hw2 实验报告文档

姓名：孙浩然

学号：1652714

特征提取：

本次作业中，老师不仅给出了原始的数据，更重要的是，还给出了很多从原始数据中提取出的特征，这让我们的工作简单了很多。但是给出的这些特征很多和我一开始的想法还不太完全一样，所以我还是对老师给出的原始数据和特征分别做了进一步的处理。总共形成了三个特征向量，其中共包括了四个特征：

1. 评论文本的 word embeding (200维向量)
2. 用户的 emoji、语气助词和拟声词使用平均数量和频率
3. 异常用户数据
4. 评论文本的前十常用词频率统计

下面，我将分别介绍我对以上特征的处理操作。

已有特征处理：

特征一：

在上方的四个特征中，第一个特征是将每个用户的所有评论进行分词，然后将每一个词依据上下文转化为一个200维的词向量，然后将取平均值，这样每个用户就拥有了一个200维的特征向量。我认为这个特征向量已经可以很好地表示每个用户的特征，所以我没有对这个特征数据进行大的变动，只是将评论信息中的不必要的数据删除，只留下用户ID和200维坐标，期间用逗号分隔，存放在：`./data/id_vec.csv` 文件中。这个处理过程可以在脚本文件 `./parse_data.py` 中看到。

特征二：

用户的 emoji、语气助词和拟声词使用情况，已经在 `./data/dianping_figuration.csv` 文件中给出了，但是这个数据并不能直接使用，打开文件就可以看出，数据的长度都是不相同的。所以我对这个数据进行了进一步的处理，将每个用户的数据统计成：

1. 用户发表评论数量
2. 用户平均每条评论出现 emoji、语气助词和拟声词的数量

这样，元数据中的特征的向量就转化为了一个二维的特征向量。这个处理过程可以在脚本文件 `./pre_prossesor.py` 中看到。

特征三：

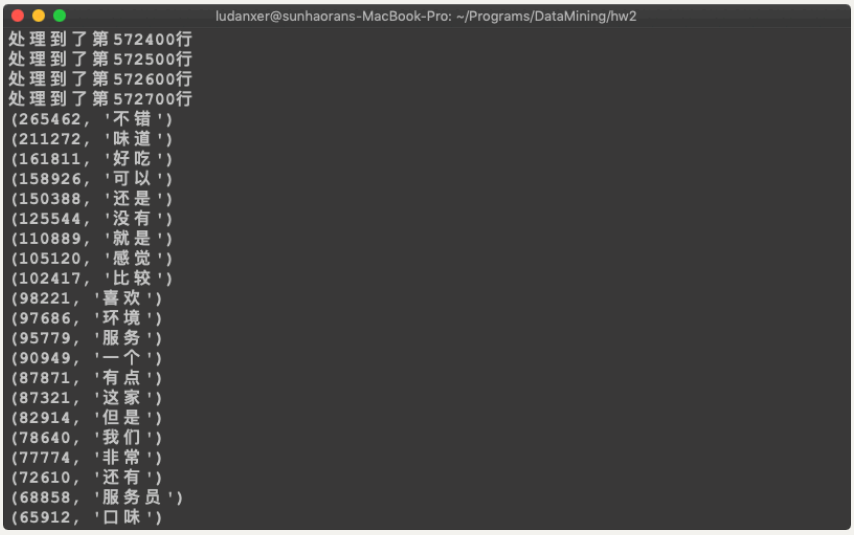
这个数据我没有做任何处理，因为本身已经很简练了，我为了将这个数据加进来，一个是我主观的认为这个特征的确有助于判断用户到底是男是女，一个是我希望能通过以下的实验进一步确认我的判断。

新特征获取：

特征四：

评论文本的前十常用词频率统计是我自己从原始数据中(`./data/dianping_review.csv`)提取的一个特征，因为这个方法我在上一次的作业中也用过，感觉效果还是不错，所以也在这次作业中用作特征提取。这个特征的提取步骤如下：

1. 首先遍历所有的评论信息，使用结巴分词对每条评论信息进行分词，将其中长度大于一的词汇全部删去
2. 将所有词汇集中起来，统计每个词汇使用的频次，输出前五十使用次数最多的词汇，选取其中使用频次最高的前十个词汇：`'不错', '味道', '好吃', '可以', '还是', '没有', '就是', '感觉', '比较', '喜欢'`



3. 将每个用户使用的使用的全部词汇集中起来，统计每个用户的使用前十个词汇使用的频次，生成一个十维特征向量：[`不错_出现次数`,`味道_出现次数`,...]

