

# 数据挖掘 E-mail Classification

小组成员：

张 霖 2120151063

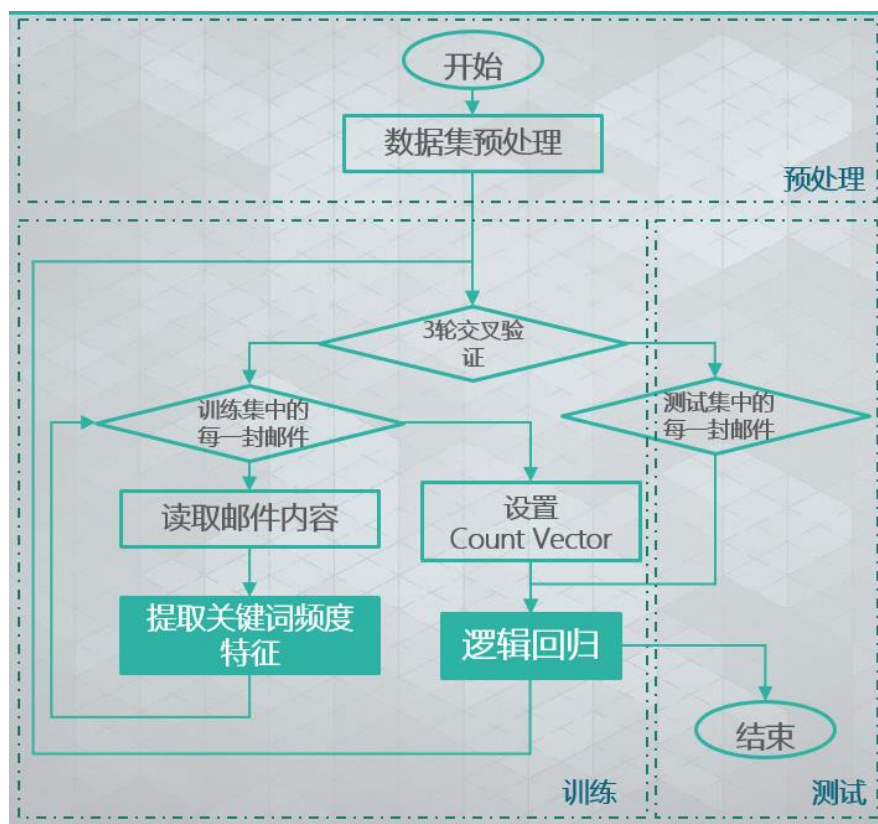
杨 冰 2120151052

## 1 邮件分类系统设计

我们的邮件分类系统大致步骤如下：

- (1) 首先训练分类器，提取文本的关键词频度特征。
- (2) 使用关键词频度训练逻辑回归分类器。
- (3) 交叉验证分类器效果。
- (4) 测试数据，观察分类结果。

