主要的代码模块及功能汇总如下：

# 1，KoNLP.R

【主要功能】

利用KoNLP包，将文本文档分词，并转化为词频矩阵，并将有效词的词频矩阵存储至“newsdata.csv”。

【代码说明】

该段代码读入数据时牵扯到file.choose()函数，因此不适宜整段运行，需要一个一个文件选取，过程中注意不要重复和遗漏。

约在line424有设置停用词的语句，这句的编码格式很可能存在错误，打开是乱码情况，这时请从python文件复制停用词，或复制下面红色这句：stopWords =c("은", "는", "이", "가", "으로", "로", "에게", "게", "하게", "을", "를", "과", "와", "이여", "여", "대로",”1”,”2”,”3”,”4”,”5”,”6”,”7”,”8”,”9”,”,”,”.”,”;”,”:”) 粘贴至乱码处。

将数据全部读入并分词，对应统计频率后，组成矩阵，矩阵的200行代表200个文本，矩阵的每一列代表每个词，对应的数字意为该词在该文本中出现的次数。

在所有列前附加名为“TorF“的列，取值为0或1，取值为1表示该文本为TRUE文本，取值为0表示该文本为FAKE文本。

最终将文件存储在电脑上，以CSV格式，运行line644的getwd()函数可以查看存储位置。

【相关知识】

词频矩阵相关：<https://blog.csdn.net/gundanseed/article/details/16357353>

# 2，classifier.R

【主要功能】

基本分类器（分类器即指三种算法构成的程序）的测试，参数的选择/修正，绘制ROC曲线，计算AUC

【代码说明】

该模块使用未经处理的词频矩阵进行算法训练，运行至line20时，请将上步中得到的“newsdata.csv “文件存放在line20中getwd() 函数显示的路径。

Line26-Line76的#logit，#Bayes，#svm三个模块即调用调试好的算法，进行训练与预测，每个模块最后一句话会显示本算法在当前数据下的准确率，关于准确率的定义，参见下面的 相关知识

Line81后即为计算ROC曲线即AUC的模块

【相关知识】

ROC曲线及AUC的基本概念：<https://www.jianshu.com/p/2ca96fce7e81>

朴素贝叶斯入门：<https://blog.csdn.net/qiu_zhi_liao/article/details/90671932>

逻辑回归入门：

<https://blog.csdn.net/jacke121/article/details/79838312?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522159230736019725222432843%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=159230736019725222432843&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-1-79838312.first_rank_v2_rank_v25&utm_term=%E9%80%BB%E8%BE%91%E5%9B%9E%E5%BD%92>

SVM入门：

<https://blog.csdn.net/jiaoyangwm/article/details/81117958?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522159230342219195264539669%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=159230342219195264539669&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-6-81117958.first_rank_v2_rank_v25&utm_term=svm>

混淆矩阵与准确率：

<https://blog.csdn.net/qq_38194299/article/details/106557631?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522159230831419725211906873%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=159230831419725211906873&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-2-106557631.first_rank_v2_rank_v25&utm_term=%E6%B7%B7%E6%B7%86%E7%9F%A9%E9%98%B5>

# 3，TFIDF.R

【主要功能】

根据词频矩阵，计算TFIDF矩阵，并作为训练集测试算法，绘制ROC曲线，计算AUC

【代码说明】

关于TDIDF值的理解与计算参见下面的相关知识，这部分代码同上部分的娶别主要是将数据从词频矩阵，转换为了TFIDF矩阵，相应的正确率，ROC曲线及AUC值的计算大都相同

计算TDIDF矩阵的过程大致为：先建立运算过程中可能用到的中间变量，然后循环计算，将表格中的值转化为对应的TFIDF值

【相关知识】

TFIDF的相关具体概念可参考：

<https://blog.csdn.net/ShuiYuanShan/article/details/79871863?utm_medium=distribute.pc_relevant.none-task-blog-baidujs-2>

# 4, TureFalseRate.R

【主要功能】

根据TRUE/FAKE文本中词语的比例，对小于特定比例的词进行剥离，提高训练集特征强度

绘制ROC曲线，计算AUC

【代码说明】

在#main模块将原来的词频矩阵，根据提到的真/假文本中词的比例统计，删除掉比例小于一定值的列，让数据变得更有特征

后续模块与前几部分相同，是三个算法的测试及ROC曲线，AUC值的计算

# 5，kfold.R

【主要功能】

实现分为K组的交叉验证，绘制ROC曲线，计算AUC

【代码说明】

分为K组采用类似系统抽样的模式，即从第一行开始逐行抽取，逐个分组，最终将200行数据分为10组，每组20行，代表20个文档，通过180个文档集合（称为训练集）训练模型，通过20个文档几个（称为测试集）检测模型性能

【相关知识】

训练集与测试集/验证集

<https://blog.csdn.net/kieven2008/article/details/81582591?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase>

# 6. naiveBayes.R，svm.R，glm.R

算法代码，与其他代码放在同一路径即可运行

# 7，分析结果

使用KoNLP方法生成的基本词频矩阵，在逻辑回归LOGIT，支持向量机SVM，朴素贝叶斯NaiveBayes三种算法下的准确率分别是：0.99，0.87，0.58，对应的ROC曲线指AUC值分别为：0.99，0.9023，0.7627，AUC之和为0.6550

结果TFIDF方法整理数据后，LOGIT， SVM，Bayes方法的准确率变为0.78，0.795，0.965，可以看到LOGIT方法和SVM方法的准确率有所下降，Bayes方法的准确率大幅度上升，可以认为TFIDF数据非常适合Bayes算法，TFIDF下三种算法的AUC分别是：0.8543，0.8165，0.9650，AUC之和为0.6808

根据真/假新闻的特定此比率，在基本词频矩阵上做调整，删除某些列之后，LOGIT， SVM，Bayes方法的准确率变为0.99，0.93，0.49，可以看到SVM方法的性能大幅提升，LOGIT方法由于准确率基本达到顶峰，没有明显变化，而Bayes方法的准确率稍有下降，降至0.5以下，可以看到这种数据调整一定程度上不适合Bayes方法，调整数据下三种方法的AUC分别是：0.9900，0.9421，0.7385，AUC之和为0.6706

在基本算法的10倍交叉验证结果中：LOGIT， SVM，Bayes三种算法的准确率分别为0.90，0.75，0.65，可以看到相比基本方法LOGIT,SVM算法的准确率有所下降，可能是基础模型有过拟合现象，Bayes方法准确率上升，说明实际性能强于理论性能。10倍交叉验证条件下三种方法的AUC值分贝为：0.8990，0.8438，0.7812，AUC之和为0.5240

综合来看，使用TFIDF数据拥有更高的sumAUC，是综合条件下的最优选择