赛区评阅编号（由赛区组委会填写）：

2018 高教社杯全国大学生数学建模竞赛

承　诺　书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》(以下简称为“竞赛章程和参赛规则”,可从全国大学生数学建模竞赛网站下载)。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或资料（包括网上资料），必须按照规定的参考文献的表述方式列出，并在正文引用处予以标注。在网上交流和下载他人的论文是严重违规违纪行为。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D 中选择一项填写）：所有

我们的报名参赛队号（12 位数字全国统一编号）：104

参赛学校（完整的学校全称，不含院系名）：华南理工大学

参赛队员 (打印并签名) ：1. 石望华

2. 束航

3. 赵康铭

指导教师或指导教师组负责人 (打印并签名)：刘深泉

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期: 2018 年 11 月 16 日

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

赛区评阅编号（由赛区组委会填写）：

2018 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅记录（可供赛区评阅时使用）：

评阅人						
备注						

送全国评阅统一编号（由赛区组委会填写）：

全国评阅随机编号（由全国组委会填写）：

（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）

适用于恐怖袭击事件的危害程度等级分类模型

摘要

评估恐怖袭击事件的危害程度并将其按等级分类是一项无监督学习的任务。在没有目标特征、目标函数的背景下，本文基于数据分析的量化分级模型，通过对 GTD 全球恐怖主义数据库变量分级、分权重的筛选，得到初始训练集、变量等级分类列表（详见支撑材料中“训练集.xlsx”、“变量等级分类 (含权重).txt”等文件），在此基础上对筛选出的变量特征从数据集记录分布情况、特征取值范围及其实际含义的角度逐一进行预处理，并进行可信度评估，得到数据预处理事务列表、预处理训练集、特征系数记录处理表（详见支撑材料中“变量等级分类 (含权重).txt”、“训练集 _ 预处理.xlsx”、“权重因子及可信度参数归一化处理.xlsx”文件），最后在训练集上进行系数与特征比重值相乘累加并排序，得到最终的突发事件五级分类结果和危害程度最高的十大事件（详见“训练集_分数表.xlsx”文件）。

关键字： 等级分类　特征筛选　数据预处理　权重因子　可信度分析

目录

一、 问题背景

股票市场的波动牵动千家万户，预测股价的表现更是专家追求的目标，经典的理论分析几乎没有什么实用价值，主要原因是突发事件和新闻数据迅速改变股票的正常走势，片刻扭转投资者的盈利和亏损状况。由于人工智能的发展，大数据科学的渗透，如何利用新闻数据和突发事件的内容来预测股价表现，让投资者做出更快更好的投资决策，已经成为金融，数学和人工智能领域等行业精英猎取的目标。

二、 问题重述

2.1 恐怖事件危害程度评估

恐怖突发事件影响社会的多个方面，依据附件 1 的信息，基于数据分析的量化分级模型，将附件 1 给出的 114183 个突发事件按危害程度从高到低分为一至五级，并列出近二十年来危害程度最高的十大恐怖袭击事件。

2.2 恐怖事件对股票走势的影响评估

恐怖事件影响股票市场的走势，试用模型分析问题 1 中的十大恐怖袭击事件对股票市场或者单支股票的影响，股票数据需自行收集。

2.3 突发新闻对股票走势的影响评估

新闻数据对股票市场的影响十分巨大，对特定的一类突发新闻数据，建模分析该突发新闻数据对股票价格的影响，并提出一个股票预测方案。

2.4 基于新闻图片识别的股票走势影响评估

突发新闻数据集中体现在新闻图片上，能否利用人工智能的图像识别技术，快速准确给出股票价格走势的预测。确定闪电交易投资决策，开创性推动投资行业的革命。

三、 问题分析

3.1 恐怖事件危害程度问题分析

针对本问题有提供一个 114184 行的数据集，包含对近 20 年发生的恐怖事件的详细记载，但仔细查看会发现有很多列对评估危害程度几乎没有贡献，很多列对评估危害程度的贡献很大，但参差不齐，存在很多空行或者不确定的记录。另一方面，危害程度不是一个很具体明确的概念，没有详细的评估准则，需要自行拟定评估标准。基于这两个突破点，本队决定结合数据分析的量化分级模型，构造合适的针对恐怖事件危害程度的评估模型，并不断优化，以准确得出事件五等级分类结果排名和十大危害恐怖事件。

