# Analysis

March 25, 2017

In [2]:
```
from pandas import DataFrame
from __future__ import division
import numpy as np
```

In [3]:
```
data = DataFrame.from_csv('data', sep=',', index_col=None)
```

In [4]:
```
data[:3]
```

Out[4]:

| | QA | PRESCREEN | PT5 | PT6 | PT7 | PT8 | PT9 | PT10 | NORM1 | NORM2 | NORM3 | NORM4 | NORM5 | NORM6 | NORM7 | NORM8 | EUCLID | DIAMETER | AMFM | CLASS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 22 | 22 | 22 | 19 | 18 | 14 | 49.895756 | 17.775994 | 5.270920 | 0.771761 | 0.018632 | 0.006864 | 0.003923 | 0.003923 | 0.486903 | 0.100025 | 1 | 0 |
| 1 | 1 | 1 | 24 | 24 | 22 | 18 | 16 | 13 | 57.709936 | 23.799994 | 3.325423 | 0.234185 | 0.003903 | 0.003903 | 0.003903 | 0.003903 | 0.520908 | 0.144414 | 0 | 0 |
| 2 | 1 | 1 | 62 | 60 | 59 | 54 | 47 | 33 | 55.831441 | 27.993933 | 12.687485 | 4.852282 | 1.393889 | 0.373252 | 0.041817 | 0.007744 | 0.530904 | 0.128548 | 0 | 1 |

In [5]:
```
len(data)
```

Out[5]:
```
1151
```

In [6]:
```
temp = [
    [x for x in data['QA']],
    [x for x in data['PRESCREEN']],
    [x for x in data['PT5']],
    [x for x in data['PT8']],
    [x for x in data['PT10']],
    [x for x in data['DIAMETER']]
]

total = [x for x in zip(*temp)]
training = total[:1036]
testing = total[1036:]
```

```
In [7]: actual_data = {
            'data': np.array(training),
            'labels': [x for x in data['CLASS'][:1036]]
        }

        testing_data = {
            'data': np.array(testing),
            'labels': [x for x in data['CLASS'][1036:]]
        }

In [9]: from sklearn.tree import DecisionTreeClassifier

In [10]: clf = DecisionTreeClassifier()

In [11]: clf.fit(actual_data['data'], actual_data['labels'])

Out[11]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None
                     max_features=None, max_leaf_nodes=None, min_samples_leaf=1,
                     min_samples_split=2, min_weight_fraction_leaf=0.0,
                     presort=False, random_state=None, splitter='best')

In [12]: testing[:10], clf.predict(testing)

Out[12]: ([(1, 1, 66, 61, 48, 0.10626099999999999),
           (1, 1, 13, 9, 4, 0.087937000000000001),
           (1, 1, 64, 58, 37, 0.11650899999999999),
           (1, 1, 15, 14, 5, 0.092896000000000006),
           (1, 1, 33, 29, 21, 0.11494600000000001),
           (1, 1, 59, 51, 30, 0.148864),
           (1, 1, 49, 44, 27, 0.099251000000000006),
           (1, 1, 29, 27, 22, 0.103504),
           (1, 1, 18, 17, 10, 0.090307999999999999),
           (1, 1, 24, 20, 10, 0.101503)],
          array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0,
                 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1,
                 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1,
                 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0,
                 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1,

In [13]: clf.predict([(1,1,3,3,3,0.09), (1,1,33,63,93,0.10), (1,1,10,20,30,0.10)])

Out[13]: array([0, 0, 0])

In [14]: clf.score(testing_data['data'], testing_data['labels'])

Out[14]: 0.68695652173913047

In [ ]:
```