機械学習
練習問題①
回答例

管理工学科
篠沢佳久

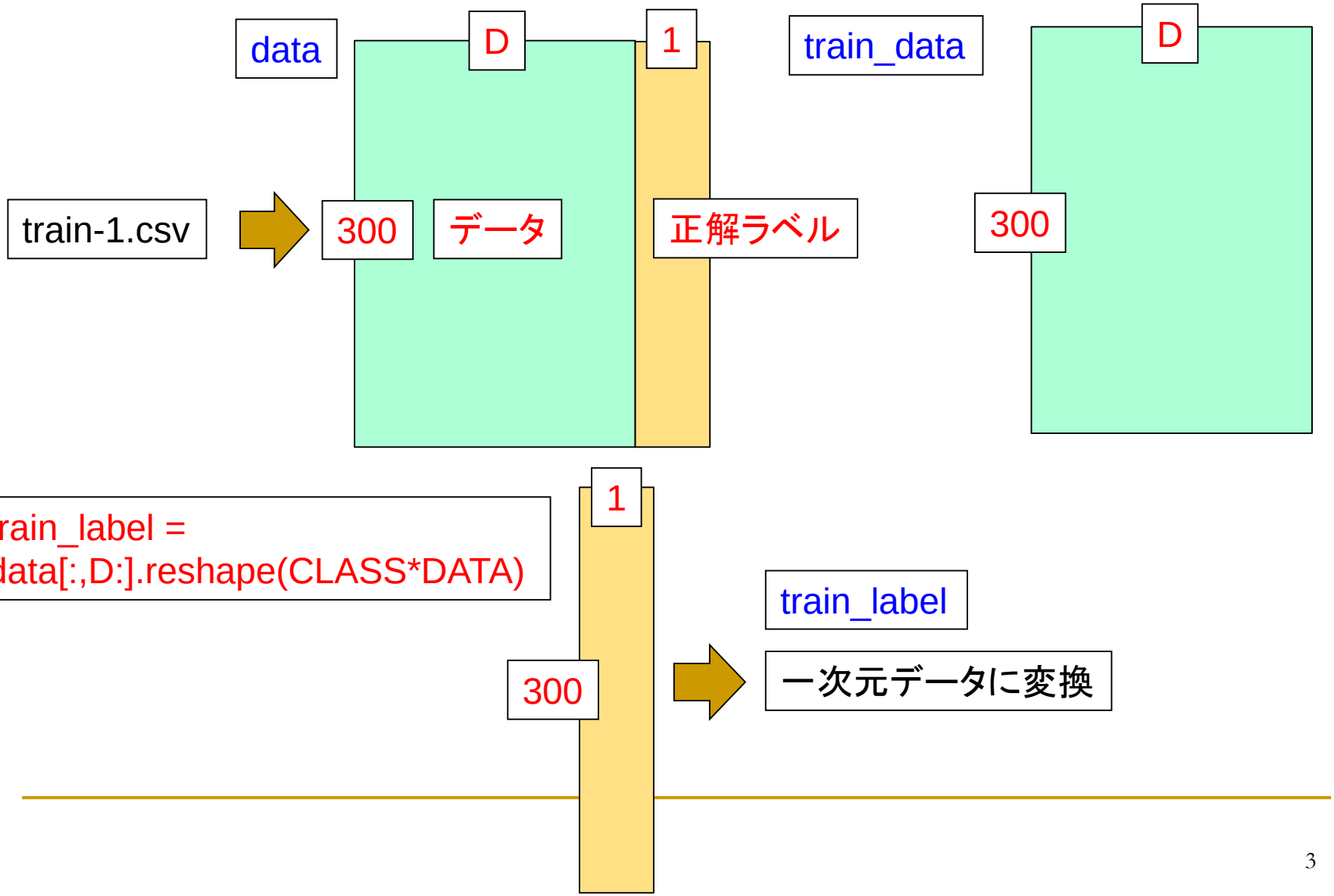
```
# ロジスティック回帰
import sys
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix, precision_score, recall_score, f1_score
from sklearn.linear_model import LogisticRegression

CLASS = 3     # クラス数
DATA = 100    # クラスごとのデータ数
D = 4         #  特徴量の次元数

# 学習データ
train_file = "train-1.csv"
data = np.loadtxt(train_file,delimiter=",")
train_data = data[:,0:D]
train_label = data[:,D:].reshape(CLASS*DATA)
```

