姓名和学号：李胜志 2210180232

章节名称：基于朴素贝叶斯算法的邮件分类

知识目标：掌握机器学习常用库的使用方法；使用朴素贝叶斯算法训练分类器。

能力目标：能够编写程序，使用朴素贝叶斯模型进行分类预测；能够使用朴素贝叶斯算法训练分类模型。

素质目标：养成分析问题、事前规划的良好习惯。

知识重点：使用jieba、re等库对数据进行预处理；使用Counter、map函数获取高频词的出现次数。

知识难点：机器学习常用库的使用方法；对邮件内容处理的流程和方法。

项目名称：基于朴素贝叶斯算法的邮件分类

项目目标：掌握机器学习常用库的使用方法；根据具体情境选择合适的朴素贝叶斯算法。

实验步骤：

1. 导入数据文件
2. 分别读取正常邮件和垃圾邮件的文件名
3. 读取停用词txt文件，并将其内容存到stopList列表里

# 导入相关的库

import os

import numpy as np

# 读取正常邮件的文件名列表

normalList = os.listdir('item5-ss-data/normal/')

# 读取垃圾邮件的文件名列表

spamList = os.listdir("item5-ss-data/spam/")

stopList = []

for line in open('item5-ss-data/stopwords.txt', encoding='utf-8'):

stopList.append(line[:len(line) - 1])

1. 数据预处理
2. 定义一个getwords函数，用于获取邮件的内容，返回结果并添加到allwordList列表里。
3. 使用Counter函数获取全部邮件中每个词出现的次数。
4. 使用most\_common(10)函数获取全部词中出现最多的十个词和次数。
5. 通过map函数获取邮件中出现频次最高的十个词在每封邮件出现的次数。
6. 输入训练集标签y：1表示正常邮件，0表示垃圾邮件，将高频词的次数做为训练集特征。

from jieba import cut

from re import sub

allwordList = [] # allwordList用来存放所有邮件的内容

# 定义一个getwords函数，用于获取邮件的内容

def getwords(path): # 参数path是邮件的完整路径

wordList = []

for line in open(path, encoding="utf-8"):

line = line.strip()

line = sub(r'[.，。/0-9【】、]','', line)

line = cut(line)

# 过滤长度小于1的词

line = filter(lambda x:len(x) > 1, line)

wordList.extend(line)

words = []

for word in wordList: # 过滤停用词

if word not in stopList and word != None: # 该词不在停用词列表里或者不是None值就添加到words列表中

words.append(word)

return words

for file in normalList:

normalListFile = ('item5-ss-data/normal/' + file)

words = getwords(normalListFile)

allwordList.append(words)

for file in spamList:

spamListFile = ('item5-ss-data/spam/' + file)

words = getwords(spamListFile)

allwordList.append(words)

from collections import Counter

from itertools import chain

# 获取邮件中出现频次最高的十个词

frep = Counter(chain(\*allwordList)) # 获取每个词出现的次数

topTen = frep.most\_common(10) # most\_common返回的是字典，对应的是出现次数最高的十个词和次数

topWord = []

for i in topTen:

topWord.append(i[0]) # 将出现次数最高的十个词添加到topWords列表中

vector = []

for word in allwordList:

temp = list(map(lambda x:word.count(x), topWord)) # 获取高频词在每封邮件出现的次数

vector.append(temp)

vector = np.array(vector)

# 1表示正常邮件，0表示垃圾邮件

y = np.array([1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0])

x = vector

1. 创建模型并训练

from sklearn.naive\_bayes import MultinomialNB # 导入多项式朴素贝叶斯

# 创建模型并训练

model = MultinomialNB()

model.fit(x, y)

1. 预测测试邮件的分类
2. 读取测试集邮件
3. 调用getWords函数对测试集邮件内容进行预处理
4. 获取10个高频词在测试集中出现的次数
5. 预测结果并输出

testList = os.listdir('item5-ss-data/test/')

# 测试邮件的内容处理

for testFile in testList: # 读取测试邮件的文件名列表

testListFile = ('item5-ss-data/test/' + testFile) # 用来存放测试邮件完整的路径

words = getwords(testListFile)

temp = list(map(lambda x:words.count(x),topWord))

test\_x = np.array(temp)

result = model.predict([test\_x])

if result == 1:

print(testFile,'是正常邮件')

else:

print(testFile,'是垃圾邮件')

输出结果：

normal-test.txt 是正常邮件

spam-test.txt 是垃圾邮件