```python
#!/usr/bin/python3

import matplotlib.pyplot as plot
import matplotlib.lines as mlines

from perceptron import Perceptron

def graph_line(p1, p2):
    ax = plot.gca()
    xmin, xmax = ax.get_xbound()

    if(p2[0] == p1[0]):
        xmin = xmax = p1[0]
        ymin, ymax = ax.get_ybound()
    else:
        ymax = p1[1]+(p2[1]-p1[1])/(p2[0]-p1[0])*(xmax-p1[0])
        ymin = p1[1]+(p2[1]-p1[1])/(p2[0]-p1[0])*(xmin-p1[0])

    l = mlines.Line2D([xmin,xmax], [ymin,ymax], color='red')
    ax.add_line(l)
    return l

def get_point(per, x):
    return (x, per.getY(x))

def each(data):
    return range(len(data))

def getData(i):
    parse = lambda s, i: float(s.split(' ')[i])
    with open('messedupdat{}'.format(i)) as inf:
        data = [[parse(l, 0), parse(l, 1)] for l in inf.readlines()]
    outputs = [i for l in each(data)]
    return data, outputs

data1, ans1 = getData(1)
data2, ans2= getData(2)

print(data1)

plot.scatter([e[0] for e in data1], [e[1] for e in data1], marker='o')
plot.scatter([e[0] for e in data2], [e[1] for e in data2], marker='^')


xarray = data1+data2
outputs = ans1+ans2
perceptron = Perceptron(xarray, outputs, False)
for i in range(30):
    perceptron.train()

#for o,xar in zip(outputs,xarray):
#    print(o,perceptron.getValue(xar),"Success: "+str(o == perceptron.getValue(xar)))

p1 = get_point(perceptron, 0)
p2 = get_point(perceptron, 1)
print(p1)
print(p2)
graph_line(p1, p2)

plot.show()
```