Linear SVM Classification

Import packages

In [1]:
```
import numpy as np
import pandas as pd
import xlrd
from sklearn import svm
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
import sys
!{sys.executable} -m pip install openpyxl
import warnings
warnings.filterwarnings('ignore')
warnings.simplefilter('ignore')
```

```
Requirement already satisfied: openpyxl in c:\users\okina\anaconda3\lib\site-packages (3.0.10)
Requirement already satisfied: et_xmlfile in c:\users\okina\anaconda3\lib\site-packages (from openpyxl)
(1.1.0)
```

In [2]:
```
new_df = pd.read_excel('Immunotherapy.xlsx')
```

In [3]:
```
new_df2 = pd.read_excel('Raisin_Dataset.xlsx')
```

Immunotherapy Dataset

In [4]:
```
def calc_error(X,Y, classifier):
    Y_pred = classifier.predict(X)
    accuracy = accuracy_score(Y, Y_pred)
    error = 1 - accuracy

    return error,accuracy
```

In [5]:
```
X = np.asarray(new_df[['sex', 'age', 'Time', 'Number_of_Warts', 'Type', 'Area',
       'induration_diameter']])

Y = np.asarray(new_df[['Result_of_Treatment']])

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, random_state = 5)

Cs = [0.1, 1, 10, 100, 1000]

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.20833333333333337
Accuracy: 0.7916666666666666
Cross Validation Score: 0.7916666666666666
C = 1
Training error: 0.20833333333333337
Accuracy: 0.7916666666666666
Cross Validation Score: 0.7777777777777777
C = 10
Training error: 0.19444444444444442
Accuracy: 0.8055555555555556
Cross Validation Score: 0.75
C = 100
Training error: 0.5972222222222222
Accuracy: 0.4027777777777778
Cross Validation Score: 0.6527777777777778
C = 1000
Training error: 0.125
Accuracy: 0.875
Cross Validation Score: 0.7361111111111112
```

In [6]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.20833333333333337
Accuracy: 0.7916666666666666
Cross Validation Score: 0.7916666666666666
C = 1
Training error: 0.18055555555555558
Accuracy: 0.8194444444444444
Cross Validation Score: 0.6666666666666666
C = 10
Training error: 0.18055555555555558
Accuracy: 0.8194444444444444
Cross Validation Score: 0.75
C = 100
Training error: 0.2222222222222222
Accuracy: 0.7777777777777778
Cross Validation Score: 0.75
C = 1000
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.75
```

In [7]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.20833333333333337
Accuracy: 0.7916666666666666
Cross Validation Score: 0.7916666666666666
C = 1
Training error: 0.20833333333333337
Accuracy: 0.7916666666666666
Cross Validation Score: 0.7777777777777777
C = 10
Training error: 0.19444444444444442
Accuracy: 0.8055555555555556
Cross Validation Score: 0.763888888888889
C = 100
Training error: 0.18055555555555558
Accuracy: 0.8194444444444444
Cross Validation Score: 0.736111111111111
C = 1000
Training error: 0.2222222222222222
Accuracy: 0.7777777777777778
Cross Validation Score: 0.6527777777777777
```

50/50 Train/Test Split

In [8]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.5, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.2222222222222222
Accuracy: 0.7777777777777778
Cross Validation Score: 0.7999999999999999
C = 1
Training error: 0.1333333333333333
Accuracy: 0.8666666666666667
Cross Validation Score: 0.7999999999999999
C = 10
Training error: 0.0888888888888889
Accuracy: 0.9111111111111111
Cross Validation Score: 0.8222222222222223
C = 100
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.7555555555555555
C = 1000
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.8222222222222223
```

In [9]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.2222222222222222
Accuracy: 0.7777777777777778
Cross Validation Score: 0.7999999999999999
C = 1
Training error: 0.19999999999999996
Accuracy: 0.8
Cross Validation Score: 0.7777777777777778
C = 10
Training error: 0.1333333333333333
Accuracy: 0.8666666666666667
Cross Validation Score: 0.7111111111111111
C = 100
Training error: 0.1333333333333333
Accuracy: 0.8666666666666667
Cross Validation Score: 0.7111111111111111
C = 1000
Training error: 0.0888888888888889
Accuracy: 0.9111111111111111
Cross Validation Score: 0.7999999999999999
```

