In [8]:
```
from sklearn.neural_network import MLPClassifier
import numpy as np
import pandas as pd

def splitInputOutput(data):
    x, y = np.array(data[:,0:-1], dtype=float), np.array(data[:,-1],dtype=int)
    y = np.squeeze(np.asarray(y)) #y.reshape(1,len(y)).T
    return x,y

df = pd.read_csv('data/prepared/dataWithTempRandomized.train.csv',sep=',',names=["Temp", "Nausea", "Lumbar", "Pushing","Micturi
df["Temp"] = df.transform(lambda x: x - 37)

X,y = splitInputOutput(df.as_matrix())

clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                    hidden_layer_sizes=(X.shape[1], X.shape[0]), random_state=1)

clf.fit(X, y)

df = pd.read_csv('data/prepared/dataWithTempRandomized.test.csv',sep=',',names=["Temp", "Nausea", "Lumbar", "Pushing","Micturit
df["Temp"] = df.transform(lambda x: x - 37)

X,y = splitInputOutput(df.as_matrix())

result = clf.predict(X)
mse = 0.5*np.sum((y - result)**2)
result = np.double(result > 0.5)
print()
print("MSE against Test data: ",mse)
print("Accuracy: ",1-np.sum(y-result)/y.shape[0])
```

```
MSE against Test data:  0.0
Accuracy:  1.0
```