# 基础

### 库说明

pandas：基于[NumPy](https://baike.so.com/doc/7632465-7906560.html) 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。==使用时前import pandas as pd==

numpy : 是Python的一种开源的数值计算扩展。这种工具可用来存储和处理大型矩阵，比Python自身的嵌套列表(nested list structure)结构要高效的多(该结构也可以用来表示矩阵(matrix))。==使用前要import numpy as np==

Matplotlib: 是一个 Python 的 2D绘图库，它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形 ==使用前要import Matplotlib as plt==

附pandas常用功能函数网址：https://blog.csdn.net/weixin\_41770169/article/details/79539232

seaborn:比Matplotlib方便的绘图工具 ==使用前要import seaborn as sns==

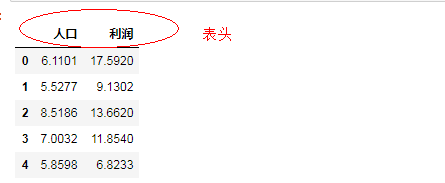
pd.read\_csv( )

作用是将csv文件读入并转化为数据框形式，有非常多的参数，用到时可查阅文档。

pd.read\_csv('数据集文件名称',names=['表头1','表头2']) ==#[,]是列表类型==

pd.read\_csv('数据集文件名称',head=None) ==#head=None表示没有表头==

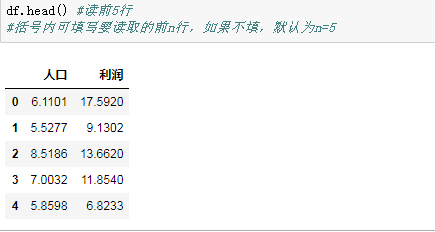
==需要注意的是，Jupyter notebook只能打开当前目录下的数据集，如csv，所以需要使用upload把数据集倒导入到当前目录下。==



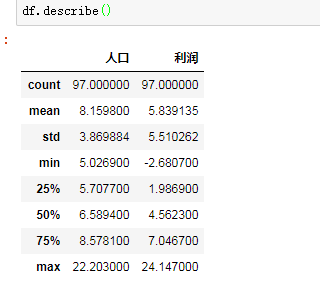
head=None的表头



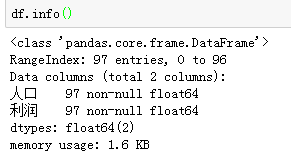
df.head(n):查看DataFrame对象的前n行,如果不填，默认为n=5



df.describe()：查看数据值列的汇总统计. ==#通过使用这个函数，我们可以知道数据集中数据的数量，平均值，最值等==



df.info()：查看索引、数据类型和内存信息



# seaborn：

## 初始化

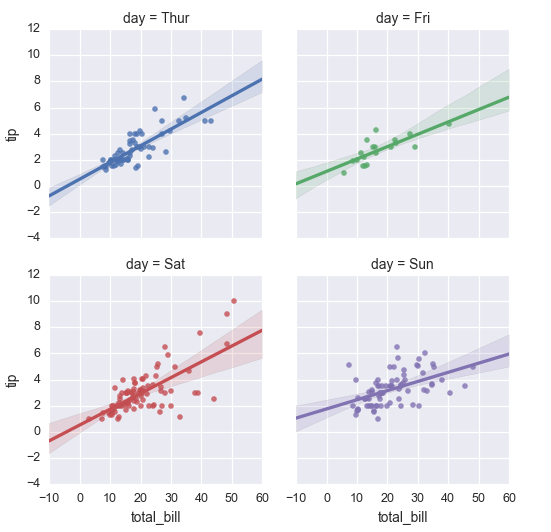
import seaborn as sns # 导入 seaborn 模块  
sns.set() # 使用 set() 方法 1 步设置默认样式  
  
#以下是sns.set()的默认参数  
sns.set(context='notebook', style='darkgrid', palette='deep', font='sans-serif', font\_scale=1, color\_codes=False, rc=None)  
  
其中：  
context='' 参数控制着默认的画幅大小，分别有 {paper, notebook, talk, poster} 四个值。其中，poster > talk > notebook > paper。  
style='' 参数控制默认样式，分别有 {darkgrid, whitegrid, dark, white, ticks}，你可以自行更改查看它们之间的不同。  
palette='' 参数为预设的调色板。分别有 {deep, muted, bright, pastel, dark, colorblind} 等，你可以自行更改查看它们之间的不同。  
font='' 用于设置字体  
font\_scale= 设置字体大小  
color\_codes= 不使用调色板而采用先前的 'r' 等色彩缩写

* darkgrid 黑色网格（默认）
* whitegrid 白色网格
* dark 黑色背景
* white 白色背景
* ticks 应该是四周都有刻度线的白背景

## 线性回归模型sns.lmplot（）函数

sns.implot是一种集合基础绘图与基于数据建立回归模型的绘图方法。旨在创建一个方便拟合数据集回归模型的绘图方法，利用'hue'、'col'、'row'参数来控制绘图变量。

seaborn.lmplot(x, y, data, hue=None, col=None, row=None, palette=None, col\_wrap=None, size=5, aspect=1, markers='o', sharex=True, sharey=True, hue\_order=None, col\_order=None, row\_order=None, legend=True, legend\_out=True, x\_estimator=None, x\_bins=None, x\_ci='ci', scatter=True, fit\_reg=True, ci=95, n\_boot=1000, units=None, order=1, logistic=False, lowess=False, robust=False, logx=False, x\_partial=None, y\_partial=None, truncate=False, x\_jitter=None, y\_jitter=None, scatter\_kws=None, line\_kws=None)  
'''  
hue, col, row:strings 定义数据子集的变量，并在不同的图像子集中绘制  
size:scalar 定义子图的高度  
markers:定义散点的图标  
col\_wrap:设置每行子图数量  
order:多项式回归，设定指数，可以用多项式拟合  
logistic:逻辑回归  
logx:转化为log(x)  
'''  
#例  
sns.lmplot(x="total\_bill", y="tip", col="day", hue="day",data=tips, col\_wrap=2, size=3)



