利用组合模型进行用户反馈意见的自动归类

B2C 开发部 ， 潘广宇 T08726

一、需求背景

同花顺系列软件每天都会收到许许多多用户反馈的意见，用户的意见很大程度决定着用户对产品的满意度。在现有的同花顺运营后台中，对用户反馈意见的处理主要是后台管理系统通过人工进行审核处理，而并没有从用户的反馈意见中挖掘出用户对本公司产品（或本期产品）的评价、满意度。而获取用户的评价与满意度对将来的产品研发和推广是至为重要的。因此，此次功能创新的方案旨在通过机器学习相关算法的组合模型，对用户的评价内容进行分类，从而提取出十分有价值的数据。

二、方案设计

通过对用户留言的后台管理系统进行大致查阅后，目前可将用户评论内容大致分为：**满意（3分）、中肯（2分）、仍需努力（1分）、差评（0分）**四种情况。

接下来，我们对用户的留言进行分类：

步骤1：获取从历史至今的用户反馈意见数据，从现有后台的数据库中可导出**18**年**6**月**21**日至今的数据，约**1800**条，作为训练数据集。

addtime	isvalid	userip	content	uid
2018-05-08 17:18:04	1	172.17.12.111	版本号：8.70.50　意见：我爱小胡子	315078002
2018-05-30 13:21:45	1	119.48.201.172	版本号：8.70.50　意见：无法同步指标，以前的版本，指标增加或者减少后能正常增减指标并且显示出来。现在的版本指标删除	317279484
2018-05-30 14:52:18	1	111.196.76.159	版本号：8.70.50　意见：多股分时同列保存不了板式, 每个我选的满占坐标, 退回到其它界面, 比如板块, 再回到多股同列, 满占坐	393041348
2018-06-01 10:51:22	1	171.94.254.100	版本号：8.70.50　意见：行业板块都不显示	411127384
2018-06-01 13:24:46	1	115.231.11.13	版本号：8.70.50　意见：在我的板块里面不能自动选择需要的指数了，只出现股票，和问财不同步	401691047
2018-06-01 21:59:43	1	60.222.106.1	版本号：8.70.50　意见：没找见	366323660
2018-06-02 12:49:07	1	60.218.126.102	版本号：8.70.50　意见：为什么以前的指标不能用了，不如从前的。	170717790
2018-06-02 15:56:31	1	119.166.217.32	版本号：8.70.50　意见：一般	207930120
2018-06-05 09:15:27	1	183.54.205.88	要是能有.....功能就更好了	386690985
2018-06-05 22:45:00	1	36.22.52.224	版本号：8.70.50　意见：新版在委托银河证券委托登录方面与其它券商界面不一样，每次都要输入密码！	305320737
2018-06-05 23:10:56	1	39.130.124.78	版本号：8.70.50　意见：用用看	4023664
2018-06-05 23:35:07	1	120.239.172.187	版本号：8.70.50　意见：很好用	172141177
2018-06-06 14:45:56	1	123.233.53.146	版本号：8.70.50　意见：使用高清屏太虚了，软件需要能自动适应屏幕的分辨率。	35640299
2018-06-06 15:01:05	1	222.90.86.35	版本号：8.70.50　意见：容易崩溃	24428218
2018-06-06 16:07:26	1	218.63.186.18	版本号：8.70.50　意见：同花顺最近几个月太卡了，用鼠标点击指标时跳出其它指标，请贵公司软件开发部解决这一问题，每	347449955
2018-06-06 20:35:02	1	103.254.71.226	版本号：8.70.50　意见：自定义板块手机、PC端同步丢失或出错	215301113

步骤**2**：对训练集进行数据清洗，主要是过滤掉：重复的评论内容、单字、内容、大量无意义的符号数字、空白行、无意义的文字等

步骤 **3**：标记训练集，对 **1800** 余条的评论内容进行分类标记（详见附件 **feedback.xlsx**），主要标记方式是：对每条用户的评论内容进行分值的标记，（满意**-3** 分，中肯**-2** 分，仍需改进**-1** 分，差评**-0** 分）

