In [362]:
```
import numpy as np
def split(ratio,data):
#     np.random.shuffle(data)
    index = int(ratio * len(data))
    return data[:index,:],data[index:,:]
```

In [369]:
```
data = np.genfromtxt('tennis.csv',delimiter=',',skip_header=1,dtype='str')
train,test = split(0.85,data)
print("Train set: \n{0}\n".format(train))
print("Test set: \n{0}\n".format(test))

#classify
n = train.shape[0]
target,idx= np.unique(train[:,-1],return_inverse=True)
target_count = np.bincount(idx)
target_prob = target_count/float(n)
target_dict = {k:v for k, v in zip(target, target_count)}

print("target\tcount\tprobablity\n")
for i,j,k in zip(target,target_count,target_prob):
    print("{0}\t{1}\t{2}".format(i,j,k))

values,idx = np.unique(train[:,:-1],return_inverse=True)
counts = np.bincount(idx)
#feature probability
d = {k: {'total': float(v),'yes':0.0,'no':0.0} for k, v in zip(values, counts)}
print("\nDict for each feature:\n")
for row in train:
    for attrib in row[:-1]:
        if(row[-1]=='yes'):
            d[attrib]['yes']+=1
        else:
            d[attrib]['no']+=1
print(d)
```

```
Train set:
[['sunny' 'hot' 'high' 'Weak' 'no']
 ['sunny' 'hot' 'high' 'Strong' 'no']
 ['overcast' 'hot' 'high' 'Weak' 'yes']
 ['rainy' 'mild' 'high' 'Weak' 'yes']
 ['rainy' 'cool' 'normal' 'Weak' 'yes']
 ['rainy' 'cool' 'normal' 'Strong' 'no']
 ['overcast' 'cool' 'normal' 'Strong' 'yes']
 ['sunny' 'mild' 'high' 'Weak' 'no']
 ['sunny' 'cool' 'normal' 'Weak' 'yes']
 ['rainy' 'mild' 'normal' 'Weak' 'yes']
 ['sunny' 'mild' 'normal' 'Strong' 'yes']]

Test set:
[['overcast' 'mild' 'high' 'Strong' 'yes']
 ['overcast' 'hot' 'normal' 'Weak' 'yes']
 ['rainy' 'mild' 'high' 'Strong' 'no']]

target  count   probablity

no      4       0.363636363636
yes     7       0.636363636364

Dict for each feature:

{'rainy': {'yes': 3.0, 'total': 4.0, 'no': 1.0}, 'normal': {'yes': 5.0, 'total': 6.0, 'no': 1.0}, 'sun
ny': {'yes': 2.0, 'total': 5.0, 'no': 3.0}, 'overcast': {'yes': 2.0, 'total': 2.0, 'no': 0.0}, 'Weak':
{'yes': 5.0, 'total': 7.0, 'no': 2.0}, 'mild': {'yes': 3.0, 'total': 4.0, 'no': 1.0}, 'high': {'yes':
2.0, 'total': 5.0, 'no': 3.0}, 'hot': {'yes': 1.0, 'total': 3.0, 'no': 2.0}, 'Strong': {'yes': 2.0, 't
otal': 4.0, 'no': 2.0}, 'cool': {'yes': 3.0, 'total': 4.0, 'no': 1.0}}
```

In [370]:
```
print("\nProbabity Dict for each feature:\n")
for i in d:
    d[i]['yes']/=target_dict['yes']
    d[i]['no']/=target_dict['no']
    d[i]['total']/=n
print(d)
```

```
Probabity Dict for each feature:

{'rainy': {'yes': 0.42857142857142855, 'total': 0.36363636363636365, 'no': 0.25}, 'normal': {'yes': 0.
7142857142857143, 'total': 0.5454545454545454, 'no': 0.25}, 'sunny': {'yes': 0.2857142857142857, 'tota
l': 0.45454545454545453, 'no': 0.75}, 'overcast': {'yes': 0.2857142857142857, 'total': 0.1818181818181
8182, 'no': 0.0}, 'Weak': {'yes': 0.7142857142857143, 'total': 0.6363636363636364, 'no': 0.5}, 'mild':
{'yes': 0.42857142857142855, 'total': 0.36363636363636365, 'no': 0.25}, 'high': {'yes': 0.285714285714
2857, 'total': 0.45454545454545453, 'no': 0.75}, 'hot': {'yes': 0.14285714285714285, 'total': 0.272727
2727272727, 'no': 0.5}, 'Strong': {'yes': 0.2857142857142857, 'total': 0.36363636363636365, 'no': 0.
5}, 'cool': {'yes': 0.42857142857142855, 'total': 0.36363636363636365, 'no': 0.25}}
```

In [371]:
```
def predict(t):
    t_yes =  np.array([d[i]['yes'] for i in t])
    t_no =  np.array([d[i]['no'] for i in t])
    t_x = np.array([d[i]['total'] for i in t])

    p_yes = np.prod(t_yes)*target_prob[1]
    p_no = np.prod(t_no)*target_prob[0]
    p_x = np.prod(t_x)
    p_yes /= p_x
    p_no /= p_x

    if p_yes > p_no:
        return 'yes'
    else:
        return 'no'
```

In [372]:
```
def get_accuracy(testdata):
    accuracy = 0.0
    size = testdata.shape[0]
    for i in testdata:
        p = predict(i[:-1])
        print("actual: {0}\nprediction:{1}\n".format(i,p))
        if p == i[-1]:
            accuracy += 1
    print("Accuracy: {0}%".format(accuracy/size *100))
```

In [373]:
```
get_accuracy(test)
```

```
actual: ['overcast' 'mild' 'high' 'Strong' 'yes']
prediction:yes

actual: ['overcast' 'hot' 'normal' 'Weak' 'yes']
prediction:yes

actual: ['rainy' 'mild' 'high' 'Strong' 'no']
prediction:yes

Accuracy: 66.6666666667%
```

In [374]:
```
t = np.array(['sunny','cool','high','Strong'])
predict(t)
```

Out[374]: 'no'

In [ ]:

In [ ]: