In [105]:
```
import numpy as np
import pandas as pd
```

In [106]:
```
train_df = pd.read_csv('nnTrain.csv')
train_df
```

Out[106]:

	Instance	var1	var2	var3	class
0	1	0.248979	0.872253	0.676742	yes
1	2	0.762864	0.368187	0.725981	yes
2	3	0.235919	0.872500	0.062327	yes
3	4	0.084664	0.162050	0.903461	yes
4	5	0.093110	0.722764	0.710013	yes
5	6	0.212510	0.722030	0.605719	yes
6	7	0.779621	0.631424	0.765310	yes
7	8	0.514526	0.649687	0.282341	no
8	9	0.548212	0.952251	0.361861	no
9	10	0.361917	0.621872	0.649182	no
10	11	0.924183	0.232324	0.412386	no
11	12	0.541250	0.479775	0.603006	no
12	13	0.701861	0.727638	0.338815	no
13	14	0.502315	0.605147	0.423545	no
14	15	0.285370	0.386854	0.287218	no

In [107]:
```
test_df = pd.read_csv('nnTest.csv')
test_df
```

Out[107]:

	Unnamed: 0	var1	var2	var3	class
0	instance1	0.530309	0.510636	0.808336	NaN
1	instance2	0.674608	0.326183	0.200622	NaN
2	instance3	0.836194	0.454514	0.729288	NaN
3	instance4	0.530309	0.510636	0.808336	NaN

In [108]:
```
n = train_df.shape[0]
m = test_df.shape[0]
print(n, " ", m)
dist = pd.DataFrame(np.zeros((n, m)))
for i in range(n):
    dist[0][i] = (train_df['var1'][i] - test_df['var1'][0]) ** 2 + \
                 (train_df['var2'][i] - test_df['var2'][0]) ** 2 + \
                 (train_df['var3'][i] - test_df['var3'][0]) ** 2

dist[0]
```

```
15   4
```

Out[108]:
```
0     0.227230
1     0.081156
2     0.774140
3     0.329160
4     0.245809
5     0.186737
6     0.078597
7     0.296255
8     0.394684
9     0.066059
10    0.389371
11    0.043233
12    0.296970
13    0.157780
14    0.346881
Name: 0, dtype: float64
```

In [118]:
```
n = train_df.shape[0]
m = test_df.shape[0]
print(n, " ", m)
dist = pd.DataFrame(np.zeros((n, m)))
for j in range(m):
    for i in range(n):
        dist[j][i] = (train_df['var1'][i] - test_df['var1'][j]) ** 2 + \
               (train_df['var2'][i] - test_df['var2'][j]) ** 2 + \
               (train_df['var3'][i] - test_df['var3'][j]) ** 2
dist.columns = ['test0', 'test1', 'test2', 'test3']
dist
```

```
15   4
```

Out[118]:

	test0	test1	test2	test3
0	0.227230	0.706043	0.522088	0.227230
1	0.081156	0.285556	0.012841	0.081156
2	0.774140	0.510036	0.979879	0.774140
3	0.329160	0.868956	0.680669	0.329160
4	0.245809	0.754896	0.624503	0.245809
5	0.186737	0.534333	0.475816	0.186737
6	0.078597	0.423072	0.035795	0.078597
7	0.296255	0.136959	0.341324	0.296255
8	0.394684	0.433935	0.465678	0.394684
9	0.066059	0.386414	0.259364	0.066059
10	0.389371	0.115941	0.157537	0.389371
11	0.043233	0.203288	0.103577	0.043233
12	0.296970	0.181006	0.245111	0.296970
13	0.157780	0.157200	0.227644	0.157780
14	0.346881	0.162686	0.503411	0.346881

In [127]:
```
freq = train_df.iloc[dist.nsmallest(4, 'test3')['test3'].index]['class'].value_co
freq
```

Out[127]:
```
yes    2
no     2
Name: class, dtype: int64
```

In [130]:
```
#k=3
k=4
m = test_df.shape[0]
decision = pd.Series([])
for j in range(m):
    test_inst = 'test'+str(j)
    freq = train_df.iloc[dist.nsmallest(k, test_inst)[test_inst].index]['class'].
freq = pd.DataFrame(freq)
decision[j] = freq[freq['class']==freq['class'].max()].reset_index()
['index']
print('test_inst: ', decision)
```

```
test_inst:  3      index  class
0   yes      2
1    no      2
dtype: object

<ipython-input-130-abedd1fe5782>:4: DeprecationWarning: The default dtype for e
mpty Series will be 'object' instead of 'float64' in a future version. Specify
a dtype explicitly to silence this warning.
  decision = pd.Series([])
```

In [131]:
```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors = 3)
X = train_df.drop(columns = ['class'])
y = train_df['class']
neigh.fit(X, y)
X_test = test_df.drop(columns = ['class'])
y_test = test_df['class']
print(X)
neigh.predict(X)
```

```
    Instance      var1      var2      var3
0          1  0.248979  0.872253  0.676742
1          2  0.762864  0.368187  0.725981
2          3  0.235919  0.872500  0.062327
3          4  0.084664  0.162050  0.903461
4          5  0.093110  0.722764  0.710013
5          6  0.212510  0.722030  0.605719
6          7  0.779621  0.631424  0.765310
7          8  0.514526  0.649687  0.282341
8          9  0.548212  0.952251  0.361861
9         10  0.361917  0.621872  0.649182
10        11  0.924183  0.232324  0.412386
11        12  0.541250  0.479775  0.603006
12        13  0.701861  0.727638  0.338815
13        14  0.502315  0.605147  0.423545
14        15  0.285370  0.386854  0.287218
```

Out[131]:
```
array(['yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'no', 'no', 'no',
       'no', 'no', 'no', 'no', 'no'], dtype=object)
```