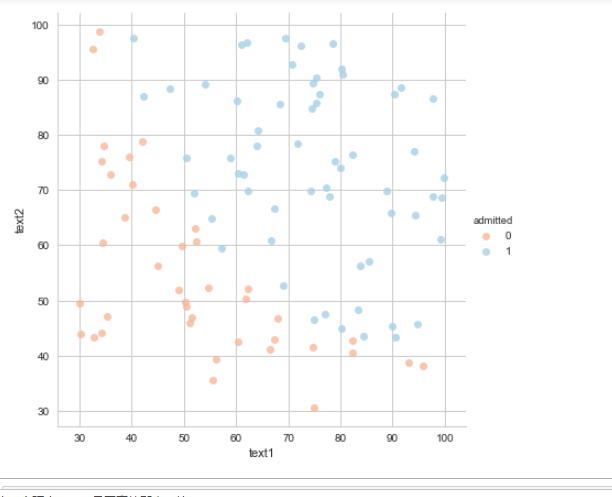
参考：https://www.cnblogs.com/fanghao/p/7500817.html

https://blog.csdn.net/cymy001/article/details/78420478

参数详细说明：http://seaborn.pydata.org/generated/seaborn.lmplot.html

吴恩达作业：

sns.set(context='notebook',style='whitegrid',palette=sns.color\_palette("RdBu", 2)) #RdBu为颜色设置参数，2为颜色个数  
sns.lmplot('text1','text2',data=data1,  
 hue='admitted', #定义数据子集的变量，绘制在图中  
 size=6,#定义子图中每一格尺寸的高度，单位应为英寸  
 fit\_reg=False, #不画拟合直线  
 scatter\_kws={"s": 50}   
 #附加的关键字参数以字典的形式传递给plt.scatter()函数  
 #将plt.scatter()的参数s的值改为50，默认值为20 ，参数s代表散点的大小  
 )  
plt.show()



### 画散点图plt.scatter()函数参数

参考：https://blog.csdn.net/qiu931110/article/details/68130199



### sns.color\_palette()调色板函数

sns.color\_palette(颜色设置，颜色个数)

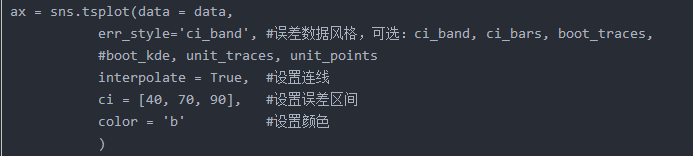
color*palette()能传入任何Matplotlib,不写参数则默认颜色，可以填写muted,RdBu,Blues*r,Set1,hls,Paired set\_palette()设置所有图的颜色

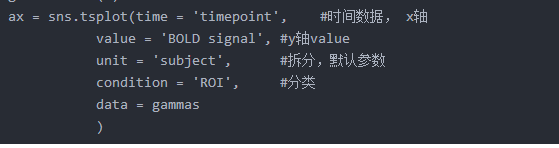
参考：https://www.jianshu.com/p/cce56332f80c

## tsplot()时间线图表

参考：https://blog.csdn.net/qq\_29831163/article/details/90437311

|  |  |
| --- | --- |
| 参数 |  |
| data | 作图的数据 |
| err\_style | 误差数据风格，可选：ci*band, ci*bars, boot*traces, boot*kde, unit*traces, unit*points |
| interpolate | bool，是否显示连线 |
| ci | 误差区间 |
| n\_boot | 迭代次数 |





#### plt.subplots()函数

|  |  |
| --- | --- |
| 参数 | 说明 |
| nrows | subplot的行数 |
| ncols | subplot的列数 |
| sharex | 所有subplot应该使用相同的X轴刻度（调节xlim将会影响所有subplot） |
| sharey | 所有subplot应该使用相同的Y轴刻度（调节xlim将会影响所有subplot） |
| subplot\_kw | 用于创建各subplot的关键字字典 |
| \*\*fig\_kw | 创建figure时的其他关键字，如plt.subplots(2,2,figsize=(8,6)) |

参考：https://blog.csdn.net/C\_chuxin/article/details/83994457

#### figsize参数

figsize=(a,b)

