实验三

贝叶斯垃圾邮件识别

2022年6月30日

1. 读取数据

1）划分数据集

```
spam ../data/000/000
ham ../data/000/001
spam ../data/000/002
spam ../data/000/003
spam ../data/000/004
spam ../data/000/005
ham ../data/000/006
spam ../data/000/007
spam ../data/000/008
ham ../data/000/009
spam ../data/000/010
```

Label数据集的数据是这样的，前面的是邮件的种类，后面的是邮件的地址，所以可以根据这个读出数据集

```
line_list = []
with open('trec06c-utf8/label/index', 'r') as f:
    for line in f.readlines():
        line_data = {}
        label, email_file = line.split(' ')
        line_data['label'] = 1 if label == 'spam' else 0
        email_text = open(email_file.strip().replace('../data', 'trec06c-utf8/data_cut')).read()
        line_data['header'], line_data['body'] = split_email(email_text)
        line_list.append(line_data)
```

```
data = pd.DataFrame(line_list)
```

这里根据空行划分邮件头和邮件正文

```
def split_email(text):
    lines = text.split('\n')
    n = len(lines)
    for i in range(n):
        if len(lines[i].strip()) == 0:
            break
    return ' '.join(lines[:i]), ' '.join([line.strip() for line in lines[i:] if len(line.strip()) > 0])
```

可以得到的数据集为：

```
data = pd.DataFrame(line_list)
```

```
data.head()
```

	label	header	body
0	1	Received: from hp-5e1fe6310264 ([218.79.188.13...	[课 程 背 景] 每 一位 管理 和 技术人员 都 清楚 地 懂得， 单...
1	0	Received: from jdl.ac.cn ([159.226.42.8]) \tby...	讲 的 是 孔子 后人 的 故事 。 一个 老 领导 回到 家乡， 跟 儿子 感情 不 和...
2	1	Received: from 163.con ([61.141.165.252]) \tby...	尊敬 的 贵 公司 (财务 / 经理) 负责人 您好！ 我 是 深圳 金海 实业 有限...
3	1	Received: from 12.com ([222.50.6.150]) \tby sp...	贵 公司 负责人 (经理 / 财务) 您好： 深圳市 华龙 公司 受 多家 公司 委托...
4	1	Received: from dghhkjk.com ([59.36.183.208]) \...	这是 一封 HTML 格式 信件！ ----------...

2）划分数据集

```
from sklearn.model_selection import train_test_split, cross_validate # 划分数据集函数
RANDOM_SEED = 2020
# 划分训练集和测试集
def split(X, Y, test_size=0.2):
    x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=test_size, random_state=RANDOM_SEED)
    return x_train, x_test, y_train, y_test
```

使用正文数据作为模型的数据集，按照8:2划分训练和测试集

```
X = data['body']
Y = data['label']
x_train, x_test, y_train, y_test = split(X, Y)
print(len(x_train), len(x_test))
```

```
51696 12924
```

```
x_train.iloc[:10]
```

```
12641    当然 要 先 说 说 啊 一个 寝室 三个 人 ， 周末 或者 有时 一 室友 会 回家 ，...
54076    电子邮件 地址 库 分为 企业 用户 地址 ， 个人 地址 以及 国际 邮件地址 。 其中 ...
15196    您好 ！ 很 高兴 认识 您 。 我司 有意 与 您们 合作 ： 可 长久 给 您们 带来 ...
980      随着 市场经济 的 发展 ， 名优 商品 被 假冒 的 事件 也 屡 有 发生 ， 这 不仅...
5494     >  TO ; 负责人 : > >                                    ...
53188    没有 环境 ， 我 就 自己 创造 环境 ， 难到 非 得到 什么 大学 .. 什么 研究所...
10638    ☆ — — — — — — — — — — — — — — — — — — — — — — — — — — ...
62595    需要 用 小心翼翼 来 形容 的 联系 ， 亲爱 的 楼主 你 认为 它 能 持续 多久 ?...
18572    深 圳 市 安 科 达 实 业 有 限 公 司 ...
20513    第 98 届 交易会 联营 参展 邀请 尊敬 的 客户 ： 中国 出口商品 交易会 ， 又称...
Name: body, dtype: object
```

2.特征提取

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(min_df = 0.01, max_df = 0.8)
cv_fit = cv.fit_transform(x_train)
```

```
cv.get_feature_names_out()
```

```
array(['00', '01', '010', ..., '高级', '魅力', '麻烦'], dtype=object)
```

使用sklearn的CountVectorizer提取特征

可以展示出各个词的频次