3.2 恐怖事件对股票走势的影响问题分析

针对本问题，在通过第一题筛选后，我们提取出了近二十年来危害程度最高的十大恐怖事件，然而题目并没有提供有效的股市数据集，通过网络信息检索，在国家统计局和 UNSD（联合国统计司）找到相关股市数据后，决定采用机器学习中无监督学习方法 Clustering 来初步判断恐怖事件对股市变化的影响因素，之后通过 SVM 或 DNN 来进行时序 feature 的预测，之后通过 AdaBoost 分类器不断迭代优化模型，以验证规律的有效性。

3.3 突发新闻对股票走势的影响问题分析

针对本问题，类似第二题的问题解决办法，通过对一些主流新闻网页的信息爬取，获得相关数据集，预处理之后将新闻数据集与对应的股票信息数据集依据时间特征合并，通过对新闻特征分类并与股票变化信息关联，再用处理后的数据集投入 DNN 中进行有监督训练，得到验证率较高模型，再用 AdaBoost 进行迭代优化，获得验证率较高的预测模型。

3.4 基于新闻图片识别的股票走势影响问题分析

针对本问题，最重要的点在于从图片中识别出有效的特征并与 3.3 中数据集中的特征相匹配，完成特征筛选后用第三题中训练好的模型求解即可。因此，需先从新闻图片中收集、提取出可区分新闻类型与重要性的特征标签，特征提取后成为类似第三题中的新闻数据集，进行关联股票数据集分析。

四、 模型的假设

1. 赛题提供的数据集中所有数据都正确或者对评估的干扰可忽略，且每一行代表一个不同的独立的事件
2. 忽略不同年代信息传播效率不同导致恐怖事件传播对事件危害评估的影响
3. 忽略不同发生日期（节日等）对恐怖事件重要性的影响
4. 忽略恐怖事件声明负责团体对恐怖事件重要性的影响
5. 假设模型只考虑数值型特征变量
6. 变量说明文档不完善导致不能考虑相关变量（如 individual 变量）的危害程度，排除忽略不明确变量带来的误差

五、 符号说明

符号	意义
n	事件件数：114183
m	训练集中特征个数：18
α	特征变量的权重因子
β	特征变量的可信度度量参数
p	特征变量的最终系数
features_name[j]	训练集中第 j 个特征的名称（即第 1 行第 j+1 列）
features_weight[j]	训练集中第 j 个特征的权重
x[i][j]	训练集中第 i 个事件的第 j 个特征变量的值
rank[i][j]	训练集中第 j 个特征下第 i 个事件的排名名次
scores[i][j]	训练集中第 i 个事件在第 j 个特征上的分数
total_scores[i]	第 i 个事件的最终评估分数

六、 模型建立与评估

6.1 “排名”模型

题目要求评估 11 万多个事件的危害程度并将其分成五个等级，最终选出 10 个事件，我们自然地想到可以构造一个分数排名模型：提取出对危害评估有贡献的特征 features_name，在每一个特征上对所有事件的取值进行排序，由于每一个特征的取值范围不一，考虑到每一列排名的取值范围是特定的（1-114183），可以用行号代替，而且排名是名次越小代表成绩越好，即危害越大，所以在每一列上取名次 rank[i][j] 乘以这一列的权重 features_weight[j]，从而得到这一事件的 scores[i][j]，即第 i 个事件在第 j 个特征上的分数，最后把每个特征上的分数相加，得到总分 total_scores[i]。用公式表示如下：

$$total\_scores[i] = \sum_{j=1}^{m} rank[i][j] * features\_weight[j] \tag{1}$$

对 total_score 进行排序即可得到预想的结果。但刚开始执行这个模型就遇到了许多问题，比如：

1. 排名是个等差数列，体现不出一个特征下行与行之间的差异性，比如第一名与第二名之间的差距很大，第二名和第三名差距很小时，这一模型还是使得第一二名之间的差分与第二三名之间的差分相同，即都是相差一个特征权重 features_weight[j]。
2. 排名并不一定数值越低越好，大多特征是值越大越好（这个问题可通过预处理将最大值设置为第一名即在 excel 表上倒序排序得到解决）
3. 这种模型得出的结果是分数越低越好，有违背常理，必须在处处保证分数越低越好的同时保证等级越高的特征权重越大，这就给特征变量的权重的设置带来了难题：如果依据前面越小越重要的思想，把特征变量权重越小表示对危害评估的贡献越大，那么对危害评估贡献越小的特征的权重大，它们对事件的加分大于贡献大的特征的加分，也就是贡献低的特征反而对分数的影响占据了主要位置，这显然是不合理的模型；如果特征变量的权重越大表示对危害程度的贡献越大，那么在排名越低越重要、权重越大越重要、分数越低越重要的同时如何保证特征变量权重之间的比例设置合理不至于让低贡献特征变量的加分基本起不了影响、调节的作用依旧是一个难点。

