**1. python strip() and split()**

e.g.; strr=’1;2;3;4;\n’

\n 换行, \r 回车

strip()删除开头和结尾处空白符, 包括’\n’,’\r’,’\t’,’’

>>> ('1;2;3;4;\n').strip()

'1;2;3;4;'

>>> ('1;2;3;4;\n').strip().split(';')

['1', '2', '3', '4', '']

>>> ('1;2;3;4;\n').split(';')

['1', '2', '3', '4', '\n']

>>> ('1;2;3;4;\n').strip(';\n')

'1;2;3;4'

>>> ('1;2;3;4;\n').strip(';\n').split(';')

'1;2;3;4'

**2. numpy mat(), asmatrix() / transpose() 矩阵转置/ shape() (行, 列)形式的元组([1, 1, 1], [1, 1, 1]) 2行3列**

import numpy as np

np.asmatrix(x)

from numpy import \*

简介：matrix和array 区别

matrix 必须是二维, array 可以是多维

在numpy中matrix的主要优势是：相对简单的乘法运算符号，比如a,b两个matrices, a\*b, 就是矩阵积.

a=np.mat(‘4 3; 2 1’) b=np.mat(‘1 2; 3 4’) print (a) : [[4 3] [2 1]] print(b): [[1 2] [3 4]]

print (a\*b) : [[13 20] [5 8]]

x=array ([[1, 2], [3, 4]]) y=array ([3, 4], [2, 1]) x\*y 相当于点乘 x.\*y

对于matrix

m=np.mat([1, 2], [2, 3]) matrix([1, 2], [2, 3]) mm=m.mean(1)

mm=matrix([1.5], [2.5]) mm.shape=(2, 1) m-mm matrix([[-0.5, 0.5], [-0.5, 0.5]])

对于array

a=np.array([1, 2], [2, 3]) array([1, 2], [2, 3]) am=a.mean(1)

am=array([1.5, 2.5]) am.shape (2, ) a-am = matrix([[-0.5, 0.5], [-0.5, 0.5]]) a-am[ : , np.newaxis] array ([-0.5, 0.5], [-0.5, 0.5])

x=np.array([[1, 2], [3, 4]]) m=np.asmatrix(x) x[0, 0]=5 m matrix([[5, 2], [3, 4]])

x=linspace(0, 4, 5) 0 到4 按照5个数等分 print x [0. 1. 2. 3. 4.]

x.shape (5, ) 说明一行 5列 想把x从1行变成1列 y=transpose(x) 会失败，

正确的做法是：x.shape(5, 1) y=transpose(x)

另外对于一位矩阵转置尽量使用 matrix, mat, asmatrix 转化为矩阵形式,如

m=np.asmatrix([2, 3])/np.matrix([2, 3])/np.mat([2, 3]) 变成 [[2 3]], 然后

n=m.transpose() 变成 [[2] [3]]

**3 zeros((m, n)) m行n列 0矩阵 数组创建和数据类型**

import numpy as np

arr = np.arange(10) #创建拥有10个元素的数组 [0,1,2…9] print type(arr), np.shape(arr)

type(arr) is <type 'numpy.ndarray'> N-dimensional array np.shape(arr) 1维 (10, )

larr = arr.tolist() #转换list

arr = np.zeros((2,3,3)) #创建n维数组

[[[ 0. 0. 0.]

[ 0. 0. 0.]

[ 0. 0. 0.]]

[[ 0. 0. 0.]

[ 0. 0. 0.]

