中山大學
SUN YAT-SEN UNIVERSITY

Data Mining Summer Course

Project: Predict Stress in English Words

学 院 名 称 ： 数据科学与计算机学院

学 生 姓 名 ： 饶宇熹 15331262
苏惠玲 15331281
明友芬 15331242

时 间 ： 2017 年 8 月 7 日

一、 实验内容

In this project, you need to build a classifier to predict the stresses for a list of English words. Output the position of the primary stress.

二、 实验要求

- train()

In order to successfully predict the stress, you need to train a classifier. You are required to implement a function named train(). Its two arguments are the training data (stored as a list of strings) and the output file path.
You need to dump the classifier and relevant data/tools (if there is any) into one single file.

- test()

You also need to implement a function named test(), which takes the test data as input and returns a list of integers which indicate the positions of the primary stress.

- Restrictions
  - The **total** running time of training and testing **should not exceed 10 minutes** in the submission system. The system will force stop your program if it took more than 10 minutes, and you will receive 0 point for the programming part.
  - You are encouraged to use **any** classifiers from sklearn, but you **can not use** any other machine learning package.

三、 环境

- python: Python 3.6.1 |Anaconda 4.4.0 (64-bit)
- pandas: 0.19.2
- numpy: 1.12.1
- scikit-learn: 0.18.1

四、 实验过程

（一）确定 feature

➢ Model1

Features：单词的元音数

思路：

我们要预测单词在哪个地方重读，其实可以简化为**分类问题**和**预测问题**。分类问题即将单词按第 i 个音节重读分类。预测问题则在给定发音的基础上，求哪

个音节重读的概率最高。

对于单词本身而言，做分类和预测最基础的性质便是**单词的元音数(音节数)**。音节数为 1 的单词无需预测其重音；而音节数为 2、3、4、5 的单词则需要通过训练，寻找音节数与重音位置的联系，算出某音节数下与某音节位置重读的概率。

首先，我们尝试仅以单词的音节数为 feature。

不同的 classifier 有各自的特点，为找到最佳 classifier，我们对以下三种经典的 classifier，在以单词的元音数作为 feature 的基础上分别进行测试。

Classifiers	Avg F1
Decision Tree	0.7203
(Gaussian) Naive Bayes	0.7056
KNN	0.6998

Avg F1：0.7086

效果分析：

用单词的元音数作为 feature 的平均 F1 能达到 0.70 左右。但是只采用一个 feature 肯定是欠拟合(underfit)的。因此还需寻找新的有效的 feature。

从平均 F1 来看，Decision Tree 稍有优势，因此之后的 model 都选用 Decision Tree 作为 Classifier。

- Model2

Features：①单词的元音数

②前缀

③后缀

④首字母

⑤尾字母

Classifier：Decision Tree

思路：

对于人的学习而言，对一个单词的重音的最直观的预测方式便是看这个单词的结构、组成。而**前缀**和**后缀**这两个关键特征，对单词的发音有着极为重要的影响。

通常，对于前缀而言，带下列前缀的词：a-, ab-, ac-, ad-, al-, be-, con-, de-, dis-, em-, en-, in-, mis-,re-, un-重音通常在第二音节上。

而对于后缀而言，通常有下列 4 条规律：(1) 具有某些后缀的单词的重音位置不变，即与词根的重音一致。(2) 具有某些后缀的单词的重音，通常在第一个音节上。(3)具有某些后缀的单词的重音，一般在这些后缀的前一个音节上。(4) 词尾有 -ain, -ee, -eer, -ese 后缀的词，重音在该后缀上，而且有一个次重音。为便于记录，我们分别记后缀的 4 条规律为后缀 C1、后缀 C2、后缀 C3 和后缀 C4。

根据这些规律，我们建立前后缀与对应元音的关系，并辅以首尾字母作为features。

分析：

增加前缀、后缀和首尾字母作为feature后，平均F1可达到0.82左右。虽然有所提升，但是效果还是不够理想。我们认为，前缀和后缀对重音的影响不是绝对的。且具备前缀或后缀的单词占比并不是非常高。因此还需寻找更有效的feature。

- Model3

Features：①单词的元音数

②前缀

③后缀

④首字母

⑤尾字母

⑥重读开音节

⑦将元音编号后，取组成该单词的所有元音的整数串

思路：

单词的重读还与**音节发音**和**音节顺序**有着重大的关系。音节顺序最直观的表示方法即“单词的所有元音组成的整数串”。

我们将元音分别编号，取“单词的所有元音组成的整数串”作为 feature。

分析：

当前模型下，平均F1提升至0.87左右。当前模型体现了单词的“音节数与概率”“音节顺序”、“音节发音”和“特定发音规律”四大基本特征，结合机器学习和人脑学习的特点，加以组合形成我们的features。