* ==figsize 设置图形的大小，a 为图形的宽， b 为图形的高，单位为英寸==

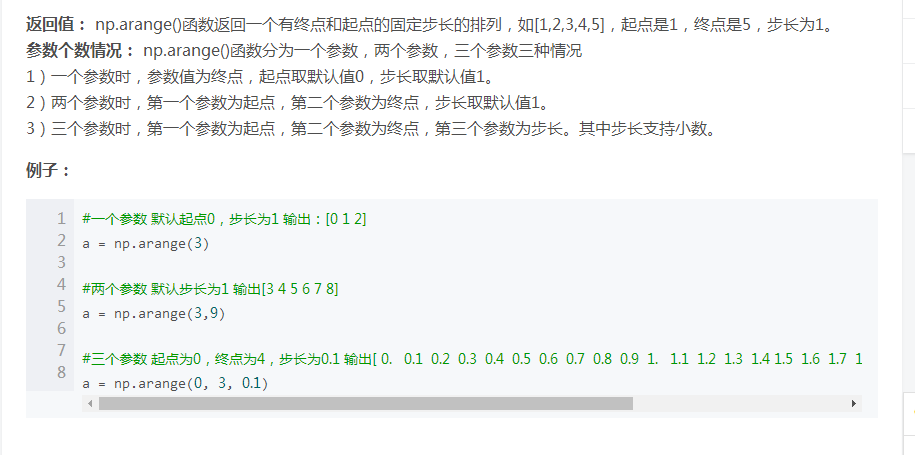
#### plot()函数



参考：https://blog.csdn.net/fljhm/article/details/88871831

https://blog.csdn.net/sinat\_36219858/article/details/79800460

### np.arrange()



### np.linspace()等差数列函数

## np.logspace（）等比数列函数

https://blog.csdn.net/shenpengjianke/article/details/29356755

### scatter函数

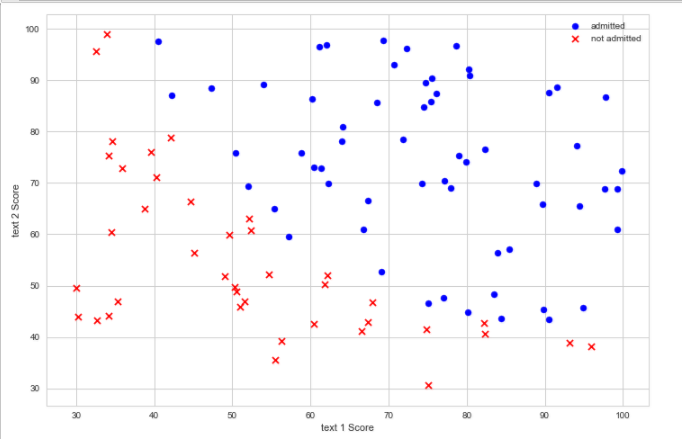


### 筛选数据函数isin()

参考：https://www.jianshu.com/p/805f20ac6e06

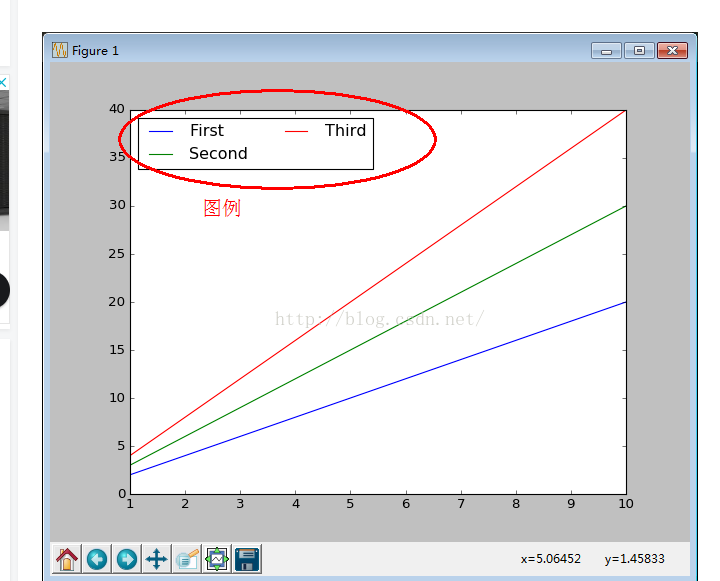
https://blog.csdn.net/lzw2016/article/details/80472649

positive = data1[data1['admitted'].isin([1])] #isin（）为筛选函数，令positive为数据集中为1的数组  
negative = data1[data1['admitted'].isin([0])] #isin（）为筛选函数，令negative为数据集中为0的数组  
  
fig, ax = plt.subplots(figsize=(12,8))#以其他关键字参数\*\*fig\_kw来创建图  
#figsize=(a,b):figsize 设置图形的大小,b为图形的宽,b为图形的高,单位为英寸  
ax.scatter(positive['text1'], positive['text2'], s=50, c='b', marker='o', label='admitted')   
#x为Exam 1，y为Exam 2，散点大小s设置为50,颜色参数c为蓝色，散点形状参数marker为圈，以关键字参数\*\*kwargs来设置标记参数labele是Admitted  
ax.scatter(negative['text1'], negative['text2'], s=50, c='r', marker='x', label='not admitted')  
#x为Exam 1，y为Exam 2，散点大小s设置为50,颜色参数c为红色，散点形状参数marker为叉，以关键字参数\*\*kwargs来设置标记参数labele是Not Admitted  
ax.legend() #显示图例  
ax.set\_xlabel('text 1 Score') #设置x轴变量  
ax.set\_ylabel('text 2 Score') #设置y轴变量  
plt.show()



### 显示图例函数legend（loc,ncol）

参考https://blog.csdn.net/you*are*my\_dream/article/details/53440964



# 计算代价函数

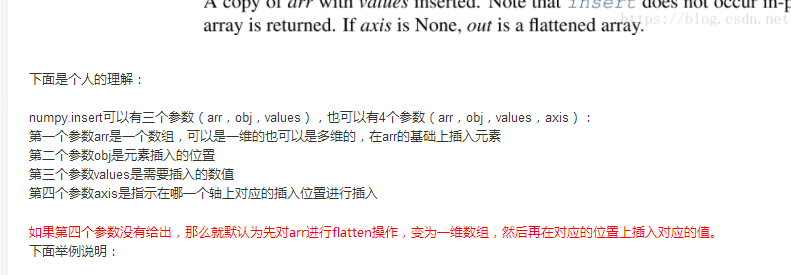
首先，我们将创建一个以参数θ为特征函数的代价函数

其中：

def computerCost (X,y,theta):  
 inner=np.power((X\*theta.T)-y,2)   
 #theta.T就是矩阵theta的转置矩阵  
 #np.power(A,B) ## 对A中的每个元素求B次方  
 return np.sum(inner)/(2\*len(X)) #np.sum()求和函数

# Pandas

#### insert()函数



axis=1为列插入，axis=0为行插入

参考：https://blog.csdn.net/lcxxcl\_1234/article/details/80869152

#### shape[]函数

获取dataFrame的行数和列数，使用的命令是：dataframe.shape[0]和dataframe.shape[1]  
此外，获取行数还有方法：len(DataFrame.index)

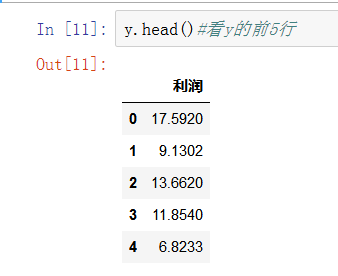
#### iloc()函数

用法见：https://blog.csdn.net/W\_weiying/article/details/81411257

吴恩达作业用到里面的第二点

#设置训练值变量X和目标变量y  
cols=df.shape[1] #获取表格df的列数  
X=df.iloc[:,0:cols-1] #除最后一列外，取其他列的所有行，即X为Theta0和人口组成的向量  
y=df.iloc[:,cols-1:cols]#取最后一列的所有行，即y为利润  
#注： A:B是索引切片功能