我认为这个新的特征和老师给出的200维特征向量有一点重复，这个向量所包含的数据虽然不如200维向量全面，但是我认为这个特征比200维向量更具有特点，因为200维向量是对用户评论中的每一个词都求取了词向量，这样很多男女都会使用的不具有任何特点的一些词汇（比如：`的`）也会被包含在特征向量中，而十维向量就不会有这方面的问题。这个特征的处理过程可以在脚本文件`./get_new_feature.py`中看到。

当然，真正的结果也不能全靠我的猜想，真实的情况还是要看我在下面的测试结果。

数据预处理：

筛选数据：

在上文特征提取的同时，我一并对数据进行了一次的数据筛选，整体数据中不仅有很多不是特征的数据，还有很多没有用的数据，最终经过数据筛选和特征的提取的数据结果如下：

```
├── data
|   ├── id_gender.csv
|   ├── id_malicious_fnum_fave.csv
|   ├── id_review_vec.csv
|   ├── id_vec.csv
|   ├── ......
```

每一个以`id_`开头的`csv`文件都对应上文特征提取中的一个特征，特征提取的过程我不再赘述，这里只是说一下我筛选出去的一下无用的数据：

1. 首先，我读取了`dianping_gender.csv`文件中的所有的用户ID。因为只有这些用户才有label，没有 label 的用户对我们是完全没有用途的
2. 在特征提取的过程中，所有没有出现在`dianping_gender.csv`文件中的用户ID及其对应的数据一律删除，这些数据虽然可以提取出特征，但是没有 label 对应，也是没有用途的
3. 所有的数据只留下**用户ID**和**特征向量**，剩下的如`comment_id`等不必要的数据全部删去，对我们接下来的数据预测并没有任何帮助

训练集、验证集和测试集的划分：

在本测试过程中，我们将数据分为三部分：训练集、验证集和测试集。比例依次为：8：1：1。其中，我们依靠训练集对分类模型进行训练，接下来使用验证集对初步训练好的分类模型进行验证，最后使用测试集对分类模型的准确性进行检测。

分类的初期实验结果：

首先，我直接使用四种分类算法对三种特征数据进行训练以及预测，得到的结果依次如下：

id_vec 特征向量：

算法\结果评估	ACCURACY	PRECISION	RECALL	F1-SCORE
Logistic Regression	82.1%	94.0%	84.0%	87.0%
SVM	74.1%	100%	74.1%	85.1%
Random Forest Classifier	81.0%	95.0%	82.0%	88.0%
Neural Network(ReLu)	80.8%	91.1%	84.2%	87.5%

可以明显看出，利用本特征向量训练所得的结果最好，最为明显。说明200维向量还是最能体现用户特征的一个特征。

id_review_vec 特征向量：

算法\结果评估	ACCURACY	PRECISION	RECALL	F1-SCORE
Logistic Regression	74.7%	99.2%	74.8%	85.3%
SVM	75.6%	98.1%	76.0%	85.6%
Random Forest Classifier	75.9%	92.1%	78.9%	85.0%
Neural Network(ReLu)	73.8%	87.3%	79.4%	83.1%

虽然这个特征向量所得的结果不如 `id_vec` 特征向量所得的结果好，但是效果相比之下也不是很差，差了不到10个百分点。同时，我们还需要注意是，这个特征向量的获得的过程可是比200维特征向量简单多了，所花的时间也少多了，所以相应的准确度差一点也是可以理解的。

id_mff 特征向量：

算法\结果评估	ACCURACY	PRECISION	RECALL	F1-SCORE
Logistic Regression	74.1%	100%	74.1%	85.1%
SVM	75.1%	98.8%	75.3%	85.4%
Random Forest Classifier	70.8%	89.3%	75.7%	81.9%
Neural Network(ReLu)	74.5%	99.5%	74.6%	85.3%

这个特征向量所获得的训练结果是几个结果中最差的一个，同时，这个特征向量的维度也是最少的，只有3维，所能体现的特征自然也就相对较少，效果自然也就相对较差。在下文提高准确率的尝试中，我曾尝试组合这几种特征向量，看所得出的结果是否能有提高。