此后，无论再加什么样的 feature，平均 F1 都没有太大改进，有时竟然还不升反降，于是我们决定确认当前features，并进行了第一次提交。至此，模型已经初具雏形，下一步进入模型优化阶段。

（二）模型优化

1、优化features

（1）增强某些好的features。

“单词的所有元音组成的整数串”这个feature表现佳,为使得这一feature有所增强，故增加了“最后两个音节组成的整数串”这一feature。

（2）从Importance看，剔除无用的feature，避免过拟合（overfit）。

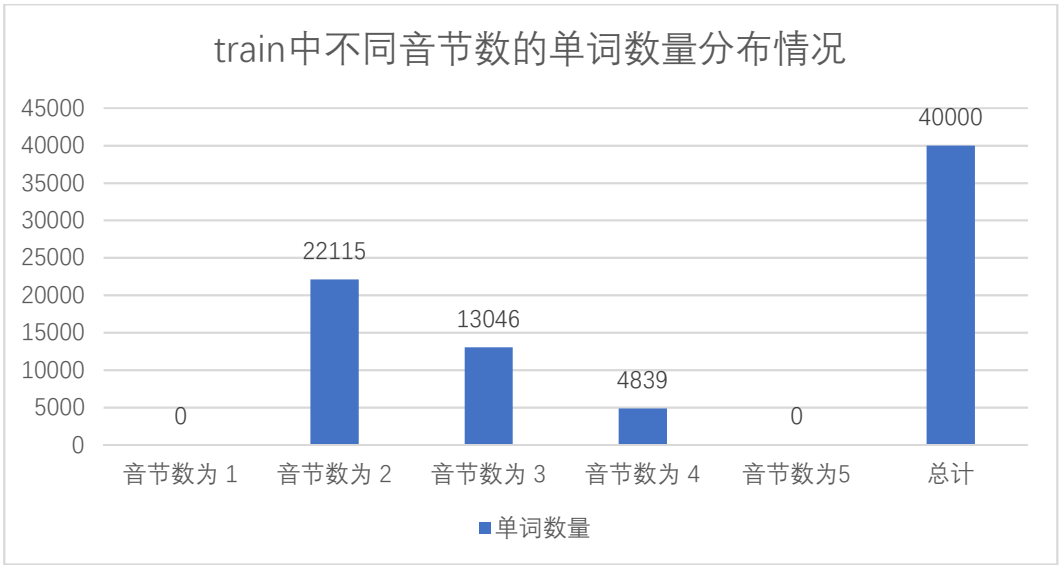
此过程淘汰的features有：前缀、后缀C1、C3、C4、首字母。

（3）完成以上两步优化后，对音节数为2、3、4、5的单词分别优化。（音节数

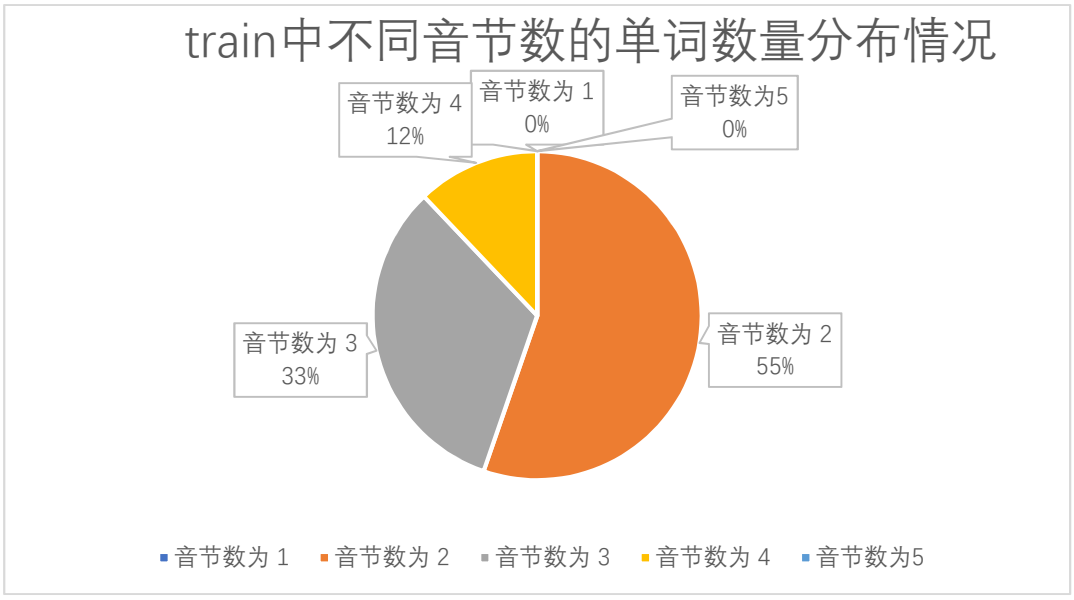
为 1 的单词无需预测其重音）

① 先对音节数为 1、2、3、4、5 的单词的数量做一个统计。

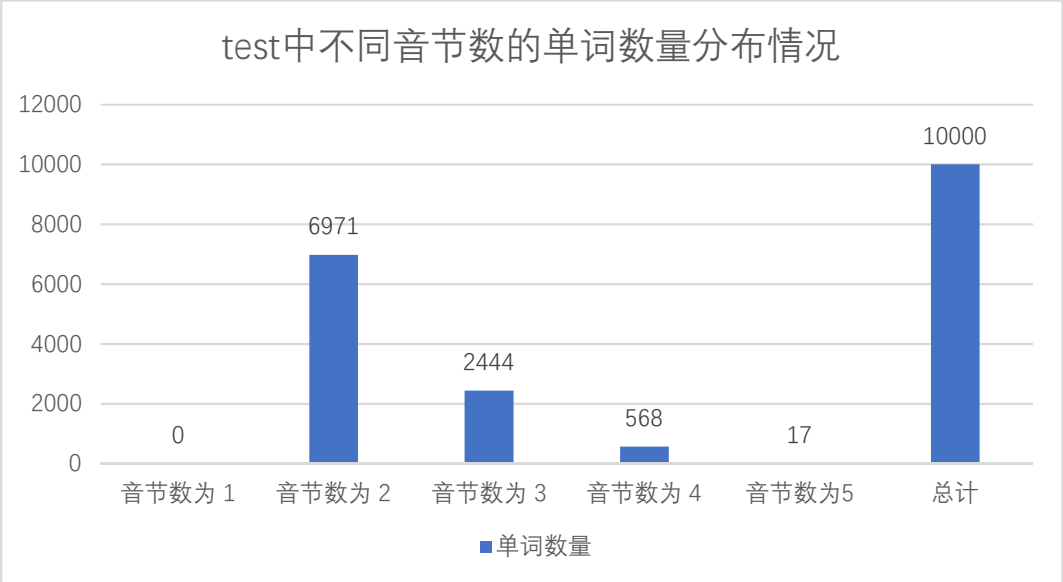
Ⅰ、train 中不同音节数的单词数量分布情况（柱状图）



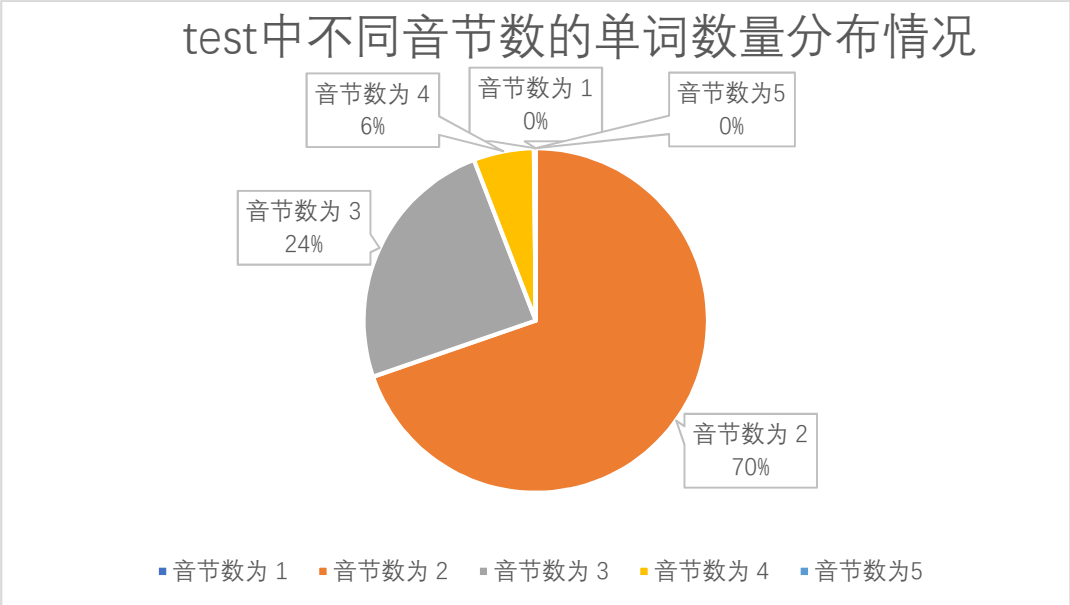
Ⅱ、train 中不同音节数的单词数量分布情况（饼图）



III、test 中不同音节数的单词数量分布情况（柱状图）



IV、test 中不同音节数的单词数量分布情况（饼图）



② 对音节数为 2、3、4 的预测准确率做一个统计。

	precision	recall	f1-score	support
1	0.91	0.94	0.93	6971
2	0.83	0.77	0.80	2444
3	0.79	0.74	0.76	568
4	1.00	0.12	0.21	17
avg / total	0.89	0.89	0.89	10000

分析：

结合①和②中的柱状图、饼图，我们可知：

音节数为 2 的占比大，预测准确率较高，效果较满意。

音节数为 3、4 的预测准确率不理想，尤其是音节数为 4 的部分，优化空间很大。此部分可作重点优化对象。

