0N	1N	2N	3N	4N	0c	1c	2c	3c	4c
4.54247067	0.38345031	0.175563	0.14178365	0.16083836	4.54311903	0.38461353	0.17781528	0.14444051	0.16083836
4.54247067	0.64109318	0.71562849	0.90839291	1.11565786	4.54953899	0.64864211	0.75062113	0.94197282	1.11565786
4.54247067	1.29037245	1.96724039	2.65084114	3.65327973	4.55747296	1.32346215	2.11974805	3.02737992	3.65327973
4.54247067	0.79994274	0.82808255	0.98494977	1.194	4.56619867	0.84061416	1.2070898	1.27119197	1.194
4.54247067	1.91776775	3.3317221	4.5482572	5.13926667	4.919928	2.83567943	4.6514345	4.97124727	5.13926667

```
def generate_matrix(name):
    result = []
    with open(name, 'r') as csvfile:
        data = csv.reader(csvfile, delimiter=',')
        count = 0
        for line in data:
            if count == 0:
                count += 1
                continue
            list = []
            for integer in line:
                list.append(float(integer))
            result.append(list)
    return result

def dim_reduction(original, iris, dim, is_iris):
    X = []
    mean = 0
    if is_iris == True:
        X = np.cov(iris.T)
        mean = iris.mean(axis=0)
    else:
        X = np.cov(original.T)
        mean = original.mean(axis=0)
    eigenvalue, eigenvector = la.eig(X)
    idx = eigenvalue.argsort()[::-1]
    eigenvalue = eigenvalue[idx]
    eigenvector = eigenvector[:,idx]
    mse = 0
    new_x = []
    for i in range(len(original)):
        x = np.zeros(4)
        for j in range(dim):
            x = x + np.dot(eigenvector[:, j].T, original[i] - mean) * eigenvector[:, j]
        x = x + mean
        new_x.append(x)
        mse += la.norm(iris[i] - x)**2
    mse = mse / len(original)
```