緑色の箇所が Cancer_logistic.py と異なる点

区切り文字は「，」

data = np.loadtxt(train_file,delimiter=",")

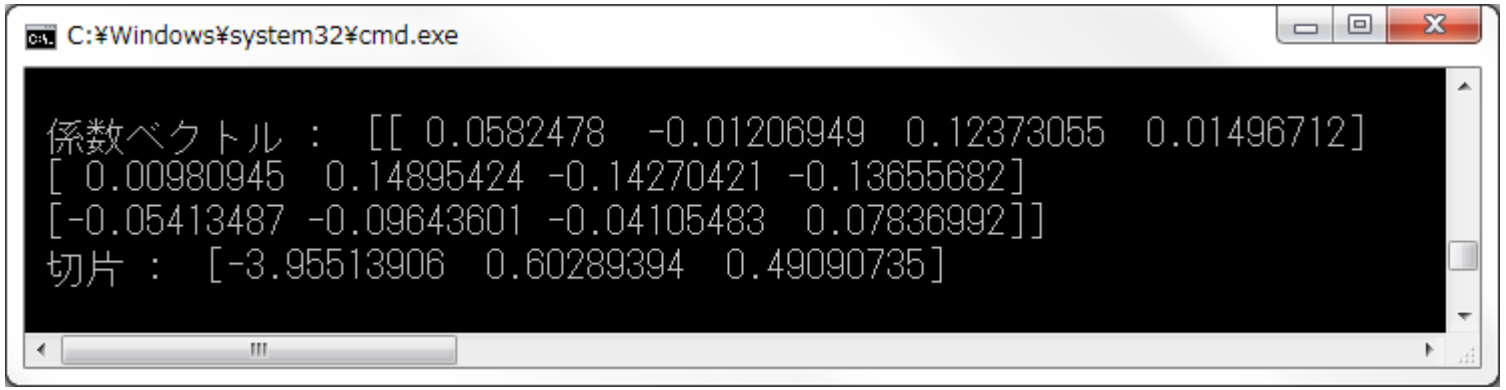
train_data = data[:,0:D]



```
# テストデータ
test_file = "test-1.csv"
data = np.loadtxt(test_file,delimiter=",")
test_data = data[:,0:4]
test_label = data[:,4:].reshape(CLASS*DATA)

# ロジスティック回帰
model = LogisticRegression(C=1.0,penalty='l2',solver='lbfgs',max_iter=100)

# 学習
model.fit(train_data, train_label)

# 予測
predict = model.predict(test_data)

# 係数と切片
print( '¥n 係数ベクトル : ' , model.coef_ )
print( ' 切片 : ' , model.intercept_)
```

```
# 予測値，教師ラベル
print( '¥n [ 予測値 ： 教師ラベル ]' )
predict_proba = model.predict_proba(test_data)
for i in range(len(test_label)):
    print( predict_proba[i] , ':' , predict[i] , "(" , test_label[i] , ")" )

# 予測結果の表示
print( "¥n [ 予測結果 ]" )
print( ' accuracy  : ' , accuracy_score(test_label, predict) )
print( ' precision : ' , precision_score(test_label, predict, average='micro') )
print( ' recall    : ' , recall_score(test_label, predict, average='micro') )
print( ' f1-score  : ' , f1_score(test_label, predict, average='micro') )

print( "¥n [ 予測結果 ]" )
print( classification_report(test_label, predict) )

print( "¥n [ 混同行列 ]" )
print( confusion_matrix(test_label, predict) )
```