6.2 “高考-学分”模型

基于“排名”模型依据分数越低危害程度越大的问题，我们结合其优缺点，调整模型，在保证依旧是通过设计权重，与特征变量值相乘，再累加求分数的大方向上，大胆调整模型，得出了“高考-学分”模型：

1. 类比高考分数越高越好，事件的最终得分设置为越高危害越大；
2. 类比每一科目都有一个满分，给每一个特征值设定一个满分，每个事件依照自己在每一特征下的得分除以相应满分得到一个百分比；
3. 类比大学里不同学科学分不同导致绩点不同，将每一事件在每一特征下的百分比乘以这一特征的权重。

通过以上三步得到每个事件在每个特征下的分数，其中每个特征下的满分是取的该特征下的最大值，基于此，再对每一特征求和，最后倒序排序，可得到比较合理的最终结果。本模型可用公式表示如下：

$$total\_scores[i] = \sum_{j=1}^{m} x[i][j]/max(features\_name[j]) * features\_weight[j] \quad (2)$$

其中 max(features_name[j]) 表示第 j 个特征变量下事件的最大取值。

本模型在前一模型的基础上有了较大改进，克服了不同事件同一特征下不能体现事件之间的差距的问题，但仔细推敲，发现与前一模型相比，本模型改变了特征之间即列与列的权重。比如某个重要特征下的最大值可能很大，但大多数行的取值都是比较小的，这导致这一列的加分上个别事件取值远大于其他事件，与此同时如果有某个不重要的特征变量下行的取值范围比较小，大多数事件都加上了近似的且趋近与满分的得分，这显然很有可能导致重要特征的影响地位减弱，从而危害大的事件由于在某个重要特征下取值不突出而被弱化，得不到应有的加分，并在其他列被危害较之偏小的事件超越。基于这个考虑，我们提出了最后的模型——“四六级”模型。

6.3 “四六级”模型

我国英语四六级考试的每一题的分数每年都是不固定的，总原则简单地描述就是让每个分段的人数比例保持每年一致。换一种说法，就是如果每年考试人数不变，则每年所有考四六级的人的总分是一个恒定值。类比这个思路，尝试对每一个特征下所有事件的得分之和为一定值，这样就容易控制列与列之间的权衡。为达到这个目的，可以把每一个特征下每一个事件的加分设置为事件取值除以本特征下所有有效事件的和，用公式表示如下：

$$total\_scores[i] = \sum_{j=1}^{m}(x[i][j]/(\sum_{i=1}^{n} x[i][j])) * features\_weight[j] \quad (3)$$

本模型可以保证同一特征下不同事件的得分是与取值（危害程度）成比例的，也可以保证每一特征下所有事件的和是可控的定值，兼顾了前两个模型的优点，并巧妙地避开了它们的不足。我们队采纳了这个模型进行评估，最终，也得到了不错的结果。

七、 数据集处理

数据集处理包含三部分：训练集构造，训练集预处理，计算总分。训练集是指用来评估事件危害程度的数值型有用属性集合，赛题提供的数据参差不齐，需要进行大量的人工筛选工作，把一个个没用的、不适合模型的特征变量去除，我们以附件三为入手点，可将数据集分成六大类，即六个特征等级，再根据附件二对每一个等级内部进行二级划分，每一个等级内部的二级等级共同分享给这一等级分配的分数（权重）；训练集预处理是指对特征变量权重的合理设计以及可信度分析，每一个等级的级数代表本等级的权重，引入 α 权重因子代表每一等级内部各特征所占这一等级的权重，每一等级内部 α 的和是 1，针对空值、不确定记录引入可信度因子 β 表示特征的可信程度。两个部分详细步骤如下：