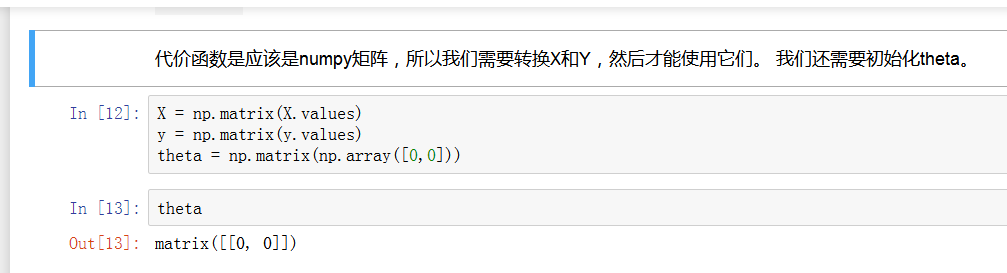




#### matrix()和array()函数

分别是矩阵处理函数和数组处理函数

参考：https://blog.csdn.net/qq403977698/article/details/47254539



#### numpy中的shape方法

矩阵.shape：查看矩阵的维数

参考：https://blog.csdn.net/u010758410/article/details/71554224

#### zero()函数

zeros(shape, dtype=float, order='C')：返回一个数目为shape的元素为0的数组

用法：https://blog.csdn.net/qq\_26948675/article/details/54318917

#### ravel()函数

功能：将多维数组转换为一维数组，并不会产生源数据的副本

参考：https://blog.csdn.net/tymatlab/article/details/79009618

#### .multiply()函数

功能：数组/矩阵对应位置相乘，输出与相乘数组/矩阵大小一致

参考：https://blog.csdn.net/zenghaitao0128/article/details/78715140

# batch gradient decent（批量梯度下降）对应笔P21页

## ${{\theta }*{j}}:={{\theta }*{j}}-\alpha \frac{\partial }{\partial {{\theta }\_{j}}}J\left( \theta \right)$

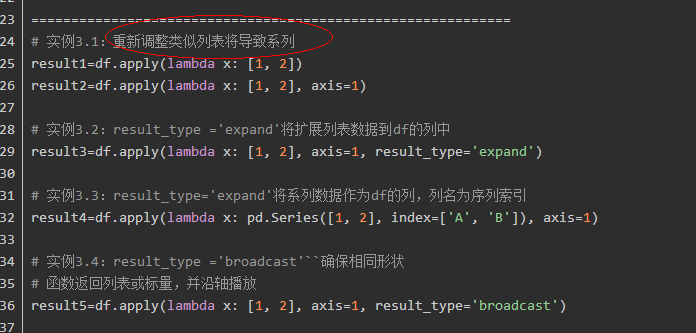
def gradientDescent(X,y,theta,alpha,iters): #alpha是学习率,iters为迭代次数  
 temp=np.matrix(np.zeros(theta.shape)) #np.zeros(theta.shape)=[0.,0.],然后将temp变为矩阵[0.,0.]  
 parameters= int(theta.ravel().shape[1])   
 #theta.ravel()：将多维数组theta降为一维，.shape[1]是统计这个一维数组有多少个元  
 #parameters表示参数  
 cost=np.zeros(iters) #初始化代价函数值为0数组，元素个数为迭代次数  
   
 for i in range(iters): #循环iters次  
 error=(X\*theta.T)-y  
   
   
 for j in range(parameters):  
 term = np.multiply(error, X[:,j]) #将误差与训练数据相乘，term为偏导数，参考笔记P27  
 temp[0,j] = theta[0,j] - ((alpha / len(X)) \* np.sum(term)) #更新theta  
   
   
 theta=temp  
 cost[i] = computeCost(X,y,theta) #计算每一次的代价函数  
   
 return theta,cost  
   
#初始化一些附加变量,比如学习速率α和要执行的迭代次数。  
alpha=0.01  
iters=1500  
  
现在让我们运行梯度下降算法来将我们的参数θ适合于训练集。  
g, cost = gradientDescent(X, y, theta, alpha, iters) #令g和cost分别等于函数的两个返回值  
g #输出g  
computeCost(X, y, g) #最小化的低价函数

### np.zeros和np.ones函数

参考https://blog.csdn.net/u014386899/article/details/82706470

# 特征缩放

### df.apply



def FeatureScaling(df):  
 # """Applies function along input axis(default 0) of DataFrame."""  
 return df.apply(lambda column: (column - column.mean()) / column.std()) #根据公式重新调整列表

### sklearn 线性回归LinearRegression()参数介绍

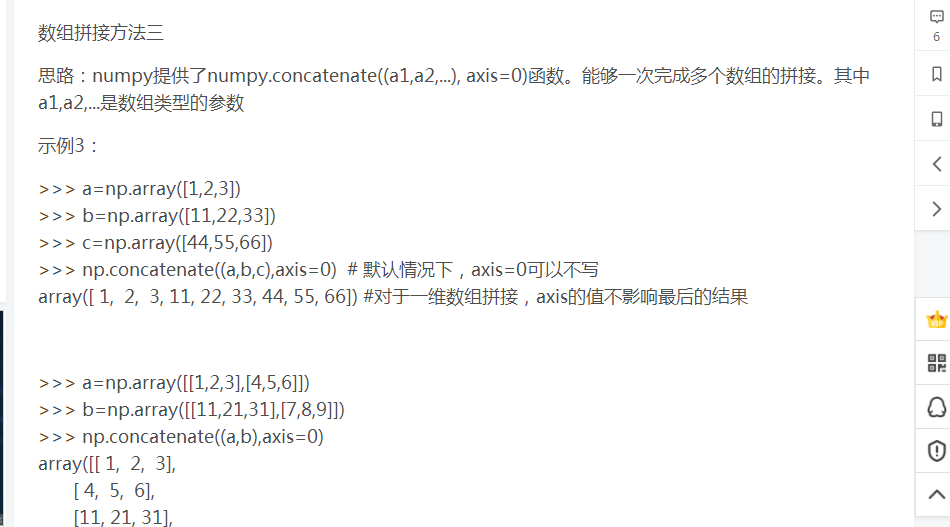


## sklearn.linear\_model.LogisticRegression接口参数

class sklearn.linear\_model.LogisticRegression(penalty='l2', dual=False, tol=0.0001, C=1.0,fit\_intercept=True, intercept\_scaling=1,class\_weight=None, random\_state=None,solver='liblinear', max\_iter=100, multi\_class='ovr', verbose=0, warm\_start=False, n\_jobs=1)

参数列表参考：https://blog.csdn.net/liyuanshuo\_nuc/article/details/82831633

### 数组拼接函数np.concatenate



参考：https://blog.csdn.net/zyl1042635242/article/details/43162031

### 排序函数np.sort()



参考：https://blog.csdn.net/haiyang\_duan/article/details/79221458

### 矩阵求逆函数np.linalg.inv（）

theta = np.linalg.inv(X.T@X)@X.T@y#X.T@X等价于X.T.dot(X) .dot()表示点积，也就是矩阵相乘的意思

参考：https://www.jianshu.com/p/0d5107596d86

### data.iloc[:, :-1].as\_matrix()函数

参考：https://blog.csdn.net/weixin\_41884148/article/details/88783328.

#### 画图函数.plot()

ax.plot(x变量，y变量，‘线条颜色和；类型’)