実行結果



クラス1

$$y_1 = \frac{1}{1+\exp(-(-3.955+0.0582x_1-0.012x_2+0.1237x_3+0.0149x_4))}$$

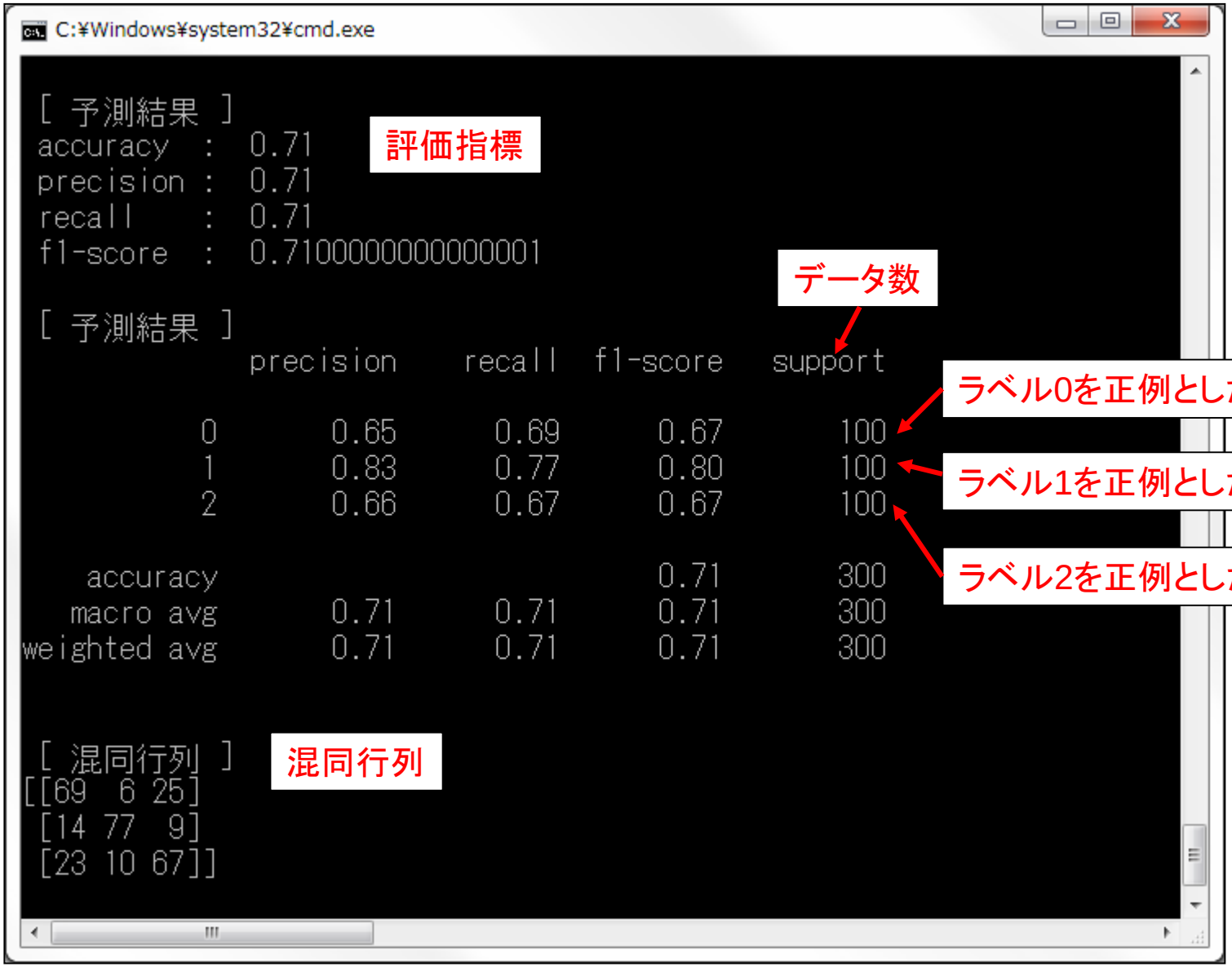
クラス2

$$y_2 = \frac{1}{1+\exp(-(0.6028+0.0098x_1+0.1489x_2-0.1427x_3-0.1365x_4))}$$

クラス3

$$y_3 = \frac{1}{1+\exp(-(0.4909-0.0541x_1-0.0964x_2-0.041x_3-0.0783x_4))}$$


C:¥Windows¥system32¥cmd.exe
予測クラス
正解クラス
[予測値 : 教師ラベル]
[0.66738151 0.05653821 0.27608028] : 0 (0)
[0.82971773 0.08720645 0.08307583] : 0 (0)
[0.56055266 0.05973294 0.3797144] : 0 (0)
[0.58312721 0.00362597 0.41324682] : 0 (0)
[0.38674115 0.00862248 0.60463638] : 2 (0)
[0.52613765 0.04015424 0.43370811] : 0 (0)
[0.36690359 0.2363116 0.39678481] : 2 (0)
[0.31178634 0.02938914 0.65882452] : 2 (0)
[0.25423944 0.02059986 0.72516071] : 2 (0)
[0.26557822 0.70307611 0.03134567] : 1 (0)
[0.72399897 0.0689844 0.20701663] : 0 (0)
[0.86211128 0.04544386 0.09244487] : 0 (0)
[0.77514574 0.15569803 0.06915622] : 0 (0)
[0.94234218 0.00097945 0.05667838] : 0 (0)
[0.91336355 0.0011908 0.08544564] : 0 (0)
[0.75668862 0.00083901 0.24247238] : 0 (0)
[0.82845906 0.04971979 0.12182115] : 0 (0)
クラス1の予測確率
クラス2の予測確率
クラス3の予測確率

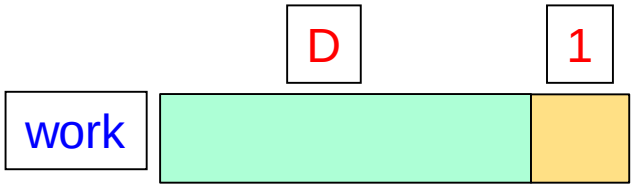

C:¥Windows¥system32¥cmd.exe
[予測結果]
accuracy : 0.71
precision : 0.71
recall : 0.71
f1-score : 0.7100000000000001
評価指標
[予測結果]
precision recall f1-score support
データ数
0 0.65 0.69 0.67 100
1 0.83 0.77 0.80 100
2 0.66 0.67 0.67 100
accuracy 0.71 300
macro avg 0.71 0.71 0.71 300
weighted avg 0.71 0.71 0.71 300
ラベル0を正例とした場合
ラベル1を正例とした場合
ラベル2を正例とした場合
[混同行列]
混同行列
[[69 6 25]
[14 77 9]
[23 10 67]]

データの読み込み①（ファイル）

```