系统流程图大致如下：



## 2 环境配置

### 2.1 环境配置

系统：Windows10  
开发语言：Python2.7.11  
IDE：Ipython Jupyter

## Enron E-mail Dataset

环境: Anaconda 数据分析环境

工具包: Sklearn, Nltk, pandas, numpy

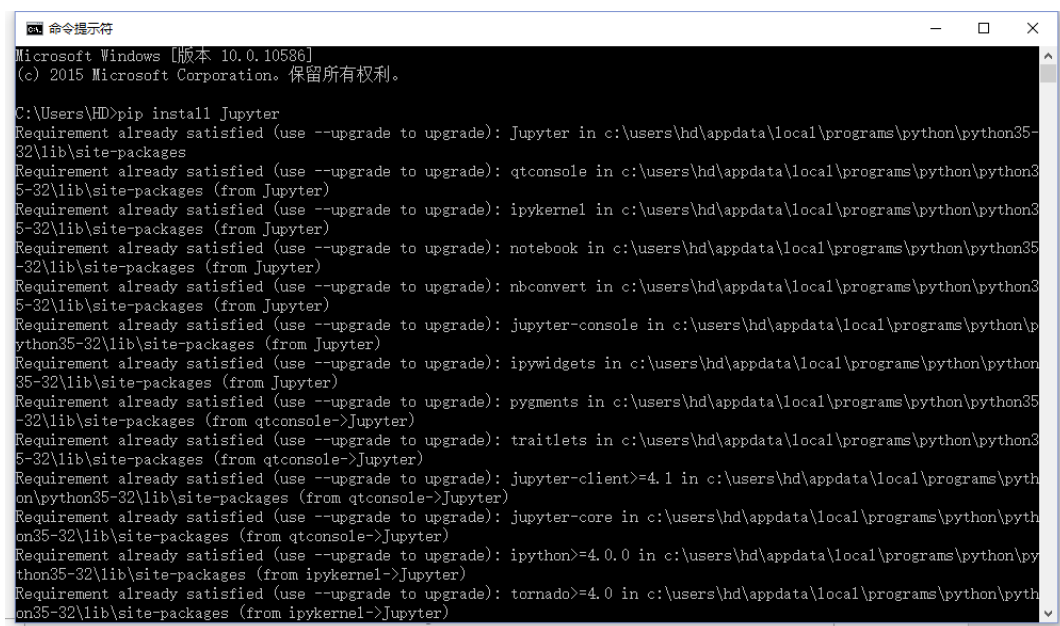
### 2.2 项目运行

(1) 下载安装 Python2.7.11。

(2) 下载安装 Anaconda3-4.0.0

(3) 安装 Ipython Jupyter

在 cmd 中输入 `pip install Jupyter`, 安装 python 的开发 IDE Jupyter。



```
命令提示符
Microsoft Windows [版本 10.0.10586]
(c) 2015 Microsoft Corporation. 保留所有权利。

C:\Users\HD>pip install Jupyter
Requirement already satisfied (use --upgrade to upgrade): Jupyter in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages
Requirement already satisfied (use --upgrade to upgrade): qtconsole in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from Jupyter)
Requirement already satisfied (use --upgrade to upgrade): ipykernel in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from Jupyter)
Requirement already satisfied (use --upgrade to upgrade): notebook in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from Jupyter)
Requirement already satisfied (use --upgrade to upgrade): nbconvert in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from Jupyter)
Requirement already satisfied (use --upgrade to upgrade): jupyter-console in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from Jupyter)
Requirement already satisfied (use --upgrade to upgrade): ipywidgets in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from Jupyter)
Requirement already satisfied (use --upgrade to upgrade): pygments in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from qtconsole->Jupyter)
Requirement already satisfied (use --upgrade to upgrade): traitlets in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from qtconsole->Jupyter)
Requirement already satisfied (use --upgrade to upgrade): jupyter-client>=4.1 in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from qtconsole->Jupyter)
Requirement already satisfied (use --upgrade to upgrade): jupyter-core in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from qtconsole->Jupyter)
Requirement already satisfied (use --upgrade to upgrade): ipython>=4.0.0 in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from ipykernel->Jupyter)
Requirement already satisfied (use --upgrade to upgrade): tornado>=4.0 in c:\users\hd\appdata\local\programs\python\python35-32\lib\site-packages (from ipykernel->Jupyter)
```

(4) 安装依赖环境包

在 cmd 中输入 `pip install Sklearn`, `pip install nltk`, `pip install pandas`, `pip install numpy`。依次安装完成。

(5) 项目运行

在命令行中输入 `cd` 命令跳转至代码所在文件夹中, 即 `E-mail-classification-Spam-filter-master` 中。

## Enron E-mail Dataset

```
命令提示符
2016/06/26 17:06 <DIR> Spam-Classification-Enron-Dataset-master
2016/06/24 19:43 32,176,004 Spam-Classification-Enron-Dataset-master.zip
2016/06/20 16:14 <DIR> Spamfilter-master
2016/06/20 15:39 802,475 基于用户行为的邮件分类算法.pdf
2016/06/20 14:07 74,235 机器学习-实验提交材料说明-2016.pdf
2016/06/27 14:08 676,047 机器学习Precipitation Prediction 实验报告.docx
2016/06/13 11:36 2,316,741 第二次：内存堆查看-沈啸东,马宝利.docx
2016/06/13 11:36 1,614,900 第二次：虚拟实验分类-沈啸东,马宝利.docx
20 个文件 274,442,525 字节
10 个目录 115,102,859,264 可用字节

C:\Users\HD\Desktop\机器学习>cd Spam-Classification-Enron-Dataset-master

C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master>dir
驱动器 C 中的卷没有标签。
卷的序列号是 00D5-63D6

C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master 的目录

2016/06/26 17:06 <DIR> .
2016/06/26 17:06 <DIR> ..
2016/06/24 19:44 <DIR> .ipynb_checkpoints
2016/06/24 19:44 <DIR> enron
2015/09/18 15:24 18,046 LICENSE
2016/06/26 17:06 15,402 main.ipynb
2015/09/18 15:24 255 README.md
3 个文件 33,703 字节
4 个目录 115,102,937,088 可用字节

C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master>
```