In [10]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.2222222222222222
Accuracy: 0.7777777777777778
Cross Validation Score: 0.7999999999999999
C = 1
Training error: 0.1333333333333333
Accuracy: 0.8666666666666667
Cross Validation Score: 0.7999999999999999
C = 10
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.6888888888888888
C = 100
Training error: 0.0888888888888889
Accuracy: 0.9111111111111111
Cross Validation Score: 0.8444444444444444
C = 1000
Training error: 0.1333333333333333
Accuracy: 0.8666666666666667
Cross Validation Score: 0.7777777777777777
```

20/80 Train/Test Split

In [11]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.8, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.8333333333333334
C = 1
Training error: 0.05555555555555558
Accuracy: 0.9444444444444444
Cross Validation Score: 0.7777777777777778
C = 10
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.6111111111111112
C = 100
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.6111111111111112
C = 1000
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.6666666666666666
```

In [12]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.8333333333333334
C = 1
Training error: 0.05555555555555558
Accuracy: 0.9444444444444444
Cross Validation Score: 0.7777777777777778
C = 10
Training error: 0.05555555555555558
Accuracy: 0.9444444444444444
Cross Validation Score: 0.6111111111111112
C = 100
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.6666666666666666
C = 1000
Training error: 0.05555555555555558
Accuracy: 0.9444444444444444
Cross Validation Score: 0.4444444444444445
```

In [13]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.8333333333333334
C = 1
Training error: 0.05555555555555558
Accuracy: 0.9444444444444444
Cross Validation Score: 0.7777777777777778
C = 10
Training error: 0.05555555555555558
Accuracy: 0.9444444444444444
Cross Validation Score: 0.6666666666666666
C = 100
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.4444444444444445
C = 1000
Training error: 0.11111111111111116
Accuracy: 0.8888888888888888
Cross Validation Score: 0.5
```

Raisins Dataset

In [14]:
```
new_df2.columns
```

Out[14]:
```
Index(['Area', 'MajorAxisLength', 'MinorAxisLength', 'Eccentricity',
       'ConvexArea', 'Extent', 'Perimeter', 'Class'],
      dtype='object')
```

In [15]:
```
X2 = np.asarray(new_df2[['Area', 'MajorAxisLength', 'MinorAxisLength', 'Eccentricity',
       'ConvexArea', 'Extent', 'Perimeter',]])

Y2 = np.asarray(new_df2[['Class']])

X_train, X_test, Y_train, Y_test = train_test_split(X2,Y2, test_size = 0.2, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.17500000000000004
Accuracy: 0.825
Cross Validation Score: 0.7347222222222222
C = 1
Training error: 0.2236111111111111
Accuracy: 0.7763888888888889
Cross Validation Score: 0.7458333333333335
C = 10
Training error: 0.14583333333333337
Accuracy: 0.8541666666666666
Cross Validation Score: 0.6847222222222222
C = 100
Training error: 0.22083333333333333
Accuracy: 0.7791666666666667
Cross Validation Score: 0.7263888888888889
C = 1000
Training error: 0.19305555555555554
Accuracy: 0.8069444444444445
Cross Validation Score: 0.7138888888888889
```