	A	
1	classification	comment
2	3	做的软件越来越好看了，支持一下！
3	3	做的更好
4	2	昨晚才装，未及使用，熟悉后再说。
5	2	昨天刚升级，还要适应呢
6	1	最右侧的主营增长率、净利润增长率等数字能否用红色或绿色等显眼和颜色，灰色看起来不舒服
7	1	最新版的软件非常卡，怎么回事？还有网络经常不稳定常挂线
8	3	最喜欢的炒股软件，非常棒。
9	3	最满意的炒股软件，大拇指。赞一个。虽然还是亏本状态。呵呵呵
10	1	最近总是时不时的没办法缩小和放大k线图，感觉顿顿的
11	1	最近一直不能下载历史数据啊。
12	1	最近无法同步预警！
13	1	最近为什么不能正常使用？有损上市公司的形象！
14	1	最近同花顺软件用的非常非常卡顿，很不顺心，我以为是我电脑的原因，我换了几台电脑都是卡顿

步骤 **3**：对用户的意见进行分词。分词的目的是为了计算词频，词频是作为衡量文本特征最主要的方式。利用 **jieba** 分词可快速获取到一个文本分词后的结果。如：“最满意的炒股软件”将得到分词结果：“最 满意 的 炒股 软件”

步骤 **4**：选取合适的算法训练模型。本次算法模型考虑采用 **TFIDF+朴素贝叶斯法+snowNLP 组合模型**进行分类。

原因：

1、**TFIDF** 算法主要用于提取意见文本的特征值，该算法可对文本的频率进行统计，评论分类问题本质上是文本分类问题，对文本分类，最好的方式是通过计算文本之间的特征值，根据特征值近似度进行归类。如：“最喜欢的炒股软件”与“最满意的炒股软件”在特征上非常相近，可近似为一个类别。

2、朴素贝叶斯法用于训练特征值：训练 **TFIDF** 的矩阵（作为意见的特征值），得到后验概率最大的分类。朴素贝叶斯法实现简单并且学习与预测的效率都很高，因此通过该模型找出意见的特征值与分类之间的联系。

3、**snowNLP** 是用于情感分析计算：主要是权衡一下由上述两个步骤得出的分类的误差。因为用户的意见本身是带有情感色彩的，可根据情感分析得到的积极与消极的分值对上述算法得到的分类值进行比较，最终得到最后的分类结果。

步骤 **5**：由于现有的训练数据集较少，无法通过大量的测试集进行分类的测试，目前只通过随机抽样的方式进行测试，估算大致的准确率

步骤 **6**：对训练好的模型进行保存，并可根据实时计算或离线计算的方式，传入用户的反馈意见并得到分类结果进行保存。分析部门可根据该分类的结果进行统计与分析，得到用户总体的评价分布。

三、算法过程

步骤 **1**：导入训练数据集，并进行 **jieba** 分词，得到所有训练集分词后的数组，如 **[**“最 满意 的 炒股 软件”，“最 喜欢 的 炒股 软件”，**...]**

```
# 导入训练数据
data = pd.read_excel('feedback.xlsx')

data = np.array(data)

# 获取分类集和数据集
classification = np.array(data[:, 0], dtype='int')
comment = np.array(data[:, 1], dtype='object')

# 对数据集（文本）进行分词
commentStack = [];
for number in range(len(comment)):
    commentWord = jieba.cut(str(comment[number]))
    commentWord = " ".join(commentWord)
    commentStack.append(commentWord)
```

步骤 **2**：利用 **sklearn** 计算每个词在每个意见文本下的词频

```
vectorizer = CountVectorizer()    # 计算文本的词频矩阵
arr = vectorizer.fit_transform(commentStack)  # 矩阵元素a[i][j] 表示j词在i个文本下的词频
```