之后输入 `Ipython.exe notebook` 命令，启动 python 开发 IDE，此时会在 web 上建立一个 python 服务器，用来编辑 python 代码以及编译运行。

```
命令提示符 - ipython.exe notebook
20 个文件 274,442,525 字节
10 个目录 115,102,859,264 可用字节

C:\Users\HD\Desktop\机器学习>cd Spam-Classification-Enron-Dataset-master

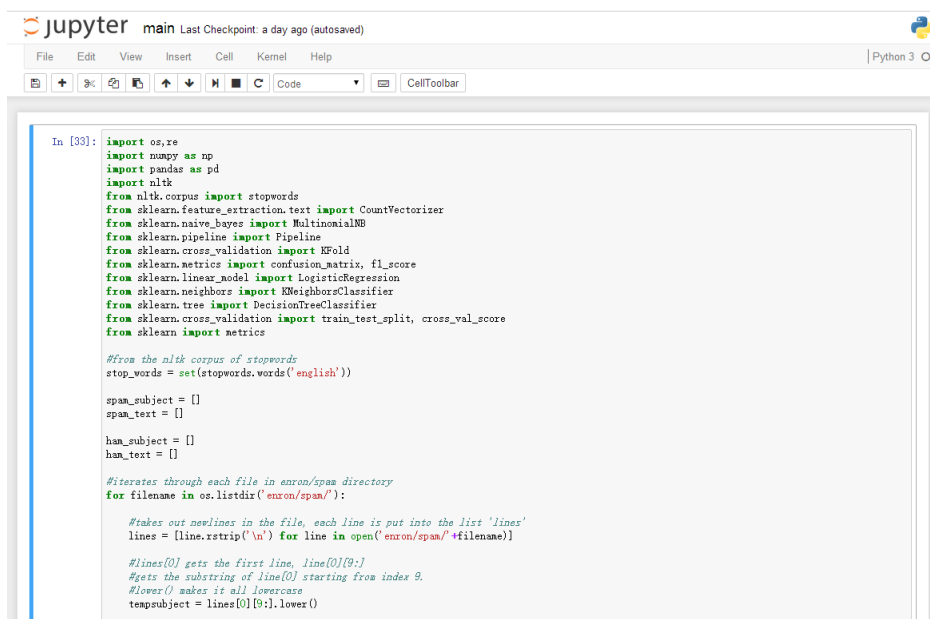
C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master>dir
驱动器 C 中的卷没有标签。
卷的序列号是 00D5-63D6

C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master 的目录

2016/06/26 17:06 <DIR> .
2016/06/26 17:06 <DIR> ..
2016/06/24 19:44 <DIR> .ipynb_checkpoints
2016/06/24 19:44 <DIR> enron
2015/09/18 15:24 18,046 LICENSE
2016/06/26 17:06 15,402 main.ipynb
2015/09/18 15:24 255 README.md
3 个文件 33,703 字节
4 个目录 115,102,937,088 可用字节

C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master>ipython.exe notebook
[TerminalIPythonApp] WARNING | Subcommand 'ipython notebook' is deprecated and will be removed in future versions.
[TerminalIPythonApp] WARNING | You likely want to use 'jupyter notebook' in the future
[15:42:04.136 NotebookApp] Serving notebooks from local directory: C:\Users\HD\Desktop\机器学习\Spam-Classification-Enron-Dataset-master
[15:42:04.138 NotebookApp] 0 active kernels
[15:42:04.139 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/
[15:42:04.140 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

# Enron E-mail Dataset



```
In [33]: import os, re
import numpy as np
import pandas as pd
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.cross_validation import KFold
from sklearn.metrics import confusion_matrix, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.cross_validation import train_test_split, cross_val_score
from sklearn import metrics

# from the nltk corpus of stopwords
stop_words = set(stopwords.words('english'))

spam_subject = []
spam_text = []

ham_subject = []
ham_text = []

# iterates through each file in enron/spam directory
for filename in os.listdir('enron/spam/'):

    # takes out newlines in the file, each line is put into the list 'lines'
    lines = [line.rstrip('\n') for line in open('enron/spam/' + filename)]

    # lines[0] gets the first line, line[0][9:]
    # gets the substring of line[0] starting from index 9.
    # lower() makes it all lowercase
    temp_subject = lines[0][9:].lower()
```

之后点击运行即可。

## 3 测试结果与分析

### 3.1 二分类结果

分类垃圾邮件的问题中,手动将训练数据分为两类:垃圾邮件 0 与非垃圾邮件 1,使用训练数据进行训练。交叉验证后所得回归模型的准确率预测率和召回率分别为:0.97, 0.97, 0.97, 如下图:

```
clf = LogisticRegression()

# 3-fold cross-validation of the model, 5-fold is more often used but if 3-fold performs well,
# then your model is golden.
# clf, counts, data['class'] is just the model, data/matrix, class-labels for each row in the matrix
# cv=3 is the number of folds in k-fold cross-validation (cross-validation = cv)
scores = cross_val_score(clf, counts, data['class'], cv=3)