音节数为 5 的占比极小，虽然预测准确率不理想，但是训练数据少，优化难度大。此部分不予以优化。

于是，下一步我们重点对音节数为 3、4 的部分进行优化。

③ 对音节数为 3、4 的部分进行优化。

思路：

还是从音节发音入手。利用 shingles，将单词音标拆分成“辅音-元音-辅音”的形式。将音标组合 com1、2、3、4 作为 feature。

2、优化 classifier

（1）调整 DecisionTree 的 max depth

#clf = DecisionTreeClassifier(max_depth = max_depth)

max_depth	Avg F1 score
6	0.80
10	0.860
12	0.8684
14	0.873668
16	0.875687

分析：

在一定范围内，max_depth 与 F1 score 成正相关。

因此我们选取 16 作为 Decision Tree 的最大深度。

（2）M 个子集得到 M 个 Decision Tree

将新数据投入到这 M 个树中，得到 M 个分类结果，计数看预测成哪一类的数目最多，就将此类别作为最后的预测结果，形成 Random Forest。

（三）确定最终模型

最终选定的 Classifier：Random Forest

最终选定的 Features：如下表所示

Features	Information	Importance
单词的所有元音组成的整数串	音节顺序	0.04162116
辅音-元音-辅音 com1	音节发音	0.19195295
辅音-元音-辅音 com2	音节发音	0.21820994
辅音-元音-辅音 com3	音节发音	0.10649624
辅音-元音-辅音 com4	音节发音	0.00355649
尾字母	音节顺序	0.02196327
重读开音节	特定发音	0.10533304
后缀 C2	特定发音	0.08578312
最后两个音节组成的整数串	音节发音	0.15131929
单词的元音数	音节数与概率	0.07376449

五、 实验结果及截图

- Features:

单词的所有元音组成的整数串