步骤 **3**：通过计算好的词频向量传入 **sklearn** 的 **TfidfTransformer** 计算每个意见文本对应的 **tfidf** 数组。

```
transformer = TfidfTransformer()
tfidf = transformer.fit_transform(arr) # 传入词频向量计算tf-idf
```

得到一个意见文本下，每个词语的 **tfidf** 数值：

```
几台 0.169030850946
卡顿 0.338061701891
原因 0.169030850946
同样 0.169030850946
同花顺 0.338061701891
我换 0.169030850946
我用 0.169030850946
是不是 0.169030850946
最近 0.169030850946
机子 0.169030850946
电脑 0.338061701891
真有 0.169030850946
知道 0.169030850946
财富 0.169030850946
软件 0.169030850946
```

步骤 **4**：利用*先验为多项式分布的*朴素贝叶斯法对 **tfidf** 进行训练，并把每个意见文本对应的每个词语的 **tfidf** 值作为特征，记为 **comment 1 :** 特征向量：**(x1,x2,x3,...)**，分类：**y1 ;** 得到的如下：

序列号	意见评论内容	tfidf 特征值	分类
1	最满意的炒股软件	[“最”:xxx, “满意”:xxx , … “软件”:xxx]	3
2	最喜欢的炒股软件	[“最”:xxx, “喜欢”:xxx , … “软件”:xxx]	3
…			

把组合好的二维数组（序号 **1**、**2**、**3...**的特征值加起来）作为训练的特征值，分类结果作为分类值，进行模型训练

```
In  [61]: # 朴素贝叶斯统计

          mnb = MultinomialNB()
          mnb.fit(tfidf, classification)

Out[61]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

步骤 **5**：将 **feedback.xlsx** 对应的训练数据进行排序，筛选出标记为 **3**：满意与标记为 **0**：差评的意见评论内容，分别存放在 **feedback_positive.txt** 与

feedback_negative.txt 中，分别作为积极的文本训练集、消极的文本训练集，对 **snowNLP** 模型进行训练：

```
sentiment.train('feedback_negative.txt', 'feedback_positive.txt')
sentiment.save('sentiment.marshal')
```

步骤 **5** 训练完成后，可得到一个意见对应的积极**/**消极的概率值。

步骤 **6**：对传入的意见内容进行预测：计算意见内容的每个词语 **tfidf** 数值作为该意见内容的特征值，利用上述训练好的朴素贝叶斯模型对该特征值进行预测

```
test_comment = u'最满意的炒股软件'

test_comment = jieba.cut(test_comment)
test_comment = " ".join(test_comment)

test_comment = [test_comment]

test_vectorizer_wordFrequency = vectorizer.transform(test_comment)

test_tfidf = transformer.transform(test_vectorizer_wordFrequency).toarray()

predict_classification = mnb.predict(test_tfidf)
predict_classification_proba = mnb.predict_proba(test_tfidf)
```

得到 **predict_classification** 即是朴素贝叶斯法得到的分类结果，**predict_classification_proba** 是预测每个分类的概率，如：预测为 **0**（差评）的概率是 **A**，预测为 **1** 的概率是 **B**，**...**预测为 **3**（满意）的概率是 **C**

然后，传入该意见内容到 **snowNLP** 模型进行积极**/**消极的概率计算：

