C 题：“智慧政务”中的文本挖掘应用

目录

背景

随着科技的发展，越来越多的网络问政平台成为了政府了解民生情况的重要渠道之一，但是由于科技发展迅速，以往的人工处理留言之类物已经变得十分困难，人手不足加上错误率高，导致了政府不能很好的了解实际情况，所以，大数据和人工智能就将成为代替人工的新潮流。

由于我们没有接触过此类代码，所以学起来十分困难，所有的东西都要从基础学起，我们选择观看了泰迪云课堂的赛前指导和指导进阶。经过讨论和观看，我们发现题目的内容只和部分视频的有关。比如：文本挖掘和可视化案例、案例：电商产品评论数据情感、自然语言处理技术等等，于是我们开始了解题。

问题一

对于问题一，首先我们组内进行了解题分析，发现我们要解决的是对建立可以对留言进行分类的模型。对于文本分类模型，我们是一点也不会，没有接触过，于是我们上网查找教程。我们发现我们需要学习许多种 python 库和没见过的 python 语言。首先我们在读取 Excel 文件选择了 pandas 库，这个库能很容易的提取 Excel 数据，并且还有很多其他有用的功能，在读取文件时，我们用到的代码是

df_data = pd.read_excel(r'C:\Users\Administrator\Desktop\泰迪杯\C 题全部数据\附件 2.xlsx',names=['留言编号','留言用户','留言主题','留言时间','留言详情','一级标签'],encoding='utf-8')

一开始的时候，python 报错了，我们发现是因为路径中有中文，于是我们加了 encoding='utf-8'，但他还是报错，我们发现是因为版本问题导致路径读取不成功，于是我们在路径前面加了 r，这样就可以成功了。然后我们了解到，在处理语言时需要先把句子分词，在分词之前又要先去除一些特殊的符号，但由于找不到有关特殊符号的文本，我们只能自己打出特殊符号，这可能会导致这个题的误差增大。接着为了处理文本，我们找到了 jieba 库，这个库能很好的根据我们中文的习惯进行分词，于是我们建立了 clearpara 函数，对留言详情进行分词，代码为

```
def clearpara(para):
    words = jieba.cut(para)
```

```
        words = [word for word in words if len(word)>1]
        words = [word for word in words if word not in stopwords]
        return words
```

接着我们对分词后的词组进行了词频统计，代码为：

```
items = df_data['words'].values.tolist()
words = []
for item in items:
    words.extend(item)
wordCount = pd.Series(words).value_counts()[0:topWordNum]
wordCount = wordCount.index.values.tolist()
```

然后我们在对语言相似情况用了词组转换为句向量的方法去实现文本相似。我们设了 wordsToVec 函数进行转换，其函数代码为：

```
def wordsToVec(words):
    vec = map(lambda word:words.count(word),wordCount)
    vec = list(vec)
    return vec
```

这就离我们实现建立模型又进了一步。但是，我们又陷入了难题，因为之前的代码概念相对来说比较好理解，我们可以参考其他案例来解决我们的问题，但接下来的步骤是建立模型，我们都非常不理解，很难知道其中的代码是代表了什么，于是我们开始了大量的百度搜索后我们终于总结了经验，然后在群里分享了自己的观点，大家互相学习，终于得出了一些似乎靠谱的东西，于是我们选择了构建贝叶斯模型，

代码为：

```
xTrain,xTest,yTrain,yTest = train_test_split(df_data['vec'].tolist(),df_data['一级标签'].tolist(),test_size=testRate,random_state=seed)
model = MultinomialNB()
model.fit(xTrain,yTrain)
yPred = model.predict(xTest)
yTest = np.array(yTest)
yPred = np.array(yPred)
```