In [16]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
```

```
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.26388888888888884
Accuracy: 0.7361111111111112
Cross Validation Score: 0.75
C = 1
Training error: 0.14166666666666672
Accuracy: 0.8583333333333333
Cross Validation Score: 0.8263888888888888
C = 10
Training error: 0.18472222222222223
Accuracy: 0.8152777777777778
Cross Validation Score: 0.7166666666666667
C = 100
Training error: 0.31805555555555554
Accuracy: 0.6819444444444445
Cross Validation Score: 0.7694444444444445
C = 1000
Training error: 0.3277777777777777
Accuracy: 0.6722222222222223
Cross Validation Score: 0.8347222222222223
```

In [17]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.22638888888888886
Accuracy: 0.7736111111111111
Cross Validation Score: 0.7027777777777778
C = 1
Training error: 0.21944444444444444
Accuracy: 0.7805555555555556
Cross Validation Score: 0.8319444444444444
C = 10
Training error: 0.34305555555555556
Accuracy: 0.6569444444444444
Cross Validation Score: 0.6805555555555555
C = 100
Training error: 0.14166666666666672
Accuracy: 0.8583333333333333
Cross Validation Score: 0.7777777777777778
C = 1000
Training error: 0.17083333333333328
Accuracy: 0.8291666666666667
Cross Validation Score: 0.7250000000000001
```

50/50 Train/Test Split

In [18]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X2,Y2, test_size = 0.5, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.32666666666666666
Accuracy: 0.6733333333333333
Cross Validation Score: 0.7044444444444444
C = 1
Training error: 0.2733333333333333
Accuracy: 0.7266666666666667
Cross Validation Score: 0.7022222222222222
C = 10
Training error: 0.30666666666666664
Accuracy: 0.6933333333333334
Cross Validation Score: 0.7844444444444445
C = 100
Training error: 0.1844444444444444
Accuracy: 0.8155555555555556
Cross Validation Score: 0.8044444444444444
C = 1000
Training error: 0.1466666666666666
Accuracy: 0.8533333333333334
Cross Validation Score: 0.6666666666666666
```

In [19]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.1466666666666666
Accuracy: 0.8533333333333334
Cross Validation Score: 0.6466666666666666
C = 1
Training error: 0.35777777777777775
Accuracy: 0.6422222222222222
Cross Validation Score: 0.6044444444444445
C = 10
Training error: 0.3911111111111111
Accuracy: 0.6088888888888889
Cross Validation Score: 0.7799999999999999
C = 100
Training error: 0.22444444444444445
Accuracy: 0.7755555555555556
Cross Validation Score: 0.7444444444444445
C = 1000
Training error: 0.4066666666666666
Accuracy: 0.5933333333333334
Cross Validation Score: 0.7355555555555556
```

In [20]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.1466666666666666
Accuracy: 0.8533333333333334
Cross Validation Score: 0.8155555555555556
C = 1
Training error: 0.42666666666666664
Accuracy: 0.5733333333333334
Cross Validation Score: 0.7533333333333334
C = 10
Training error: 0.18888888888888888
Accuracy: 0.8111111111111111
Cross Validation Score: 0.6755555555555556
C = 100
Training error: 0.25111111111111106
Accuracy: 0.7488888888888889
Cross Validation Score: 0.6555555555555556
C = 1000
Training error: 0.3422222222222222
Accuracy: 0.6577777777777778
Cross Validation Score: 0.6822222222222223
```

20/80 Train/Test Split

In [21]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X2,Y2, test_size = 0.8, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.3555555555555555
Accuracy: 0.6444444444444445
Cross Validation Score: 0.7444444444444445
C = 1
Training error: 0.15555555555555556
Accuracy: 0.8444444444444444
Cross Validation Score: 0.6833333333333332
C = 10
Training error: 0.13888888888888884
Accuracy: 0.8611111111111112
Cross Validation Score: 0.6833333333333332
C = 100
Training error: 0.22777777777777775
Accuracy: 0.7722222222222223
Cross Validation Score: 0.8055555555555555
C = 1000
Training error: 0.21666666666666667
Accuracy: 0.7833333333333333
Cross Validation Score: 0.7722222222222223
```

