

# KNN

```
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
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
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]: def calc_error(X,Y, classifier):
Y_pred = classifier.predict(X)
accuracy = accuracy_score(Y, Y_pred)
error = 1 - accuracy

return error,accuracy
```

```
In [41]: 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 = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.8194444444444443
C = 1
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.8194444444444443
C = 10
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.8194444444444443
C = 100
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.8194444444444443
C = 1000
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
Cross Validation Score: 0.8194444444444443

```

```

In [5]: for C in Cs:
        classifier = RandomForestClassifier(max_depth=2, random_state=5)
        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: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 1
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 10
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 100
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 1000
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334

```

```

In [6]: for C in Cs:
        classifier = RandomForestClassifier(max_depth=2, random_state=5)
        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: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 1
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 10
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 100
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334
C = 1000
Training error: 0.16666666666666663
Accuracy: 0.8333333333333334

```

50/50

```
In [7]: X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.5, random_state = 5)

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

for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 1
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 10
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 100
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 1000
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
```

```
In [8]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 1
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 10
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 100
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 1000
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
```

```
In [9]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 1
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 10
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 100
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888
C = 1000
Training error: 0.1111111111111116
Accuracy: 0.8888888888888888

```

## 20/80

```

In [10]: X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.8, random_state = 5)

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

for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.0
Accuracy: 1.0
C = 1
Training error: 0.0
Accuracy: 1.0
C = 10
Training error: 0.0
Accuracy: 1.0
C = 100
Training error: 0.0
Accuracy: 1.0
C = 1000
Training error: 0.0
Accuracy: 1.0

```

```

In [11]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.0
Accuracy: 1.0
C = 1
Training error: 0.0
Accuracy: 1.0
C = 10
Training error: 0.0
Accuracy: 1.0
C = 100
Training error: 0.0
Accuracy: 1.0
C = 1000
Training error: 0.0
Accuracy: 1.0

```

```

In [12]: for C in Cs:
        classifier = RandomForestClassifier(max_depth=2, random_state=5)
        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: 0.0
Accuracy: 1.0
C = 1
Training error: 0.0
Accuracy: 1.0
C = 10
Training error: 0.0
Accuracy: 1.0
C = 100
Training error: 0.0
Accuracy: 1.0
C = 1000
Training error: 0.0
Accuracy: 1.0

```

## Raisins Dataset

```

In [13]: new_df2 = pd.read_excel('Raisin_Dataset.xlsx')

In [42]: X2 = np.asarray(new_df2[['Area', 'MajorAxisLength', 'MinorAxisLength', 'Eccentricity',
        'ConvexArea', 'Extent', 'Perimeter',]])

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

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

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 = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.13749999999999996
Accuracy: 0.8625
Cross Validation Score: 0.8513888888888889
C = 1
Training error: 0.13749999999999996
Accuracy: 0.8625
Cross Validation Score: 0.8513888888888889
C = 10
Training error: 0.13749999999999996
Accuracy: 0.8625
Cross Validation Score: 0.8513888888888889
C = 100
Training error: 0.13749999999999996
Accuracy: 0.8625
Cross Validation Score: 0.8513888888888889
C = 1000
Training error: 0.13749999999999996
Accuracy: 0.8625
Cross Validation Score: 0.8513888888888889

```

```

In [15]: for C in Cs:
          classifier = RandomForestClassifier(max_depth=2, random_state=5)
          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: 0.13749999999999996
Accuracy: 0.8625
C = 1
Training error: 0.13749999999999996
Accuracy: 0.8625
C = 10
Training error: 0.13749999999999996
Accuracy: 0.8625
C = 100
Training error: 0.13749999999999996
Accuracy: 0.8625
C = 1000
Training error: 0.13749999999999996
Accuracy: 0.8625

```

```

In [16]: for C in Cs:
          classifier = RandomForestClassifier(max_depth=2, random_state=5)
          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: 0.13749999999999996
Accuracy: 0.8625
C = 1
Training error: 0.13749999999999996
Accuracy: 0.8625
C = 10
Training error: 0.13749999999999996
Accuracy: 0.8625
C = 100
Training error: 0.13749999999999996
Accuracy: 0.8625
C = 1000
Training error: 0.13749999999999996
Accuracy: 0.8625

```

## 50/50

```
In [17]: 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 = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 10
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 100
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1000
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
```

```
In [18]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 10
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 100
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1000
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
```

```
In [19]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 10
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 100
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1000
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112

```

## 20/80

In [20]: `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 = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 10
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 100
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1000
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112

```

In [21]: `for C in Cs:`

```

    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 10
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 100
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1000
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112

```



```
In [22]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 10
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 100
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
C = 1000
Training error: 0.14888888888888885
Accuracy: 0.8511111111111112
```

## Energy Dataset

```
In [23]: new_df3 = pd.read_excel('ENB2012_data.xlsx')

In [24]: new_df3.columns = ['Compactness', 'Surface Area', 'Wall Area', 'Roof Area', 'Heigh', 'Orientation', 'Gla
new_df3 = new_df3.drop('Heating Load', axis = 1)

In [25]: new_df3['Cooling Load'] = np.where(new_df3['Cooling Load'] > 22.08,1,-1)
```

## 80/20

```
In [43]: 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 = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397
C = 10
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397
C = 100
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397
C = 1000
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
Cross Validation Score: 0.980439980870397

```

```

In [27]: for C in Cs:
        classifier = RandomForestClassifier(max_depth=2, random_state=5)
        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
C = 1
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
C = 10
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
C = 100
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
C = 1000
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319

```

```

In [28]: for C in Cs:
        classifier = RandomForestClassifier(max_depth=2, random_state=5)
        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
C = 1
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
C = 10
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
C = 100
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319
C = 1000
Training error: 0.019543973941368087
Accuracy: 0.9804560260586319

```

50/50

```
In [29]: 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 = RandomForestClassifier(max_depth=2, random_state=5)
    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
C = 1
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
C = 10
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
C = 100
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
C = 1000
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
```

```
In [40]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
C = 10
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
C = 100
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
C = 1000
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
Cross Validation Score: 0.9738562091503268
```

```
In [31]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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
C = 1
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
C = 10
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
C = 100
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334
C = 1000
Training error: 0.02604166666666663
Accuracy: 0.9739583333333334

```

## 20/80

In [32]: `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 = RandomForestClassifier(max_depth=2, random_state=5)
    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
C = 1
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
C = 10
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
C = 100
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
C = 1000
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268

```

In [33]: `for C in Cs:`

```

    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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
C = 1
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
C = 10
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
C = 100
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268
C = 1000
Training error: 0.02614379084967322
Accuracy: 0.9738562091503268

```

```
In [34]: for C in Cs:
    classifier = RandomForestClassifier(max_depth=2, random_state=5)
    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

C = 1

Training error: 0.02614379084967322

Accuracy: 0.9738562091503268

C = 10

Training error: 0.02614379084967322

Accuracy: 0.9738562091503268

C = 100

Training error: 0.02614379084967322

Accuracy: 0.9738562091503268

C = 1000

Training error: 0.02614379084967322

Accuracy: 0.9738562091503268