```
snownlp = SnowNLP(str(test_comment))
snow_score = snownlp.sentiments

print(snow_score)
```

训练 **snowNLP** 模型后我们得到 **0-1** 的一个概率值**(snow_score)**，越接近 **1** 代表该意见越趋向于积极，越接近 **0** 代表该意见越趋向于消极。

至此，我们得到了一个朴素贝叶斯法计算的分类概率与情感分析积极与消极的概率。

为确保预测的准确率，因为在样本集相对较少的情况下，其中一种模型可能会对某些内容预测误差大，因此使用两种模型选最优的方式选择最终的分类结果。

比较 **snow_score**（积极/消极的概率）与朴素贝叶斯法下每个分类概率的大小（naive_score 表示的是朴素贝叶斯估计中 4 个分类中的最大概率）：

	Snow_score > 0.5					snow_score < 0.5 （snow_score = 1-snow_score）		
比较两个模型的概率	snow_score > naive_score				snow_score < naive_score	snow_score > naive_score		snow_score < naive_score
根据概率的大小	< =0.25	> 0.25	> 0.5	>0.75	naive_score	> 0.5	> 0.75	naive_score
分类结果	0	1	2	3	取朴素贝叶斯模型的分类结果	1	0	取朴素贝叶斯模型的分类结果

```
naive_score = predict_classification_proba[0][predict_classification[0]]     # 朴素贝叶斯得到的最大分类概率

if (snow_score >= 0.5):
    if (snow_score >= naive_score):      # 如果情感分析得分大于朴素贝叶斯最大概率，则选择该标签
        final_score = snow_score
        if (snow_score > 0.25):
            if (snow_score > 0.5):
                if (snow_score > 0.75):
                    tag = 3              # 标记为3分满意
                else:
                    tag = 2              # 标记为中肯
            else:
                tag = 1
        else:
            tag = 0
    else:
        final_score = naive_score
        tag = predict_classification[0]
else:
    snow_score = 1-snow_score
    if (snow_score >= naive_score):      # 如果情感分析得分大于朴素贝叶斯最大概率，则选择该标签
        final_score = snow_score
        if (snow_score > 0.5):
            if (snow_score > 0.75):
                tag = 0
            else:
                tag = 1
    else:
        final_score = naive_score
        tag = predict_classification[0]
```

举个栗子：

