mRMR\_feature\_package.py脚本使用说明文档。

函数名称、参数列表及作用介绍：

1. csv\_converter(csvfilename,outputfilename)

参数：csvfilename,outputfilename。csvfilename为csv数据文件的地址，outputfilename为输出文件的地址，默认为本目录下的dataset.txt文件

返回值：无

作用：从指定csv文件中读取数据并且转后成mrmr算法程序可识别的文件格式后输出至outputfilename文件中。

1. mRMR(datasetFilename,FEAS)

参数：datasetFilename和FEAS。datasetFilename为存储了mrmr算法输入数据的文件，默认为本目录下的dataset.txt文件。FEAS为mrmr算法中我们要留下来的特征的数量，默认为180.

返回值：features。存储了mrmr算法筛选出的前FEAS个特征。

作用：调用mrmr算法筛选出我们所需要的特征。注：本程序中mrmr算法采用了C++语言实现，并且编译成可执行文件后直接利用shell命令调用。

1. OutputToFiles(features)

参数：features。features中存储了我们利用mrmr算法筛选出的特征的编号。

返回值：无

作用：将features中的特征存储到该目录下的Best\_Feas.txt文件中。

1. InputFromFiles(Best\_FeasFilename)

参数：Best\_FeasFilename。该参数是一个字符串，存储了mrmr算法所筛选出来的特征编号的文件的地址，默认为本目录下的Best\_Feas.txt文件。

返回值：features。features中存储了我们利用mrmr算法筛选出的特征的编号。

作用：从指定文件中读取我们所存储的mrmr算法筛选结果。

1. converter(fList,vec)

参数：fList和vec。fList是一个特征编号列表，vec是一个向量（list）。

返回值：一个list。

作用：将vec中所有在fList中的特征筛选出来，即将vec的所有特征中，属于list的特征提取出来，将这些特征重新组成一个向量。

1. mkdataset(datasetFilename,fList,OutputToFile)

参数：datasetFilename,fList,OutputToFile。datasetFilename是数据文件的地址，默认为本目录下的dataset.txt文件。fList是一个特征编号列表。OutputToFile只能取0或1，0表示结果不输出至文件，1表示结果输出至文件。

返回值：new\_posdata,new\_negdata。分别存储了我们根据所有数据和fList指定的特征所构造的用于svm训练和测试用的阴性和阳性的数据集。

作用：根据特征集fList，构建svm训练及评估所用的数据集

1. mkXYset(datapool,x,y,testset,label)

参数：datapool,x,y,testset,label。datapool存储了我们目前类型为label的所有数据。x、y、testset均为空列表。label代表了datapool中数据的类型（0或1）.

返回值：无

作用：程序会将构建好的训练集的数据存储到x中，标签对应的存到y中。testset会存储我们的测试集。

1. classify(Vec,Y)

参数：两个list，Vec和Y。Vec是训练的数据集合，Y是Vec中每条数据对应的标签。

返回值：clf，一个sklearn的LDA分类器。该分类器是根据传入的Vec和Y建立、训练的。

作用：根据给定数据构造LDA分类器。

1. get\_k\_fold\_Cross\_Validation\_classifier(Vec,Y,k)

参数：Vec,Y,k。Vec是一个list，存储了所有训练数据。Y是一个list，存储了Vec中每一条数据对应的标签。k代表k折交叉验证中的参数k。

返回值：clfs和testset。clfs是一个list，里面存储了k个已经训练好的sklearn分类器，这是k折交叉验证过程中建立的k个分类器。testset是一个list，里面存储了这k个分类器每一个分类器所用的测试集（每个测试集包含了测试用的数据向量以及该向量对应的标签）。

作用：实现k折交叉验证。返回k折交叉验证中用到的k个分类器和这些分类器所用的测试数据。

1. calcACC(clf,testx,testy)

参数：clf,testx,testy。clf是一个已经训练好的sklearn分类器。testx和testy是两个list，分别存储了测试该分类器用的训练数据以及对应的标签

返回值：一个数字：ACC。代表分类器clf在该测试集上的准确率。

作用：计算指定分类器在指定测试集上的运行准确率。

1. LDATesting(posData,negData,POSLABEL,NEGLABEL)

参数：posData,negData,POSLABEL,NEGLABEL。posData为进行LDA所用的阳性数据集，negData为进行LDA所用的阴性数据集。POSLABEL和NEGLABEL分别指定了阳性数据的标签和阴性数据的标签，默认POSLABEL为1，NEGLABEL为0.

返回值：ACC,MCC。ACC为LDA算法在该数据集上运行的准确率，MCC为LDA算法在该数据集上运行得到的的马修斯相关系数。

作用：在指定训练集上运行LDA算法+留一交叉验证，并计算ACC和MCC，以此来判断我们根据某指定特征集所构造的数据集上算法的运行效果。

1. ChooseFea(datasetFilename,fList,features,fea\_ACC,fea\_Set)

参数：

datasetFilename：程序运行所需的数据文件的位置，默认为本目录下的dataset.txt文件。

fList：目前的已挑选的特征的集合

features：所有可选特征的集合

fea\_ACC：记录算法运行过程中，LDA算法准确率的列表。

fea\_Set：存储了从算法运行开始至今我们所有出现过的fList，与fea\_ACC列表中的每一项都是一一对应的。

返回值：Best\_ACC。表示我们在当前已选特征列表的情况下，再加入一个新特征所能达到的最好准确率。

作用：根据准确率上升最大原则，挑选一个新的特征加入原有的特征集fList中，并且返回加入新特征后的准确率

1. IncSearch(datasetFilename,features)

参数：

datasetFilename：程序运行所需的数据文件的位置，默认为本目录下的dataset.txt文件。

features：所有可选特征的集合

返回值：fList。存储了我们利用增量搜索+LDA+留一交叉验证后所得到的最佳的特征集合。

作用 ：进行增量搜索，找到最佳的特征集合。

1. getVec(str)