参考：https://www.cnblogs.com/nxld/p/7435930.html

### scipy优化算法包

#### 梯度下降法最小化代价函数常用到的fmin\_tnc()和minimize()函数

参考：https://www.cnblogs.com/tongtong123/p/10634716.html

### 数组中的数值类型转换函数np.astype（）



### sklearn库的评价函数classification\_report



参考：https://blog.csdn.net/akadiao/article/details/78788864

#### zip()打包对象函数



参考：https://www.runoob.com/python3/python3-func-zip.html

#### map()函数



参考：https://www.runoob.com/python/python-func-map.html

#### 格式化字符串函数format(）

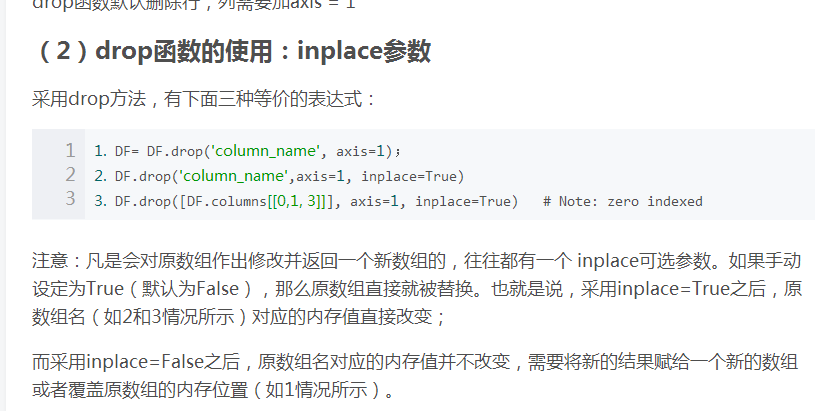


参考：https://www.runoob.com/python/att-string-format.html

theta\_min = np.matrix(result[0]) #将最优化最优参数中优化问题目标值数组x转化为矩阵  
predictions = predict(theta\_min, X)  
correct = [1 if ((a == 1 and b == 1) or (a == 0 and b == 0)) else 0 for (a, b) in zip(predictions, y)]  
#predictions为预测值,y为实际值.zip为打包对象函数，将(predictions, y)打包成元组，并赋值为（a,b），然后判断正确的则返回1，判断错误的返回0  
accuracy = (sum(map(int, correct)) % len(correct))  
# map(int, correct):将crrect列表内容的类型映射成int型，原来应该布尔型.sum为求和函数，%为求模运算，计算除法的余数。  
#实际是求1占数据总数的比例  
print ('accuracy = {0}%'.format(accuracy)) #format为格式化字符串的函数,{0}为设置指定位置.

设置轴范围的函数plt.xlim(起点，终点),plt.ylim(起点，终点)

### pandas的drop函数

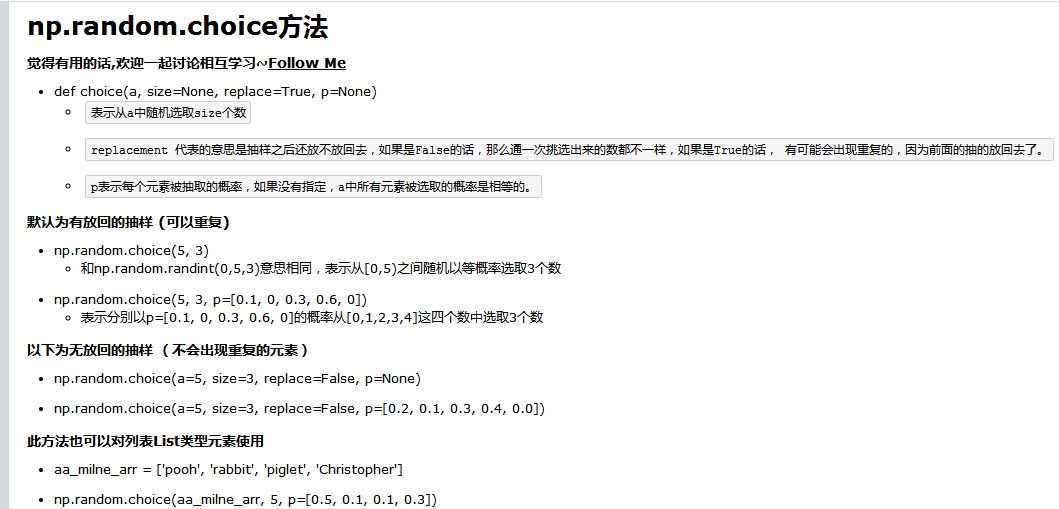


参考：https://blog.csdn.net/nuaadot/article/details/78304642

#### 读取mat文件（matlab数据）函数loadmat()

参考：https://blog.csdn.net/raby\_gyl/article/details/72540002

### np.random.choice方法



参考：https://www.cnblogs.com/cloud-ken/p/9931273.html

### np.arange函数()



### matshow()函数



参考：http://www.manongjc.com/article/111574.html

### 改变数组形状函数reshape()



参考：https://blog.csdn.net/DocStorm/article/details/58593682

### append函数



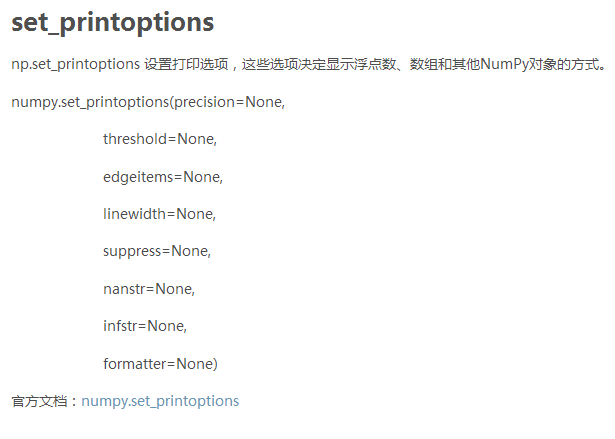
参考：https://blog.csdn.net/chiyiwei7384/article/details/83988712

### shape方法



参考：http://localhost:8888/notebooks/NG-machine%20learning/LY/%E5%A4%9A%E5%88%86%E7%B1%BB%E5%92%8C%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/%E5%A4%9A%E5%88%86%E7%B1%BB%E5%92%8C%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C.ipynb

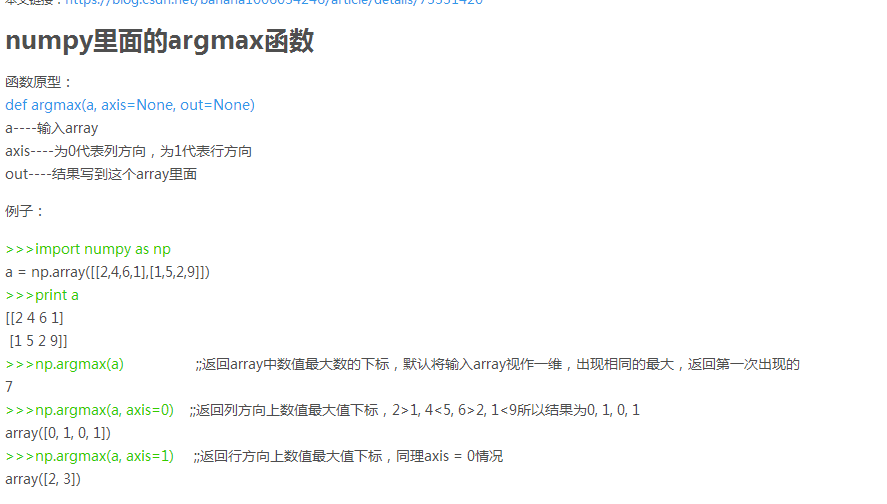
### np.set\_printoptions（）函数



参考：https://blog.csdn.net/tz\_zs/article/details/87105524

https://docs.scipy.org/doc/numpy/reference/generated/numpy.set\_printoptions.html