# print out the average of the 3 values from the 3-fold cross-validation
print 'Accuracy: ', np.mean(scores)

# can make precision/recall/f1/etc. with different scoring
precisions = cross_val_score(clf, counts, data['class'], cv=3, scoring='precision_weighted')
print 'Precision: ', np.mean(precisions)
recalls = cross_val_score(clf, counts, data['class'], cv=3, scoring='recall_weighted')
print 'Recall: ', np.mean(recalls)
f1s = cross_val_score(clf, counts, data['class'], cv=3, scoring='f1_weighted')
print 'F1: ', np.mean(f1s)

Accuracy: 0.976677941962
Precision: 0.976783846269
Recall: 0.976677941962
F1: 0.976674495815
```

效果较优异。使用测试邮件如下进行测试,发现该邮件为垃圾邮件,逻辑回归分类器正确将其分类为 0,即垃圾邮件类。

## Enron E-mail Dataset

---

```
line = '''At the moment our team is looking for a manager in your area.
We are
looking for somebody who is ready to learn and start immediately. After
reviewing your CV, we came to the conclusion that you are an ideal
candidate for this job position.

Our company is engaged in providing services in the area of health
insurance. During our work, we have helped thousands of people around t
he
world and we earned an irrefutable reputation. Now you have the opportu
nity
to become a part of our friendly team.

Position requirements:
- You must be a US citizen.
- Your age must be more than 21 years.
- You must have a stable internet connection.
- You must be willing to learn and improve.

Position responsibilities:
- Keeping your projects documentation and filling of reports.
- Providing quality service to clients of the company.
- Perform the tasks on time.
- Close cooperation with other managers and our experts.

Your salary will be 3000 $ per month plus 500 $ every week. Also, you w
ill
have the personal bonuses. If you are ready to start working, respond t
o
this email. We will give you a trial period after which you can decide
this
job is right for you or not. Hope to hear from you soon.

Best regards,
Orli Irwin.'''
```

分类结果如下图:

## Enron E-mail Dataset

```
#gotta clean it the same way I did with the training examples for it to work properly
temptext = line.lower()
cleaned = [e for e in temptext.split(' ') if e.isalnum()]
cleaned = [word for word in cleaned if word not in stop_words and word!='']
cleaned = ' '.join(cleaned)
transformed = count_vectorizer.transform([cleaned])

#make the model, train the model, make a prediction.
clf = LogisticRegression()
clf.fit(counts, data['class'])

#probabilities for choosing a class. first in the array is 0's prob, next is 1's prob.
#picks the one with the highest prob.
print clf.predict_proba(transformed)

#spits out the highest prob prediction.
print clf.predict(transformed)

#tada

[[ 9.99542132e-01  4.57867503e-04]]
[0]
```

### 3.2 多分类结果

按照用户姓名分类邮件的问题中，手动将训练数据分为四类：用户 0，用户 1，用户 2，和用户 3，使用训练数据进行训练。交叉验证后所得回归模型的准确率预测率和召回率分别为：0.76, 0.80, 0.66，如下图：

```
clf = LogisticRegression()

#3-fold cross-validation of the model, 5-fold is more often used but if 3-fold performs well,
#then your model is golden.
#clf, counts, data['class'] is just the model, data/matrix, class-labels for each row in the matrix
#cv=3 is the number of folds in k-fold cross-validation (cross-validation = cv)
scores = cross_val_score(clf, counts, data['class'], cv=3)

#print out the average of the 3 values from the 3-fold cross-validation
print 'Accuracy: ', np.mean(scores)

#can make precision/recall/f1/etc. with different scoring
precisions = cross_val_score(clf, counts, data['class'], cv=3, scoring='precision_weighted')
print 'Precision: ', np.mean(precisions)
recalls = cross_val_score(clf, counts, data['class'], cv=3, scoring='recall_weighted')
print 'Recall: ', np.mean(recalls)
f1s = cross_val_score(clf, counts, data['class'], cv=3, scoring='f1_weighted')
print 'F1: ', np.mean(f1s)

Accuracy: 0.761563258452
Precision: 0.809652145832
Recall: 0.666325485125
F1: 0.763486521965
```

### 3.3 实验结果分析

可以看出，二分类邮件问题下的训练效果比多分类邮件问题下的训练效果好很多，这说明逻辑回归模型属于一个二分类模型，虽然可以用于多分类问题，但是效果总是不尽如人意。

## Enron E-mail Dataset

---

所以实验的改进可以考虑更换分类模型为多重线性回归模型，可以分类任意多类。同时可以考虑增加提取文本的特征，组成更长的特征向量，如果特征向量维数过高影响实验效率，可以考虑 PCA 降维等方法。