# 学習データの読み込み
train_file = "train-1.csv"
train_data = np.zeros( (CLASS*DATA,D) , dtype=np.float32 )
train_label = np.zeros( CLASS*DATA , dtype=np.int32 )

count = 0
f = open( train_file , "r" )
for line in f:
    work = line.strip().split( ',' )
    train_data[count] = np.array( work[0:D] , dtype=np.float32 )
    train_label[count] = int( work[D] )
    count += 1
f.close()
```

ファイルのオープン

改行削除→「,」で区切る



ファイルを閉じる

データの読み込み②（ファイル）

```
# テストデータの読み込み
test_file = "test-1.csv"
test_data = np.zeros( (CLASS*DATA,D) , dtype=np.float32 )
test_label = np.zeros( CLASS*DATA , dtype=np.int32 )

count = 0
f = open( test_file , "r" )
for line in f:
    work = line.strip().split( ',' )
    test_data[count] = np.array( work[0:D] , dtype=np.float32 )
    test_label[count] = int( work[D] )
    count += 1
f.close()
```

データの読み込みの工夫①

```
# 学習データ
train_file = "train-1.csv"
train_data = np.loadtxt(train_file,delimiter=",", usecols=(0,1,2,3))
train_label = np.loadtxt(train_file,delimiter=",", usecols=(4))

# テストデータ
test_file = "test-1.csv"
test_data = np.loadtxt(test_file,delimiter=",", usecols=(0,1,2,3))
test_label = np.loadtxt(test_file,delimiter=",", usecols=(4))
```

0，1，2，3列目の読み込み

4列目の読み込み

データの読み込みの工夫②

```
# 学習データ
train_file = "train-1.csv"
train_data = np.loadtxt(train_file,delimiter=",", usecols=(0,1,2,3))
train_label = np.loadtxt(train_file,delimiter=",", usecols=(4)).astype(np.int32)

# テストデータ
test_file = "test-1.csv"
test_data = np.loadtxt(test_file,delimiter=",", usecols=(0,1,2,3))
test_label = np.loadtxt(test_file,delimiter=",", usecols=(4)).astype(np.int32)
```

ラベルを整数型で読み込み