辅音-元音-辅音 com1
辅音-元音-辅音 com2
辅音-元音-辅音 com3
辅音-元音-辅音 com4
尾字母
重读开音节
后缀 C2
最后两个音节组成的整数串
单词的元音数

- Classifier：Random Forest

- F1 score：0.9290

2017-07-31 00:01:35	F1 = 0.8720
2017-08-04 00:38:26	F1 = 0.8990
2017-08-07 01:19:04	F1 = 0.9290

六、 实验小结

由于我们是非英语为母语的学生，因此，在找 feature 上花了大量功夫，也做了大量工作。观察到汉语和英语的词汇重音有较大差异。

本次实验本质上是解决分类问题和预测问题的一次实践。本次实验中，在理解分类问题和预测问题的基础上，找对 feature 和 classifier 是关键。

我们的 feature 主要着眼于音节发音、音节顺序和特定发音规律三大方面，根据以上方面确定了“单词的所有元音组成的整数串、辅音-元音-辅音 com1、com2、com3、com4、尾字母、重读开音节、后缀 C2、最后两个音节组成的整数串、单词的元音数”等一系列 feature。根据 feature 特点，我们选用 Random Forest 作为本次实验的 classifier。

我们解决问题遵循以下思路：问题分析→讨论 feature→预处理→模型优化→可视化→确定模型。每个过程都有对应思路和相应关的分析。我们认为，可视化模块还可以做得更好。经过以上几个步骤，在不停的反思总结中，我们最终完成了 F1 在 0.9290 的单词重音预测，实验圆满结束。

参考文献：

[1] YJ Kim, MC Beutnagel. Automatic Assessment of American English Lexical Stress using Machine Learning Algorithms.

[2] J Hamilton and Jianna Jian Zhang. Learning English Stress Rules -Using a machine learning approach.