### argmax（）函数



参考：https://blog.csdn.net/banana1006034246/article/details/73331420

### copy（）函数

copy仅拷贝对象本身，而不对中的子对象进行拷贝，故对子对象进行修改也会随着修改。

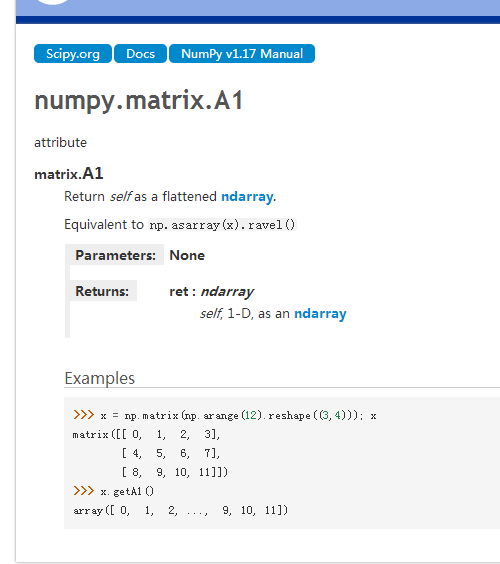
deepcopy是真正意义上的复制，即从新开辟一片空间。我们经常说的复制实际上就是deepcopy.

参考：https://blog.csdn.net/md945048125/article/details/78277552

### numpy.matrix.A1用法

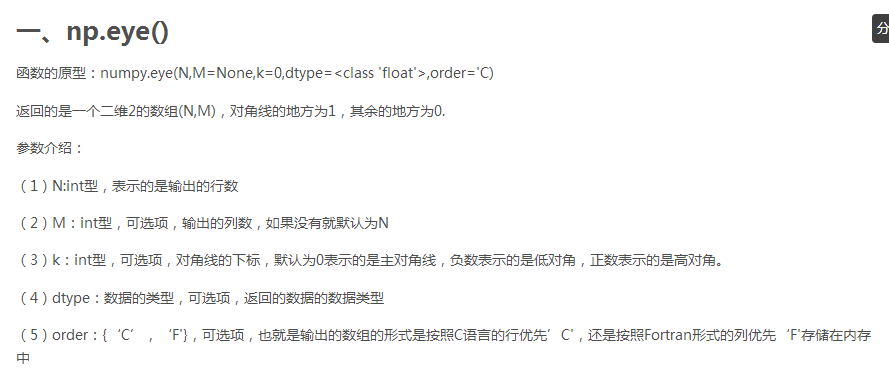
功能：把多维矩阵展开为一维矩阵

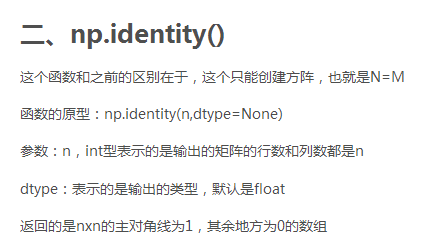
x = np.array(X[:, 1].A1) #X[:, 1]中':'表示选取所有行，1表示选取第1列 矩阵.A1是把多维矩阵展开为一维矩阵



参考:https://docs.scipy.org/doc/numpy/reference/generated/numpy.matrix.A1.html

### np.eye()和np.identity()



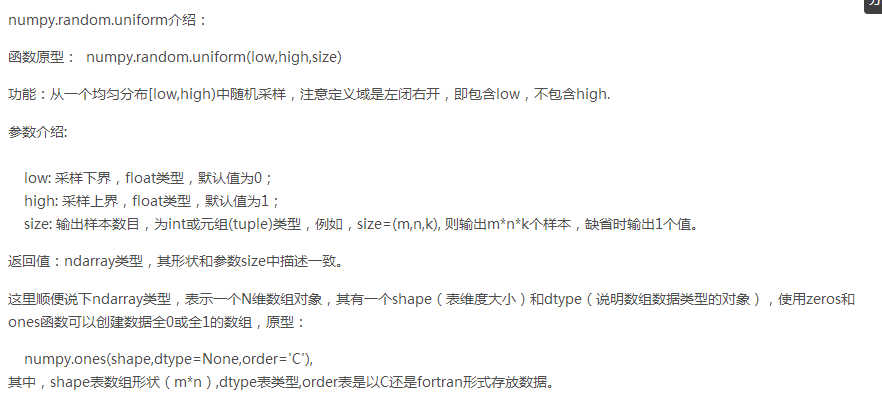
 参考:https://blog.csdn.net/m0\_37393514/article/details/81455915

### np.linalg.norm(求范数)

参考:https://blog.csdn.net/hqh131360239/article/details/79061535

### np.random.uniform()函数



# pandas.DataFrame()的参数解释



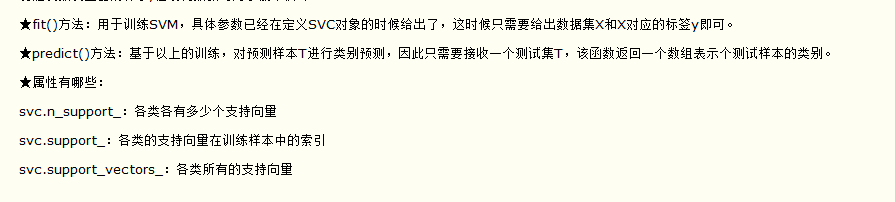
参考： https://blog.csdn.net/u011495642/article/details/82875438

### sklearn svm.LinearSVC()函数



参考：https://blog.csdn.net/weixin\_41990278/article/details/93165330

### sklearn-SVC.fit()和SVC.score()函数

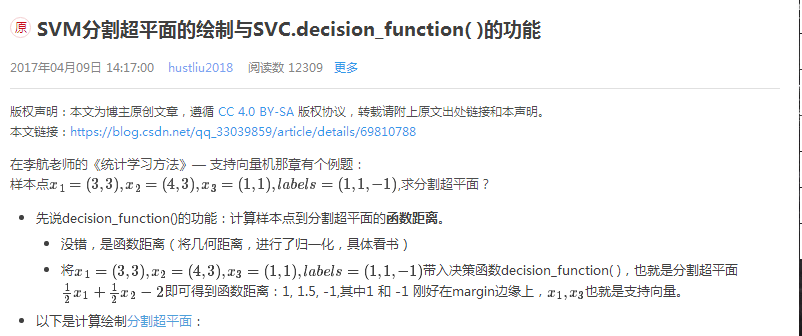


参考：https://www.cnblogs.com/xiaotan-code/p/6700290.html



参考：https://www.jianshu.com/p/a9f9954355b3?utm*campaign=maleskine&utm*content=note&utm*medium=seo*notes&utm\_source=recommendation