In [22]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.1611111111111111
Accuracy: 0.8388888888888889
Cross Validation Score: 0.7944444444444444
C = 1
Training error: 0.15555555555555556
Accuracy: 0.8444444444444444
Cross Validation Score: 0.7222222222222223
C = 10
Training error: 0.19444444444444442
Accuracy: 0.8055555555555556
Cross Validation Score: 0.7277777777777777
C = 100
Training error: 0.1777777777777778
Accuracy: 0.8222222222222222
Cross Validation Score: 0.8444444444444444
C = 1000
Training error: 0.37222222222222223
Accuracy: 0.6277777777777778
Cross Validation Score: 0.5777777777777778
```

In [23]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.18888888888888888
Accuracy: 0.8111111111111111
Cross Validation Score: 0.6611111111111111
C = 1
Training error: 0.18888888888888888
Accuracy: 0.8111111111111111
Cross Validation Score: 0.6611111111111111
C = 10
Training error: 0.3388888888888889
Accuracy: 0.6611111111111111
Cross Validation Score: 0.8055555555555557
C = 100
Training error: 0.18888888888888888
Accuracy: 0.8111111111111111
Cross Validation Score: 0.7333333333333334
C = 1000
Training error: 0.25555555555555554
Accuracy: 0.7444444444444445
Cross Validation Score: 0.7111111111111111
```

Energy Efficiency Dataset

In [24]:
```
new_df3 = pd.read_excel('ENB2012_data.xlsx')
```

In [25]:
```
new_df3.columns = ['Compactness', 'Surface Area', 'Wall Area', 'Roof Area', 'Heigh', 'Orientation', 'Gla
new_df3 = new_df3.drop('Heating Load', axis = 1)
new_df3.head()
```

Out[25]:

	Compactness	Surface Area	Wall Area	Roof Area	Heigh	Orientation	Glazing Area	Glazing Area Distribution	Cooling Load
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	28.28

In [26]:
```
#new_df3['Cooling Load'] =np.where(new_df3['Cooling Load'] >= 22.08, 0,1)
new_df3['Cooling Load'] = np.where(new_df3['Cooling Load'] > 22.08,1,-1)
```

In [27]:
```
X3 = np.asarray(new_df3[['Compactness', 'Surface Area', 'Wall Area', 'Roof Area', 'Heigh', 'Orientation'

Y3 = np.asarray(new_df3[['Cooling Load']])

X_train, X_test, Y_train, Y_test = train_test_split(X3,Y3, test_size = 0.2, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397
C = 1
Training error: 0.13029315960912047
Accuracy: 0.8697068403908795
Cross Validation Score: 0.6807269249163079
C = 10
Training error: 0.23778501628664495
Accuracy: 0.762214983713355
Cross Validation Score: 0.7360194484297784
C = 100
Training error: 0.07003257328990231
Accuracy: 0.9299674267100977
Cross Validation Score: 0.5471465008767735
C = 1000
Training error: 0.18566775244299671
Accuracy: 0.8143322475570033
Cross Validation Score: 0.7441654710664753
```

In [28]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397
C = 1
Training error: 0.5
Accuracy: 0.5
Cross Validation Score: 0.5913279132791328
C = 10
Training error: 0.30781758957654726
Accuracy: 0.6921824104234527
Cross Validation Score: 0.7131994261119082
C = 100
Training error: 0.19218241042345274
Accuracy: 0.8078175895765473
Cross Validation Score: 0.6953052765821776
C = 1000
Training error: 0.1123778501628665
Accuracy: 0.8876221498371335
Cross Validation Score: 0.7181173282321058
```

In [29]:
```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397
C = 1
Training error: 0.5
Accuracy: 0.5
Cross Validation Score: 0.687079547266061
C = 10
Training error: 0.13355048859934848
Accuracy: 0.8664495114006515
Cross Validation Score: 0.6986051331101546
C = 100
Training error: 0.08631921824104238
Accuracy: 0.9136807817589576
Cross Validation Score: 0.7295233540570699
C = 1000
Training error: 0.10912052117263848
Accuracy: 0.8908794788273615
Cross Validation Score: 0.7278813964610235
```

50/50 Train/Test Split

In [30]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X3,Y3, test_size = 0.5, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
Cross Validation Score: 0.9739583333333334
C = 1
Training error: 0.03645833333333337
Accuracy: 0.9635416666666666
Cross Validation Score: 0.7760416666666666
C = 10
Training error: 0.08072916666666663
Accuracy: 0.9192708333333334
Cross Validation Score: 0.7864583333333334
C = 100
Training error: 0.10416666666666663
Accuracy: 0.8958333333333334
Cross Validation Score: 0.7526041666666666
C = 1000
Training error: 0.03645833333333337
Accuracy: 0.9635416666666666
Cross Validation Score: 0.6510416666666666
```

