In [1]:

```
import numpy as np
import mltools as ml
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import StandardScaler

X = np.genfromtxt('data/X_train.txt', delimiter = None) #100,000 data sets, 14
features
Y = np.genfromtxt('data/Y_train.txt', delimiter = None)
Xf = np.genfromtxt('data/X_test.txt', delimiter = None)
X,Y = ml.shuffleData(X,Y)

scaler = StandardScaler()
scaler.fit(X)

X = scaler.transform(X)
Xf = scaler.transform(Xf)

Xtr = X[:80000,]
Ytr = Y[:80000,]

Xva = X[80001: 100000,]
Yva = Y[80001: 100000,]


nnTest = MLPClassifier(hidden_layer_sizes = (1000, 4), activation = 'logistic'
, solver = 'adam',
                       learning_rate_init = 0.001)


nnTest.fit(X,Y)
```

Out[1]:

```
MLPClassifier(activation='logistic', alpha=0.0001, batch_size='auto',
       beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
       hidden_layer_sizes=(1000, 4), learning_rate='constant',
       learning_rate_init=0.001, max_iter=200, momentum=0.9,
       nesterovs_momentum=True, power_t=0.5, random_state=None,
       shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1,
       verbose=False, warm_start=False)
```

In [2]:

```
print("{0:>15}: {1:.4f}".format('Train AUC 2',nnTest.score(Xtr, Ytr)))
print("{0:>15}: {1:.4f}".format('Valid AUC 2',nnTest.score(Xva, Yva)))

kagle = nnTest.predict_proba(Xf)
Vali = nnTest.predict_proba(X)

Y_sub = np.vstack([np.arange(X.shape[0]), kagle[:, 1]]).T
Y_vali = np.vstack([np.arange(X.shape[0]), Vali[:, 1]]).T

np.savetxt('data/Y_sub.txt', Y_sub, '%d, %.5f', header='ID,Prob1', comments=''
, delimiter=',')
np.savetxt('data/Y_Vali.txt', Y_vali, '%d, %.5f', header='ID,Prob1', comments=
'', delimiter=',')
```

```
    Train AUC 2: 0.7035
    Valid AUC 2: 0.7086
```

In [ ]:

In [ ]: