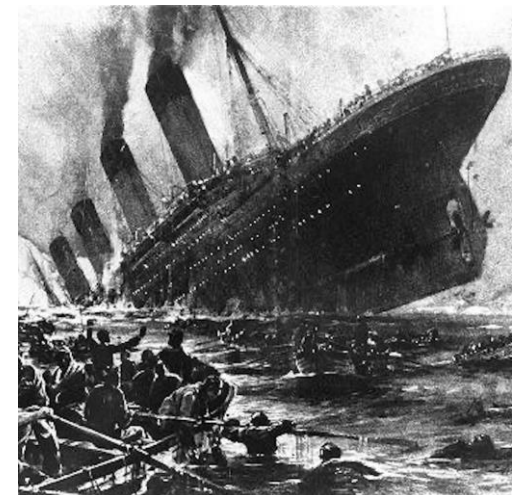
XGBoostモデル

```
a<-c()
m<-length(predict_xgb)
```

```
for(l in 1:m){
  if(predict_xgb[l]>0.5){a[l]<-1}
  else{a[l]<-0} }
```

テストデータ入力に対して、出力結果は
0 ~ 1 の確率として算出されるため、
0.5以上で生存または非生存をする

```
result[i]<-sum(a==label.data.predict)/length(a)
}
mean(result)
```



課題：SPAMデータを用いた XGBoost

タイタニック号事故データで演習したXGBoostモデルをSPAMデータに対して構築し、SVMの結果と比較しなさい。

XGBOOSTの実行 1

```
library(xgboost)
titanic$Survived<-as.factor(titanic$Survived)
sim<-10
result<-matrix(0,sim)
titanic[,4]<-as.numeric(titanic[,4])
titanic[,9]<-as.numeric(titanic[,9])

for(i in 1:sim){
  train.id<-sample(nrow(titanic),400)
  train.data<-titanic[train.id,-1]

  test.data<-titanic[-train.id,-1]
  label.data.train<-as.integer(train.data$Survived)-1
  label.data.predict<-as.integer(test.data$Survived)-1

  train.data.xg<-train.data[,1]
  test.data.xg<-test.data[,1]
```

※ パッケージインストール > `install.packages("xgboost")`

交差確認 (クロスバリデーション) では、学習データの一部分を用いて学習する
※ここでは10回(sim=10)実施

学習データから400レコードをランダムにサンプルして用いる

“Survived” (目的変数) のラベリング (1/0)

XGBoostモデルへの入力データ準備

XGBOOSTの実行 2

```
xgb.data<-xgb.DMatrix(data.matrix(train.data.xg),label=label.data.train)
xgb.data.predict<-xgb.DMatrix(data.matrix(test.data.xg))

param <- list("objective" = "binary:logistic","eta" = 0.01,"min_child_weight" = 5)
model <- xgboost(param=param,data=xgb.data,nrounds=1000)
predict_xgb<-predict(model,xgb.data.predict)

a<-c()
m<-length(predict_xgb)

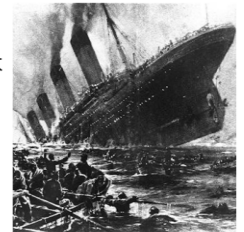
for(l in 1:m){
  if(predict_xgb[l]>0.5){a[l]<-1}
  else{a[l]<-0} }

result[i]<-sum(a==label.data.predict)/length(a)
}
mean(result)
```

XGBoost入力データ形成

XGBoostモデル

テストデータ入力に対して、出力結果は0～1の確率として算出されるため、0.5以上で生存または非生存をする





データマイニングを楽しもう！