7.1 构造训练集

1. 从附件三中筛选出 6 类特征，具体请见支撑材料“终版 _ 标红 + 分级 _ 特征变量 _ 附件 3+ 附件 2 内容摘要.doc”文件
2. 从附件二中筛选出 6 类特征下的有用的合适的 18 个特征，具体请见支撑材料“终版 _ 标红 _ 特征属性 _ 附件 2+GTD 全球恐怖主义数据库变量说明”文件
3. 从附件一中筛选出相应特征，得到训练集，具体请见支撑材料“训练集.xlsx”文件

7.2 训练集预处理

1. 对 18 个特征设置权重 α，具体情况见“变量等级分类 (含权重).txt”文件（其中包含数据筛选原则）
2. 处理空值和不确定记录（比如-99、-9 等），并计算 β 可信度因子，具体情况请见“分类后 _ 数据预处理.txt”文件（其中包括处理方法）
3. 对前两步得到的 α 和 β 进行归一化处理，具体处理步骤请见“权重因子及可信度参数归一化处理.xlsx”文件

7.3 计算总分

依据“训练集.xlsx”文件和“权重因子及可信度参数归一化处理.xlsx”文件计算乘积和，排序后得到“训练集 _ 分数表.xlsx”文件，第一列表示等级序号，第 2-22838 行是第 I 级，第 22839-45675 行是第 II 级，第 45676-68512 行是第 III 级，第 68513-91348 行是第 IV 级，第 91349-114184 行是第 V 级，倒数第二列是总分，最后一列在倒数第二列基础上乘以了 100000（为了方便查看与比较）。

参考文献

[1] @online ALTauthor = Qling, title = 数学建模方法-多属性决策模型, date = 2018-7-12, url = http://www.cnblogs.com/Qling/p/9295414.html