### SVM分割超平面的绘制与SVC.decision\_function( )的功能



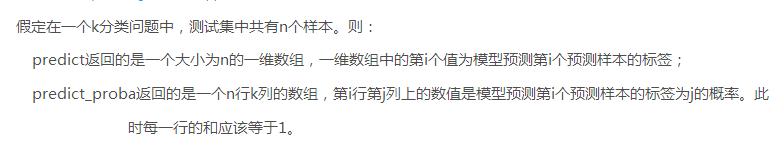
参考：https://blog.csdn.net/qq\_33039859/article/details/69810788

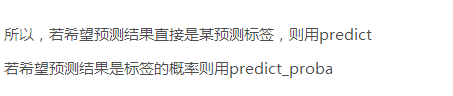
### sklearn系列之 sklearn.svm.SVC详解



参考:https://www.cnblogs.com/crawer-1/p/8870700.html

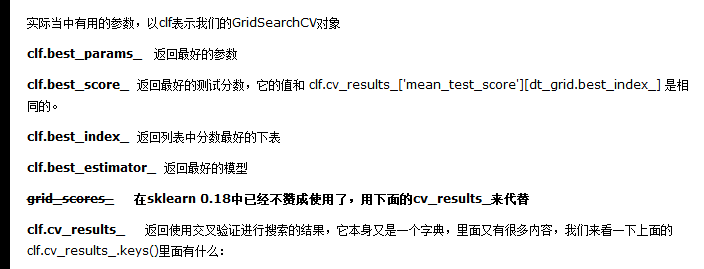
### sklearn中的predict与predict\_proba的区别





参考：https://blog.csdn.net/qq\_40875849/article/details/83052767

### sklearn中的超参数调节——best*score ,best*params



参考：https://www.cnblogs.com/jiaxin359/p/8641976.html

https://www.cnblogs.com/mdevelopment/p/9634629.html

### np.round()函数

round()方法返回 x 的小数点四舍五入到n个数字。



### np.where()用法总结

用法一：可以返回一个n维数组，可广播。  
# np.where(condition, x, y)

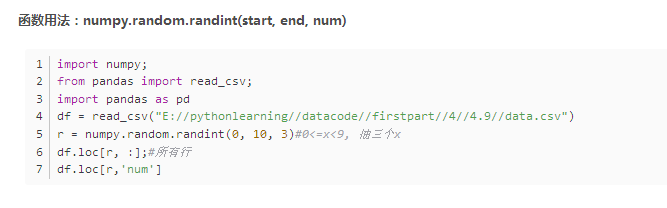
用法二：condition 也可以是布尔型数组，每个条件都和x,y对应（广播）

用法三：找到n维数组中特定数值的索引

参考：https://blog.csdn.net/zs15321583801/article/details/79645685

### 随机抽样函数np.random.randint（）





参考：https://blog.csdn.net/qq\_36330643/article/details/78315815

https://blog.csdn.net/W\_weiying/article/details/80390416

### plt.imshow()与plt.show()区别

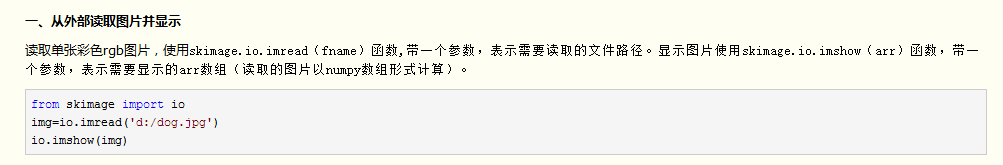
plt.imshow()函数负责对图像进行处理，并显示其格式，但是不能显示。

其后跟着plt.show()才能显示出来。

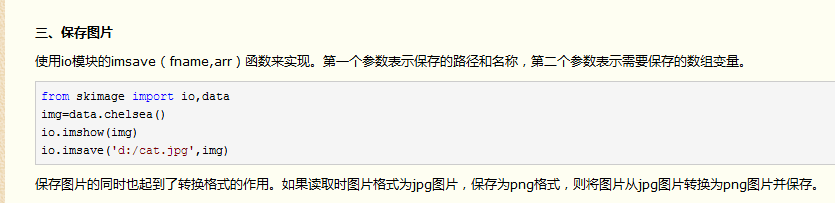


cmap:是颜色实例

### 图片的读取（imread）,显示（imshow）和保存（msave）

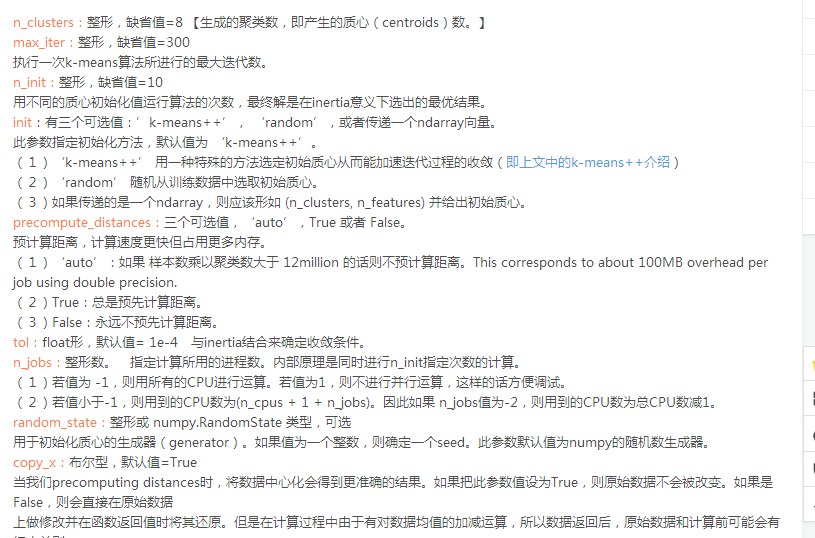






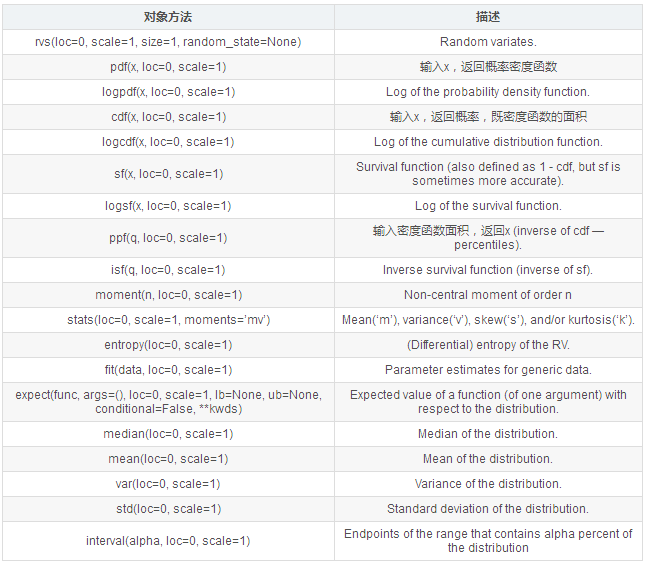
参考：https://www.cnblogs.com/denny402/p/5121897.html

