

```

196 RecogLine = namedtuple('RecogLine', 'polyExpr vectorExpr')
197
198 def recogLineOfEu(class1, class2):
199     vector = nu.flipud(class1.mean - class2.mean)
200     vector[0,0] = -vector[0,0]
201
202     x, y = sy.symbols('x,y')
203     middlePoint = (class1.mean + class2.mean) / 2.0
204     poly = y - middlePoint[1,0] - vector[1,0]/vector[0,0] * (x - middlePoint[0,0])
205     return RecogLine(poly, vector)
206
207
208 def recogLineOfWe(class1, class2, weight):
209     x,y = sy.symbols('x,y')
210     poly = 0
211     for var, m1, m2, w in zip([x,y], class1.mean[:,0], class2.mean[:,0], weight[:,0]):
212         poly += (w*(var - m1))**2 - (w*(var - m2))**2
213     return RecogLine(poly, 'NONE')
214
215
216 def recogLineOfSi(class1, class2):
217     vector = class1.mean + class2.mean
218     vector /= nu.linalg.norm(vector)
219
220     x,y = sy.symbols('x,y')
221     poly = y - vector[1,0]/vector[0,0]*x
222     return RecogLine(poly, vector)
223
224
225 def recogLineOfNe(class1, class2):
226     global recogLineFile
227     if os.path.exists(recogLineFile):
228         points = readFileOfPoints(recogLineFile)
229         return points
230     else:
231         print('processing expensive tasks...')
232         process1 = mp.Process(target=subprocessForSearchingrecogLineOfSi, args=(class1
233         process1.start()
234         process1.join(timeout=300)
235         process1.terminate()
236         if os.path.exists(recogLineFile) == False:
237             print('timeout, work not finished.')
238             print(error)
239
240         print('second process...\n')

```