[ 0. 0. 0.]]]

mat = np.matrix(arr) #将数组转换为矩阵.

alist = [1, 2, 3]

np.array(alist) #使用List创建数组

#创建

arr = np.arange(100)

arr = np.arange(10,100)#创建包含10~99数组

arr = np.linspace(1, 2, 100) #创建包含100个取值范围在1~2之间的数组

arr = np.logspace(0, 1, 100, base=10) #返回包含100个取值范围在10+[0~1]之间的数组

cube = np.zeros((5,5,5)).**astype(int) + 1** #使用astype设置数据类型 从0变成1

cube = np.ones((5, 5, 5)).astype(np.float32)#创建3维数组，元素为1

#通过指定数据类型创建n维数组数组

arr = np.zeros(2, dtype=int)

arr = np.zeros(2, dtype=np.float32)

arr1d = np.arange(1000)#一维数组

arr3d = arr1d.reshape((10,10,10))#转换为3维数组

arr3d = np.reshape(arr1d, (10, 10, 10))

arr4d = np.zeros((10, 10, 10, 10))

arr1d = arr4d.ravel()#将4维数组转换为1维数组

**4. 索引和切片**

arr[0,1]#访问单个元?

arr[: , 1]#访问第2列

arr[1, :]#访问第2行

max 和 min 函数

**mat.max(0)/min(0), #n维数组axis=0维度的最大，最小值**

**5. numpy. tile (A, reps)**

建立array by 重复A the number of times given by reps.

a=np.array([0, 1, 2]) np.tile(a, 2) : array([0, 1, 2, 0, 1, 2])

np.tile(a, (2, 2)) : array([0, 1, 2, 0, 1, 2], [0, 1, 2, 0, 1, 2])

**6 numpy.sum(axis=0)**

>>> np.sum([[0, 1], [0, 5]], axis=0) array([0, 6])

>>> np.sum([[0, 1], [0, 5]], axis=1) array([1, 5])

>>> x = np.matrix([[1, 2], [4, 3]])

>>> x.sum() 10

>>> x.sum(axis=1)

matrix([[3], [7]])

>>> x.sum(axis=1, dtype='float')

matrix([[ 3.], [ 7.]])

>>> out = np.zeros((1, 2), dtype='float')

axis=0, axis=1

>>> b = arange(12).reshape(3,4)

>>> b

array([[ 0, 1, 2, 3],

[ 4, 5, 6, 7],

[ 8, 9, 10, 11]])

>>>

>>> b.sum(axis=0) # sum of each column

array([12, 15, 18, 21]) 同列相加

>>>

>>> b.min(axis=1) # min of each row

array([0, 4, 8]) 行 最小 同行

>>>

>>> b.cumsum(axis=1) # cumulative sum along each row

array([[ 0, 1, 3, 6],

[ 4, 9, 15, 22],

[ 8, 17, 27, 38]])

**7. [example[i] for example in dataSet]**

dataset: [[1, 1, 'yes'], [1, 0, 'yes'], [0, 1, 'no'], [0, 0, 'no']]

dataset依次赋值给example, 然后list 保留并输出example[i] i为0,1…

[example[-1] for example in dataSet] dataset 中最后一位输出为list

**8 Create empty set**

vocabSet = set([]) , set no duplicate entries.

set(“my name is Eric and Eric is my name”.split())

set([‘and’, ‘is’, ‘my’, ‘name’, ‘Eric’])

a=set([‘John’, ‘Eric’])

b=set([‘John’, ‘Jill’])

a.intersection(b) = set ([‘John’])

a.difference(b) a.union(b)

dict 包含keys/values. 有时我们只想要dict的key,不关心key对应的value, 目的就是保证集合元素不重复.

set 可以遍历

s = set([('Adam', 95), ('Lisa', 85), ('Bart', 59)])

for x in s: #set的元素是tuple，因此 for 循环的变量被依次赋值为tuple。

print x[0] + ':', x[1]

vocabSet = set([])

for document in dataset:

vocabSet = vocabSet |set (document) #合并两个set, 同时去掉冗余

或者 vocabSet.union(set(document)) #注意合并后已经乱序

return list(vocabSet) #转化并返回list的列表

**9 Others**

[0]\*len(vocabList) 定义一个list,全部为0 [0], 长度 len(vocabList)