附录 A region 数据可视化

```
# encoding=utf-8
import xlrd
import numpy as np # useful for many scientific computing in Python
import pandas as pd # primary data structure library
df_can = pd.read_excel(r'C:\Users\14045\Desktop\shulidasai\dataset1.xlsx',
#sheet_name='Canada by Citizenship',
#skiprows=range(20),
#skipfooter=2
)
print('Data downloaded and read into a dataframe!')
print(df_can.shape)




# for sake of consistency, let's also make all column labels of type string
df_can.columns = list(map(str, df_can.columns))

df_can.drop(['eventid', 'iyear', 'imonth', 'iday', 'approxdate', 'extended',
'resolution', 'region',
'addnotes', 'scite1', 'scite2', 'scite3', 'dbsource', 'INT_LOG',
'INT_IDEO', 'INT_MISC', 'INT_ANY', 'related', 'country','country_txt', 'provstate', 'city',
    'latitude', 'longitude', 'specificity', 'vicinity', 'location', 'summary', 'crit1',
    'crit2', 'crit3', 'doubtterr', 'alternative', 'alternative_txt', 'multiple', 'success',
    'suicide', 'attacktype1', 'attacktype1_txt', 'attacktype2', 'attacktype2_txt',
    'attacktype3', 'attacktype3_txt', 'targtype1', 'targtype1_txt', 'targsubtype1',
    'targsubtype1_txt', 'corp1', 'target1', 'natlty1', 'natlty1_txt', 'targtype2',
    'targtype2_txt', 'targsubtype2', 'targsubtype2_txt', 'corp2', 'target2', 'natlty2',
    'natlty2_txt', 'targtype3', 'targtype3_txt', 'targsubtype3', 'targsubtype3_txt', 'corp3',
    'target3', 'natlty3', 'natlty3_txt', 'gname', 'gsubname', 'gname2', 'gsubname2', 'gname3',
    'gsubname3', 'motive', 'guncertain1', 'guncertain2', 'guncertain3', 'individual',
    'nperps', 'nperpcap', 'claimed', 'claimmode', 'claimmode_txt', 'claim2', 'claimmode2',
    'claimmode2_txt', 'claim3', 'claimmode3', 'claimmode3_txt', 'compclaim', 'weaptype1',
    'weaptype1_txt', 'weapsubtype1', 'weapsubtype1_txt', 'weaptype2', 'weaptype2_txt',
    'weapsubtype2', 'weapsubtype2_txt', 'weaptype3', 'weaptype3_txt', 'weapsubtype3',
    'weapsubtype3_txt', 'weaptype4', 'weaptype4_txt', 'weapsubtype4', 'weapsubtype4_txt',
    'weapdetail', 'nkill', 'nkillus', 'nkillter', 'nwound', 'nwoundus', 'nwoundte',
    'property', 'propextent', 'propextent_txt', 'propvalue', 'propcomment', 'ishostkid',
    'nhostkid', 'nhostkidus', 'nhours', 'ndays', 'divert', 'kidhijcountry', 'ransom',
    'ransomamt', 'ransomamtus', 'ransompaid', 'ransompaidus', 'ransomnote', 'hostkidoutcome',
    'hostkidoutcome_txt', 'nreleased'], axis=1, inplace=True)
#print(df_can.columns)
list2=list(map(str, df_can.columns))
print(list2)
```

```
# set the country name as index - useful for quickly looking up countries using .loc method
df_can.set_index('region_txt', inplace=True)
df_can['Total'] = df_can.sum(axis=1)




import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.style.use('ggplot') # optional: for ggplot-like style

# check for latest version of Matplotlib
print('Matplotlib version: ', mpl.__version__) # >= 2.0.0
#pie chart
"""
A pie chart is a circualr graphic that displays numeric proportions by dividing a circle (or
    pie) into proportional slices. You are most likely already familiar with pie charts as it
    is widely used in business and media. We can create pie charts in Matplotlib by passing in
    the kind=pie keyword.

Let's use a pie chart to explore the proportion (percentage) of new immigrants grouped by
    continents for the entire time period from 1980 to 2013.

Step 1: Gather data.

We will use pandas groupby method to summarize the immigration data by Continent. The general
    process of groupby involves the following steps:

Split: Splitting the data into groups based on some criteria.
Apply: Applying a function to each group independently:
.sum()
.count()
.mean()
.std()
.aggregate()
.apply()
.etc..
Combine: Combining the results into a data structure.
"""
# group countries by continents and apply sum() function
df_continents = df_can.groupby('region_txt', axis=0).count()

# note: the output of the groupby method is a `groupby' object.
# we can not use it further until we apply a function (eg .sum())
```

```
print(type(df_can.groupby('region_txt', axis=0)))

df_continents.head()
"""
Step 2: Plot the data. We will pass in kind = 'pie' keyword, along with the following
    additional parameters:

autopct - is a string or function used to label the wedges with their numeric value. The label
    will be placed inside the wedge. If it is a format string, the label will be fmt%pct.
startangle - rotates the start of the pie chart by angle degrees counterclockwise from the
    x-axis.
shadow - Draws a shadow beneath the pie (to give a 3D feel).

# autopct create %, start angle represent starting point
df_continents['Total'].plot(kind='pie',
figsize=(5, 6),
autopct='%1.1f%%', # add in percentages
startangle=90,  # start angle 90° (Africa)
shadow=True,    # add shadow
)

plt.title('Immigration to Canada by Continent [1980 - 2013]')
plt.axis('equal') # Sets the pie chart to look like a circle.
plt.show()
"""
"""
The above visual is not very clear, the numbers and text overlap in some instances. Let's make
    a few modifications to improve the visuals:

Remove the text labels on the pie chart by passing in legend and add it as a seperate legend
    using plt.legend().
Push out the percentages to sit just outside the pie chart by passing in pctdistance parameter.
Pass in a custom set of colors for continents by passing in colors parameter.
Explode the pie chart to emphasize the lowest three continents (Africa, North America, and
    Latin America and Carribbean) by pasing in explode parameter.
"""
colors_list = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue', 'lightgreen',
    'pink','black','antiquewhite','azure','cyan','darkred','deeppink']
explode_list = [0.1, 0, 0, 0, 0,0,0,0,0,0,0.1, 0.1] # ratio for each continent with which to
    offset each wedge.

df_continents['Total'].plot(kind='pie',
figsize=(15, 6),
autopct='%1.1f%%',
startangle=90,
shadow=True,
labels=None,      # turn off labels on pie chart
```

```
pctdistance=1.12, # the ratio between the center of each pie slice and the start of the text
    generated by autopct
colors=colors_list, # add custom colors
explode=explode_list # 'explode' lowest 3 continents
)

# scale the title up by 12% to match pctdistance
plt.title('Distribution of terrorist incidents by Region [1980 - 2013]', y=1.12)

plt.axis('equal')

# add legend
plt.legend(labels=df_continents.index, loc='upper left')
plt.show()
```