参数：str：一个字符串，形式为[num1,num2,num3,..]的形式。

返回值：一个list，表示将str转化成的一的list

作用：将一个str转化为一个list

1. loadDataSet(filename)

参数：filename。为一个文件名。

返回值：dataset，为程序运行所用的数据集

作用：从文件中加载数据集，其中参数filename指定了文件名。

1. mkDataset\_forSVM(datapool,x,y,testset,label,RATE)

参数：

datapool：数据集

x和y：为空列表，分别用来存储训练集的数据及所对应的标签

testset：空列表，存储测试集

label：指定datapool中数据的类型

RATE：训练集占总数据集的比例，默认为0.8

返回值：无

作用：程序会将构建好的训练集的数据存储到x中，标签对应的存到y中。testset会存储我们的测试集。

1. Draw\_subplot(set1,lable1,set2,label2)

参数：

set1：数据集1

label1：数据集1中数据所对应的标签

set2：数据集2

label2：数据集2中数据所对应的标签

返回值：无

作用：将set1和set2分别画到不同的散点图中，点的颜色由label的值决定。

1. svm\_classifier(svmpos\_inFilename,svmneg\_inFilename)

参数：

svmpos\_inFilename：svm运行所用的阳性数据集

svmneg\_inFilename：svm运行所用的阴性数据集

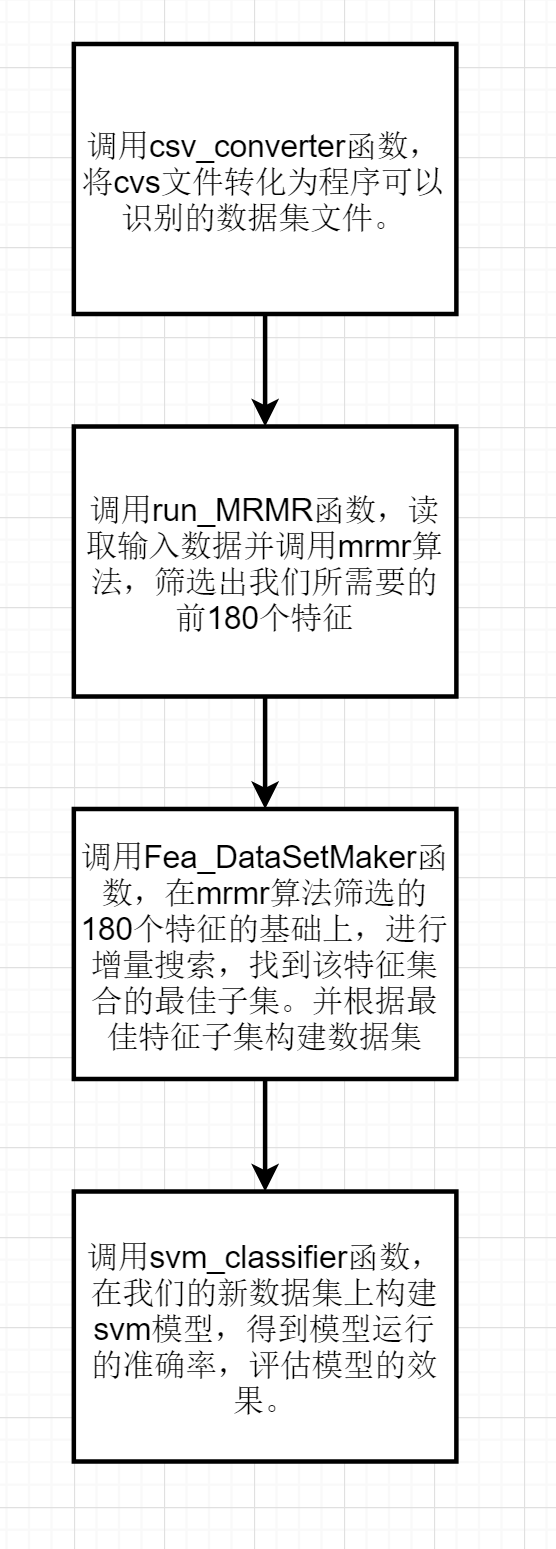
返回值：无

作用：将数据集中的数据划分为训练集和测试集，并且根据训练集训练svm，并在测试集上测试该svm，将最终结果以散点图形式表现出来。

**注：原论文中采用了mRMR特征筛选算法，后来，为了改进原有算法，我使用卡方检测代替原有的mRMR算法进行特征筛选，并且写出了改进后的代码，为了区别，我将卡方检测版本的程序所涉及到的函数名称后面都加上了Chi\_square以区分。因此InputFromFiles\_Chi\_square、ChooseFea\_Chi\_square、IncSearch\_Chi\_square、mkdataset\_Chi\_square函数分别对应于InputFromFiles、ChooseFea、IncSearch、mkdataset函数，对应的函数之间功能是相似的，因此在这里我就不再详细阐述了。**

**run\_MRMR、Fea\_DataSetMaker、Fea\_DataSetMaker\_Chi\_square、task2\_work是对整个程序流程的概括和总结，并不是程序运行所需要的核心函数。run\_MRMR函数封装了运行MRMR算法的各个步骤。Fea\_DataSetMaker、Fea\_DataSetMaker\_Chi\_square封装了增量搜索算法个各个步骤。task2\_work封装了整个程序的运行步骤。**

**程序运行流程图：**



**注：调用mrmr算法时package目录下必须要有mrmr可执行文件。**