sklearn_DTClassifier

September 26, 2017

In [ ]:
```
%matplotlib inline
```

```
# Dependencies
import random
import pandas as pd


# Set the seed (reproducibility)
random.seed(0)


# Data import and cleaning
df = pd.read_csv("./speedbumps.csv")  # read data from the .csv file
df = df.loc[:, ('speedbump', 'Speed', 'X', 'Y', 'Z', 'z_jolt')]  # only select relevan
keywords = ['yes', 'no']
mapping = [1, 0]
df = df.replace(keywords, mapping)
print(df.head(10))
```

```
   speedbump  Speed         X         Y         Z    z_jolt
0          0   0.94  0.056671 -0.032822 -0.990891  0.000000
1          0   0.55  0.056671 -0.032822 -0.990891  0.000000
2          0   0.55  0.064835  0.007797 -1.030807 -0.039916
3          0   0.55  0.078796  0.028397 -1.008896  0.021911
4          0   0.55  0.058334 -0.015610 -0.990509  0.018387
5          0   0.55  0.075516  0.004745 -0.978210  0.012299
6          0   0.55  0.056717  0.022415 -1.002472 -0.024262
7          0   0.55  0.117401  0.025574 -1.017487 -0.015015
8          0   0.55  0.099884  0.018570 -0.995087  0.022400
9          0   0.55  0.118179  0.014740 -0.993744  0.001343
```

In [4]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
```

```
# Separate Y and X variables
df_label = df.loc[:, 'speedbump']
df_feature = df.loc[:, ('Speed', 'Z', 'z_jolt')]
Y = df_label.as_matrix()
X = df_feature.as_matrix()


# Prepare for cross-validation
clf = DecisionTreeClassifier()  # create a DecisionTreeClassifier
f1_sum = 0.00  # sum of F1 scores
cv = 10;  # number of cross-validations


# Start cross-validation
for i in range(0, cv, 1):

    # split to train and test sets
    train_X, test_X, train_Y, test_Y = train_test_split(X, Y, test_size=0.2, shuffle=T

    # start training
    clf = clf.fit(train_X, train_Y)  # fit the training data

    # start testing
    predicted_Y = clf.predict(test_X)  # predict on the testing data

    # calculate the F1 score
    f1 = metrics.f1_score(test_Y, predicted_Y, average='binary')  # calculate the F1 s
    f1_sum += f1

    # calculate the confusion matrix
    matrix = metrics.confusion_matrix(test_Y, predicted_Y)

    # print iterative result
    print('\n-----------------------------------')
    print('Iteration ', i)
    print('Features: speed, Z-accel, Z-jolt')
    print('Labels: speedbump (1 = yes, 0 = no)')
    print('F1 score:', f1)
    print(matrix)


# Calculate cross-validation average
f1_average = f1_sum / cv
print('\n-----------------------------------')
print('sklearn Decision Tree Model')
print('\tFeatures: speed, Z-accel, Z-jolt')
print('\tLabels: speedbump (1 = yes, 0 = no)')
print('\tAverage F1 score:', f1_average)
```

```
----------------------------------
Iteration  0
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[422   0]
 [  0   2]]

----------------------------------
Iteration  1
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[422   0]
 [  0   2]]

----------------------------------
Iteration  2
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 0.666666666667
[[420   0]
 [  2   2]]

----------------------------------
Iteration  3
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[419   0]
 [  0   5]]

----------------------------------
Iteration  4
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[421   0]
 [  0   3]]

----------------------------------
Iteration  5
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[420   0]
 [  0   4]]
```

```
----------------------------------
Iteration  6
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[421   0]
 [  0   3]]

----------------------------------
Iteration  7
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[421   0]
 [  0   3]]

----------------------------------
Iteration  8
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[421   0]
 [  0   3]]

----------------------------------
Iteration  9
Features: speed, Z-accel, Z-jolt
Labels: speedbump (1 = yes, 0 = no)
F1 score: 1.0
[[421   0]
 [  0   3]]

----------------------------------
sklearn Decision Tree Model
        Features: speed, Z-accel, Z-jolt
        Labels: speedbump (1 = yes, 0 = no)
        Average F1 score: 0.966666666667
```