最后得出的模型准确率为 0.72，虽然这个数字只能算一般，但是对于我们几个初学者来说，这是一件十分振奋人心的事情，毕竟我们做了出来。

第一问程序代码

```
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.externals import joblib
from sklearn.metrics import confusion_matrix
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import json
import pandas as pd
import jieba
import re
import os
topWordNum = 500
testRate = 0.2
seed = 0
baseDir = ''
staticDir = os.path.join(baseDir,'Static')
resultDir = os.path.join(baseDir,'Result')

stopwords=pd.read_csv(r"C:\Users\Administrator\Desktop\ 泰 迪 杯 \ 代 码 \stopwords.txt",engine='python',index_col=False,sep="\t",quoting=3,names=['stopword'], encoding='utf-8')
print(stopwords)

df_data = pd.read_excel(r'C:\Users\Administrator\Desktop\泰迪杯\C 题全部数据\附件2.xlsx',names=['留言编号','留言用户','留言主题','留言时间','留言详情','一级标签'],encoding='utf-8')
df_data = df_data.dropna()
content = df_data.留言主题.values.tolist()
df_data.head()

def clearpara(para):
    words = jieba.cut(para)
    words = [word for word in words if len(word)>1]
    words = [word for word in words if word not in stopwords]
    return words
print('正在分词，请耐心...')
df_data['words'] = df_data['留言详情'].apply(clearpara)
```

```
df_data.head()

items = df_data['words'].values.tolist()
words = []
for item in items:
    words.extend(item)
wordCount = pd.Series(words).value_counts()[0:topWordNum]
wordCount = wordCount.index.values.tolist()

def wordsToVec(words):
    vec = map(lambda word:words.count(word),wordCount)
    vec = list(vec)
    return vec
print('正在将词组转为句向量，请耐心...')
df_data['vec'] = df_data['words'].apply(wordsToVec)
df_data = df_data.drop(['留言详情'],axis=1)
df_data.head()

xTrain,xTest,yTrain,yTest = train_test_split(df_data['vec'].tolist(),df_data['一级标签'].tolist(),test_size=testRate,random_state=seed)
model = MultinomialNB()
model.fit(xTrain,yTrain)
yPred = model.predict(xTest)
yTest = np.array(yTest)
yPred = np.array(yPred)
print('模型测试集准确率:%.2f'%accuracy_score(yTest,yPred))

def predict(para):
    words = clearpara(para)
    vec = wordsToVec(words)
    vec = np.array(vec)[np.newaxis,:]
    flagIndex = model.predict(vec)[0]
    flag = flagIndex
    return flag
para = '| 留言：| 西地省绝色服饰有限公司位于 B2 区，品牌 j.s.。作为有几十家直营店，100 多号员工，所有员工没有一人有社保。我想这应该不合理也不合法，盼望领导支持。'
print(para)
print('\n| 留言分类：| %s'%predict(para))
```

第一问运行过程及函数测试截图

导入并输出部分停用词：

```
stopwords=pd.read_csv(r"C:\Users\Administrator\Desktop\泰迪杯\代码\stopwords.txt",engine='python',index_col=False,sep="\t",quoting=3,names=[
stopwords.head()
```

	stopword
0	!
1	"
2	#
3	$
4	%

读取 Excel 文件，将留言主题赋值给 content 转换为列表：

```
df_data = pd.read_excel(r'C:\Users\Administrator\Desktop\泰迪杯\C题全部数据\附件2.xlsx',names=['留言编号','留言用户','留言主题','留言时间','
df_data = df_data.dropna()
content = df_data.留言主题.values.tolist()
df_data.head()
```

	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签
0	37	U0008473	A市在水一方大厦人为烂尾多年，安全隐患严重	2020/1/4 11:17:46	\n\t\t\t\t\t\n\t\t\t\t\t位于书院路主干道的在水一方大厦一楼至四楼人为...	城乡建设
1	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:14	\n\t\t\t\t\t\n\t\t\t\t\t尊敬的领导：A1区苑小区位于A1区火炬路，小...	城乡建设
2	303	U0007137	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	2019/12/6 14:40:14	\n\t\t\t\t\t\n\t\t\t\t\tA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设
3	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	\n\t\t\t\t\t\n\t\t\t\t\tA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设
4	379	A00016773	投诉A市盛世耀凯小区物业无故停水	2019/11/28 9:08:38	\n\t\t\t\t\t\n\t\t\t\t\t我在2015年购买了盛世耀凯小区17栋3楼，...	城乡建设

进行分词：

```
def clearpara(para):
    words = jieba.cut(para)
    words = [word for word in words if len(word)>1]
    words = [word for word in words if word not in stopwords]
    return words
print('正在分词，请耐心...')
df_data['words'] = df_data['留言详情'].apply(clearpara)
df_data.head()
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ADMINI~1\AppData\Local\Temp\jieba.cache
```

正在分词，请耐心...

```
Loading model cost 1.097 seconds.
Prefix dict has been built successfully.
```

	留言编号	留言用户	留言主题	留言时间	留言详情	一级标签	words
0	37	U0008473	A市在水一方大厦人为烂尾多年，安全隐患严重	2020/1/4 11:17:46	\n\t\t\t\t\t\n\t\t\t\t\t位于书院路主干道的在水一方大厦一楼至四楼人为...	城乡建设	[位于, 书院, 主干道, 在水一方, 大厦, 一楼, 四楼, 人为, 拆除, 电等, 设施...
1	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:14	\n\t\t\t\t\t\n\t\t\t\t\t尊敬的领导：A1区苑小区位于A1区火炬路，小...	城乡建设	[尊敬, 领导, A1, 区苑, 小区, 位于, A1, 火炬, 小区, 物业, 市程明, ...
2	303	U0007137	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	2019/12/6 14:40:14	\n\t\t\t\t\t\n\t\t\t\t\tA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设	[A1, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不洗, ...
3	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	\n\t\t\t\t\t\n\t\t\t\t\tA1区A2区华庭小区高层为二次供水，楼顶水箱...	城乡建设	[A1, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不洗, ...
4	379	A00016773	投诉A市盛世耀凯小区物业无故停水	2019/11/28 9:08:38	\n\t\t\t\t\t\n\t\t\t\t\t我在2015年购买了盛世耀凯小区17栋3楼，...	城乡建设	[2015, 购买, 盛世, 耀凯, 小区, 17, 两层, 共计, 平方, 一直, 以来,...