### K-means和K-means++的算法原理及sklearn库中参数解释、选择



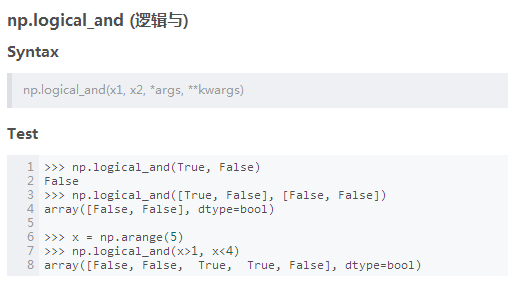
参考：https://blog.csdn.net/github\_39261590/article/details/76910689

### scipy.stats.norm函数



参考：https://blog.csdn.net/claroja/article/details/72830515

### 逻辑判断函数np.logical\_and/or/not (逻辑与/或/非)



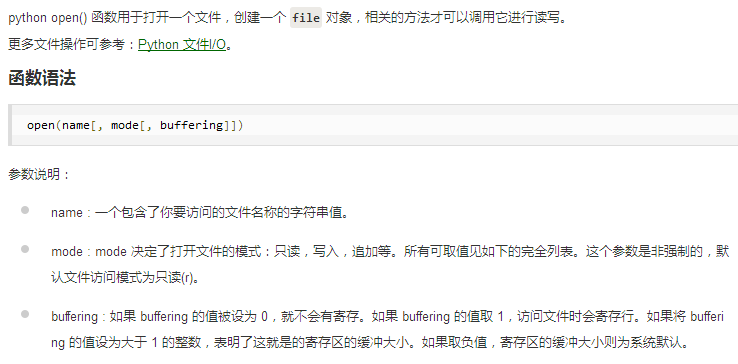
参考：https://blog.csdn.net/jningwei/article/details/78651535

### plt.tight\_layout()函数

功能:tight\_layout会自动调整子图参数，使之填充整个图像区域。这是个实验特性，可能在一些情况下不工作。它仅仅检查坐标轴标签、刻度标签以及标题的部分。

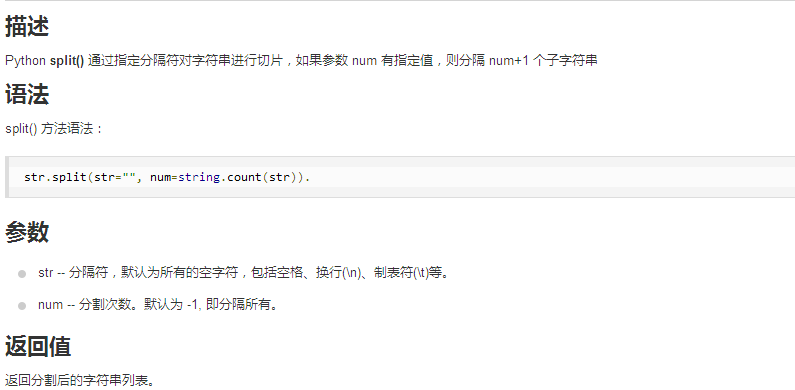
参考:https://blog.csdn.net/du\_shuang/article/details/84139716

### open() 函数



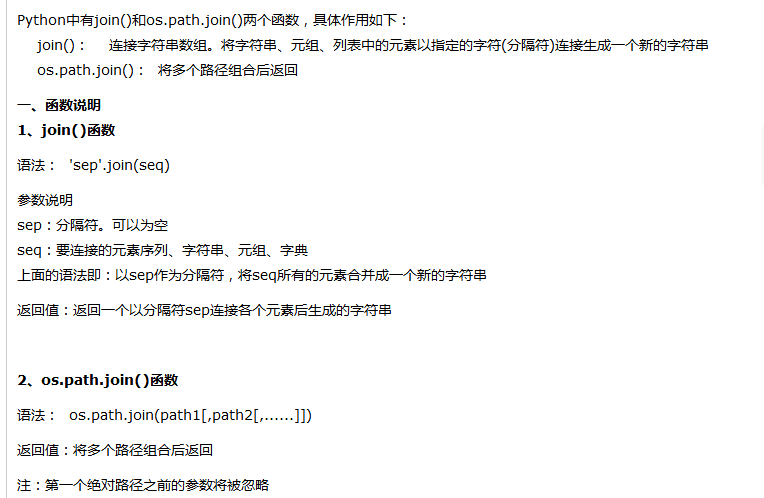
参考：https://www.runoob.com/python/python-func-open.html

### split()方法



参考：https://www.runoob.com/python/att-string-split.html

### Python中的join()函数的用法

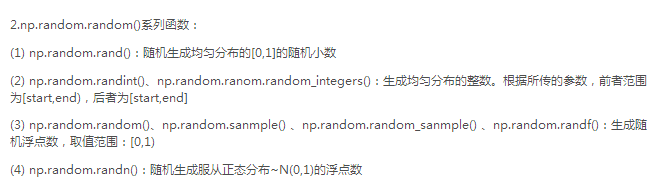


参考：https://www.cnblogs.com/jsplyy/p/5634640.html

#### Python中的axis=0,axis=1

axis=0表述列 axis=1表述行

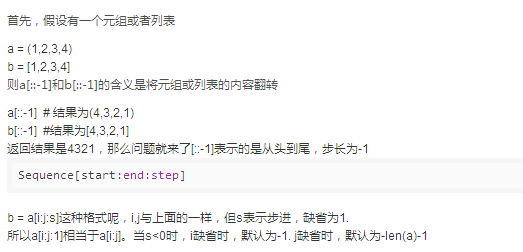
### np.random.random()系列函数



参考：https://blog.csdn.net/weixin\_41712499/article/details/82083517

https://www.cnblogs.com/DOMLX/p/9751471.html

### Python中[::-1]实现翻转列表的原理



参考：https://blog.csdn.net/qq\_39521554/article/details/79858227

https://blog.csdn.net/HARDBIRD123/article/details/82261651

### numpy中argsort函数用法

argsort函数返回的是数组值从小到大的索引值

参考：https://blog.csdn.net/maoersong/article/details/21875705