```
[[ 0.05073435  0.32995416  0.04215937  0.57715211]]
0.9808587058313183
3
```

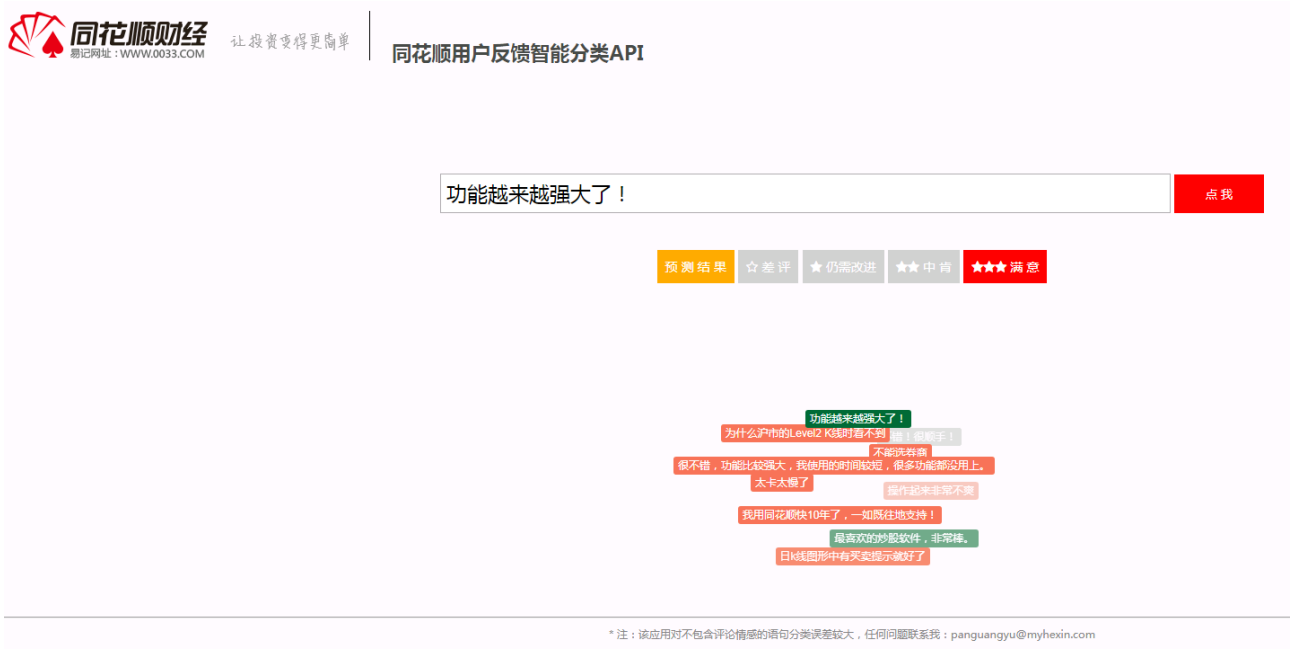
最上面的概率为分类从差评 ~ 满意 4 个分类的概率值，中间的概率为情感为积极的概率，最后的数字代表最后的分类结果为 3：满意

先判断中间的情感概率大于 0.5 时，发现都大于朴素贝叶斯概率且都大于 0.75，则分类结果为 3（满意）

四、可行性分析

由于当前可使用的意见数据稀少（至今仅存有 1800 条数据），不足以单独作为测试集使用，因此，为检验该模型的准确率，通过随机抽样 100 条评论内容进行预测，比较预测的结果，得出大致的准确率

API 地址： http://139.199.178.177:8000/index



下表列出预测分类的效果：

序号	意见内容	正确的分类	预测分类	是否预测正确
1	首页下方的大 bug	仍需改进	仍需改进	1
2	网络非常卡，到底能不能解决下，	仍需改进	仍需改进	1
3	十分好，是我很喜欢的软件	满意	满意	1
4	一般般，有待提高	中肯或仍需改进	仍需改进	1
5	希望同花顺越做越好	满意	满意	1
...				

经过大量的测试与检验：

1、本模型对不包含情感的评论内容，如“同花顺”，预测的误差较大

2、由于训练样本不足，部分语句可能预测出现误差

3、在 1800 余条训练样本中，本模型的效果达到预期目标；当训练样本增加到一定的数据集时，本模型的预测结果将会越来越高，随着训练样本数的增加而增加，认为该方案可在大量训练样本的情况下，对意见内容进行归类。

五、商业价值

1、减少人力劳动，当前对用户意见的处理是通过人工审核的方式，当该方案可实施时，可大量减少人工审核的时间。并且当用户的评论意见开始成倍增加时，使用机器进行自动归类将极为重要。

2、提高销售与个性化广告的业务，对用户的意见反馈进行归类后，销售与广告部门可有效筛选出认为满意的用户，并对这部分用户增加广告的投放，增加用户购买产品的可能性。对差评的用户尽量少投放广告，通过其他渠道改善该部分用户的用户体验。

3、预测公司业绩，同花顺很大一部分收入来源于收费业务，可根据收费的客户端软件的意见评价分类的分布大致估计用户对该产品的满意度占比，当满意度较高时，可认为下季度用户续费的可能性较大，从而判断出那部分用户可能存在续费的可能性。

4、分析新版本的用户体验，有助于改善下个版本。可根据每次版本发行的时间进行统计，分别统计出不同版本下用户的满意度占比，当出现某个版本满意度较高，说明某个功能的升级让用户较满意，可在下个版本中继续升级改进。若出现某个版本差评率较高，说明本次升级让用户体验降低，可改进新版本增加的功能。从而在以后的版本迭代中逐步提高用户的体验。

5、挖掘产品的功能。根据差评率和仍需努力的分类，找到同花顺系列软件中可提高升级的地方，或者用户的建议可供参考之处。

注：本方案使用的 TFIDF+朴素贝叶斯法+snowNLP 组合模型是本人自己构思的创新性组合方案，包括如何对比两个模型的概率值，获取最终的分类结果等均是个人创新方案，并无抄袭。