词组转换为局向量：

```
items = df_data['words'].values.tolist()
words = []
for item in items:
    words.extend(item)
wordCount = pd.Series(words).value_counts()[0:topWordNum]
wordCount = wordCount.index.values.tolist()
```

```
def wordsToVec(words):
    vec = map(lambda word:words.count(word),wordCount)
    vec = list(vec)
    return vec
print('正在将词组转为句向量，请耐心...')
df_data['vec'] = df_data['words'].apply(wordsToVec)
df_data = df_data.drop(['留言详情'],axis=1)
df_data.head()
```

正在将词组转为句向量，请耐心...

	留言编号	留言用户	留言主题	留言时间	一级标签	words	vec
0	37	U0008473	A市在水一方大厦人为烂尾多年，安全隐患严重	2020/1/4 11:17:46	城乡建设	[位于, 书院, 主干道, 在水一方, 大厦, 一楼, 四楼, 人为, 拆除, 电等, 设施...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...
1	83	A00063999	投诉A市A1区苑物业违规收停车费	2019/12/30 17:06:14	城乡建设	[尊敬, 领导, A1, 区苑, 小区, 位于, A1, 火炬, 小区, 物业, 市程明, ...	[0, 1, 2, 0, 3, 0, 0, 0, 0, 0, 0, 0, 7, 0, 3, ...
2	303	U0007137	A1区蔡锷南路A2区华庭楼顶水箱长年不洗	2019/12/6 14:40:14	城乡建设	[A1, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不洗, ...	[3, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, ...
3	319	U0007137	A1区A2区华庭自来水好大一股霉味	2019/12/5 11:17:22	城乡建设	[A1, A2, 区华庭, 小区, 高层, 二次, 供水, 楼顶, 水箱, 长年, 不洗, ...	[3, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, ...
4	379	A00016773	投诉A市盛世耀凯小区物业无故停水	2019/11/28 9:08:38	城乡建设	[2015, 购买, 盛世, 耀凯, 小区, 17, 两层, 共计, 平方, 一直, 以来,...	[3, 2, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, ...

计算准确率：

```
xTrain,xTest,yTrain,yTest = train_test_split(df_data['vec'].tolist(),df_data['一级标签'].tolist(),test_size=testRate,random_state=seed)
model = MultinomialNB()
model.fit(xTrain,yTrain)
yPred = model.predict(xTest)
yTest = np.array(yTest)
yPred = np.array(yPred)
print('模型测试集准确率:%.2f'%accuracy_score(yTest,yPred))
```

模型测试集准确率:0.72

建立函数测试模型：

```
def predict(para):
    words = clearpara(para)
    vec = wordsToVec(words)
    vec = np.array(vec)[np.newaxis,:]
    flagIndex = model.predict(vec)[0]
    flag = flagIndex
    return flag
para = '| 留言：| 西地省绝色服饰有限公司位于B2区，品牌j.s.。作为有几十家直营店，100多号员工，所有员工没有一人有社保。我想这应该不合理也不合法
print(para)
print('\n| 留言分类：| %s'%predict(para))
```

| 留言：| 西地省绝色服饰有限公司位于B2区，品牌j.s.。作为有几十家直营店，100多号员工，所有员工没有一人有社保。我想这应该不合理也不合法，盼望领导支持。

| 留言分类：| 劳动和社会保障

问题二

第二题我们找了特别久，也学了很多的知识，但可惜最后我们没有能够做出来。下面我讲一下我们的思路，首先我们在网上找不到类似的案例，然后正当我们一头雾水时，其中一位组员偶然发现了有一个知识点叫做文本聚类，他的内容和我们第二题的方向大致相同，于是我们开始往这个方向学习。我们发现，在文本聚类的时候，前面的一些步骤比如读取文件，将文件内容转换为 list，然后进行 jieba 分词，去除特殊符号和文字等等，这些都是类似于第一问的步骤，我们看了之后以为很快就可以做完了，可是后来发现难的步骤在后面。在分词的同时，它需要储存分词后的词语的词性，这一步让我们十分头疼，就是从这一步开始我们开始弄不了了，找了很多资料却不能理解，十分可惜。紧接着的步骤是短文本特征提取，这个步骤是建立在储存词性上的，所以我们也没有实现，最后的步骤就是 DBSCAN 聚类分析，接着简单对题目要求进行修改。