提高准确率的尝试：

参数调整：

在测试过程中，我对几个分类器函数中的参数进行了多次的调整，将得到的结果简单阐述如下：

Logistic Regression 与 Neural Network(ReLu):

在这两个函数中，我主要是调整了 `max_iter` 参数，这个参数的主要意义是迭代的次数，我对这个参数比较关心的原因是一开始我让它为默认值，之后函数报了warning，提示我这个函数到了最大迭代次数却还没有结束。我也不敢将这个参数直接调整的太大，因为害怕迭代次数过大了会产生过拟合，所以不断地进行尝试，最后发现即便将这个参数设置的很大很大，函数的预测结果也没有发生太大的变化，最多增长了2%而已，也就意味着没有出现过拟合的现象，所以我最后放心的将这个参数设置的很大了。

Random Forest Classfier:

在这个函数中，我调整了 `max_depth` 参数的值，我以为，这个参数的值越大，所花的时间越长，最后取得的效果越好。在实际的测试过程中，我意识到，即使这个参数增大很多，所用的时间却也不会发生很大的变化。当 `max_depth` 达到20时，这个函数的预测结果就不再变化了。

组合特征：

我尝试将三个特征向量组合起来，形成新的特征向量来对数据进行分析，希望能够得到更好的预测结果。其中，由于 `id_vec` 与 `id_review_vec` 都是对用户评论数据特征的反应，所以并没有将这两个数据连接起来。剩下的两种组合方式如下：

id_vec + id_mff 特征向量：

算法\结果评估	ACCURACY	PRECISION	RECALL	F1-SCORE
Logistic Regression	81.8%	93.2%	84.0%	88.4%
SVM	74.1%	100%	74.1%	85.2%
Random Forest Classifier	81.0%	95.0%	82.0%	88.0%
Neural Network(ReLu)	81.8%	90.5%	85.8%	88.0%

将 `id_vec` 和 `id_mff` 两个向量连接起来，形成一个203维的新的特征向量，再使用这个新向量进行训练后，所得结果如上。可以看出，虽然在 `id_vec` 上面新增加了3维数据向量，理论上说，预测的结果应该比之前更好了，但是实际结果却还不如之前的200维向量的好了。

我认为这样的结果是因为 `id_mff` 特征向量并不能充分反映用户的特征，导致拉低了 `id_vec` 特征向量的预测结果。说明我的 `id_mff` 特征向量选取的并不够好。

id_review_vec + id_mff 特征向量：

算法\结果评估	ACCURACY	PRECISION	RECALL	F1-SCORE
Logistic Regression	74.3%	98.8%	74.7%	85.1%
SVM	74.9%	95.7%	75.7%	85.2%
Random Forest Classifier	75.9%	92.4%	78.8%	85.1%
Neural Network(ReLu)	75.7%	90.2%	79.7%	84.6%

本次测试将 `id_review_vec` 与 `id_mff` 两个特征向量连接了起来，形成了一个13维的向量，使用这个特征向量预测的结果见上表。可以看出，这个新特征向量的结果好于 `id_mff` 特征向量的结果，却差于 `id_review_vec` 特征向量的结果，再结合上一小结的结果，可以看出，`id_mff` 特征向量的确不能很好地反应用户的特征，或者说不如其他两个特征向量反应的好。

组合分类器：

在这部分的测试过程中，我将实验效果最好的两种分类器（`Logistic Regeression` 和 `Neutal Network(ReLu)`）结合起来，想看看最后的预测结果会不会有提升。具体的结合方法是：

1. 得出两种分类器对每条数据的预测概率
2. 对两种概率乘以一定比例并相加
3. 选取概率较大的一个作为最终的预测结果

在这项测试中，我选用之前效果最好的200维向量最为测试数据，所得到的结果如下：

算法\结果评估	ACCURACY	PRECISION	RECALL	F1-SCORE
综合算法(6:4)	83.0%	85.6%	92.7%	88.9%
综合算法(5:5)	82.3%	83.8%	94.2%	88.7%
综合算法(6.5:3.5)	83.3%	85.3%	93.6%	89.2%

通过结果可以看出，当使用比例65%:35%结合两种分类器时，可以达到最好的效果：准确率83.3%。这也就是我的最后的结果，该结果存放在 `./data/dianping.csv` 文件中。