In [31]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
Cross Validation Score: 0.9739583333333334
C = 1
Training error: 0.0625
Accuracy: 0.9375
Cross Validation Score: 0.5286458333333334
C = 10
Training error: 0.08072916666666663
Accuracy: 0.9192708333333334
Cross Validation Score: 0.8203125
C = 100
Training error: 0.05989583333333337
Accuracy: 0.9401041666666666
Cross Validation Score: 0.7447916666666666
C = 1000
Training error: 0.1015625
Accuracy: 0.8984375
Cross Validation Score: 0.8125
```

In [32]:

```
for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
Cross Validation Score: 0.9739583333333334
C = 1
Training error: 0.5052083333333333
Accuracy: 0.4947916666666667
Cross Validation Score: 0.8125
C = 10
Training error: 0.1875
Accuracy: 0.8125
Cross Validation Score: 0.7578125
C = 100
Training error: 0.03125
Accuracy: 0.96875
Cross Validation Score: 0.7395833333333334
C = 1000
Training error: 0.06510416666666663
Accuracy: 0.9348958333333334
Cross Validation Score: 0.6223958333333334
```

20/80 Train/Test Split

In [33]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X3,Y3, test_size = 0.8, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
C = 1
Training error: 0.5359477124183006
Accuracy: 0.46405228758169936
Cross Validation Score: 0.5947712418300654
C = 10
Training error: 0.4509803921568627
Accuracy: 0.5490196078431373
Cross Validation Score: 0.5490196078431372
C = 100
Training error: 0.39869281045751637
Accuracy: 0.6013071895424836
Cross Validation Score: 0.5751633986928105
C = 1000
Training error: 0.4117647058823529
Accuracy: 0.5882352941176471
Cross Validation Score: 0.5882352941176471
```

In [34]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X3,Y3, test_size = 0.8, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
C = 1
Training error: 0.4183006535947712
Accuracy: 0.5816993464052288
Cross Validation Score: 0.5816993464052288
C = 10
Training error: 0.43137254901960786
Accuracy: 0.5686274509803921
Cross Validation Score: 0.6143790849673203
C = 100
Training error: 0.4117647058823529
Accuracy: 0.5882352941176471
Cross Validation Score: 0.6274509803921569
C = 1000
Training error: 0.47058823529411764
Accuracy: 0.5294117647058824
Cross Validation Score: 0.6274509803921569
```

In [35]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X3,Y3, test_size = 0.8, random_state = 5)

for C in Cs:
    classifier = svm.LinearSVC(penalty='l2',loss='hinge',C=C,max_iter=100000)
    classifier.fit(X_train, Y_train)
    e_training = calc_error(X_train, Y_train, classifier)
    print('C = {}'.format(C))
    print('Training error: {}'.format(e_training[0]))
    print('Accuracy: {}'.format(e_training[1]))
    print('Cross Validation Score: {}'.format(cross_val_score(classifier,X_train,Y_train, cv=3).mean()))
```

```
C = 0.1
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
C = 1
Training error: 0.4117647058823529
Accuracy: 0.5882352941176471
Cross Validation Score: 0.6013071895424836
C = 10
Training error: 0.39869281045751637
Accuracy: 0.6013071895424836
Cross Validation Score: 0.6013071895424836
C = 100
Training error: 0.47058823529411764
Accuracy: 0.5294117647058824
Cross Validation Score: 0.5751633986928104
C = 1000
Training error: 0.522875816993464
Accuracy: 0.477124183006536
Cross Validation Score: 0.5751633986928104
```