```
In [12]: cv.vocabulary_
Out[12]: {'当然': 872,
          '一个': 169,
          '三个': 203,
          '或者': 955,
          '有时': 1134,
          '回家': 631,
          '大部分': 702,
          '时间': 1093,
          '所以': 960,
          '那个': 1706,
          '经常': 1445,
          '男朋友': 1321,
          '一次': 189,
          '正在': 1198,
          '外面': 684,
          '直接': 1336,
          '还有': 1668,
          '大家': 698,
          '怎么': 898,
```

也可以看到特征个数为1799

```
cv_fit.toarray().shape
```

```
(51696, 1799)
```

用这个模型提取训练集和测试集的文本特征

```
train_x_fit = cv.transform(x_train).toarray()
test_x_fit = cv.transform(x_test).toarray()
```

3. 构建模型

这里使用了MultinomialNB, BernoulliNB, ComplementNB三个朴素贝叶斯模型作为训练模型。

```
from sklearn.metrics import accuracy_score, precision_score, recall_score

model = {
    'MultinomialNB': MultinomialNB(),
    'BernoulliNB': BernoulliNB(),
    'ComplementNB': ComplementNB()
}

def model_predict(x_train, y_train, x_test, y_test):
    data = []
    for model_name ,clf in model.items():
        clf.fit(x_train, y_train)
        y_pred = clf.predict(x_test)
        data.append({
            'model': model_name,
            'accuracy': accuracy_score(y_pred,y_test),
            'precision': precision_score(y_pred, y_test),
            'recall': recall_score(y_pred, y_test)
        })
    return pd.DataFrame(data)
```

训练结果为：

```
model_predict(train_x_fit, y_train, test_x_fit, y_test)
```

	model	accuracy	precision	recall
0	MultinomialNB	0.957521	0.964444	0.971146
1	BernoulliNB	0.895466	0.862573	0.976692
2	ComplementNB	0.945373	0.944444	0.972189

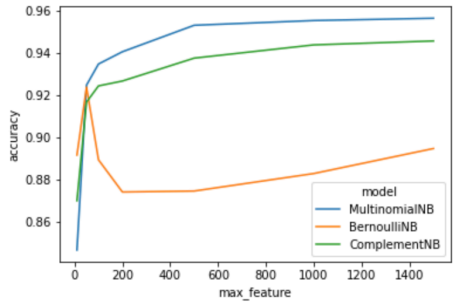
可以看出，多项式朴素贝叶斯的准确率和精准率都是最高的，三个模型的召回率差别不大，都有97%。接下来，对比特征数目（词表大小）对模型效果的影响。

```
def compare_cvsize(max_feature):
    cv = CountVectorizer(max_features=max_feature)
    cv.fit(x_train)
    train_x_fit = cv.transform(x_train).toarray()
    test_x_fit = cv.transform(x_test).toarray()
    df = model_predict(train_x_fit, y_train, test_x_fit, y_test)
    df['max_feature'] = max_feature
    return df
```

```
max_features = [10, 50, 100, 200, 500, 1000, 1500]
cmp_df = pd.DataFrame()
for max_feature in max_features:
        cmp_df = pd.concat([cmp_df, compare_cvsize(max_feature)], ignore_index=True)
```

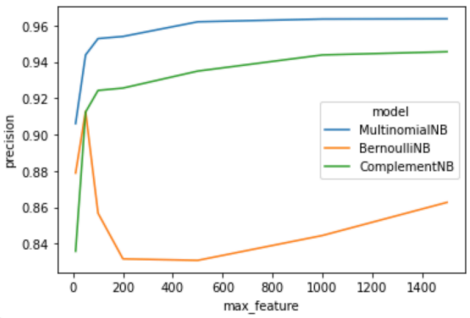
```
sns.lineplot(x='max_feature', y='accuracy', hue='model', data=cmp_df)
```

```
<AxesSubplot:xlabel='max_feature', ylabel='accuracy'>
```



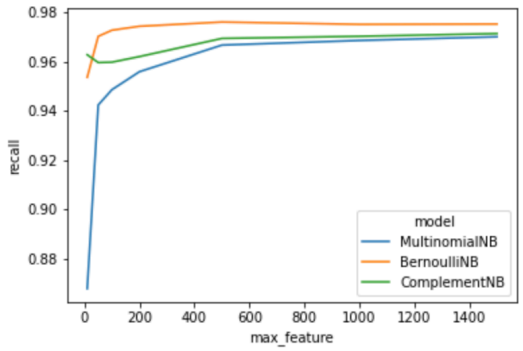

```
sns.lineplot(x='max_feature', y='precision', hue='model', data=cmp_df)
```

```
<AxesSubplot:xlabel='max_feature', ylabel='precision'>
```




```
sns.lineplot(x='max_feature', y='recall', hue='model', data=cmp_df)
```

```
<AxesSubplot:xlabel='max_feature', ylabel='recall'>
```



可以看出基本上特征数量增加，预测的准确率、精确度和召回率都会随之提升。总的来看MultinonmiaNB(多项式贝叶斯)的效果会比其他两个好，但是召回率比其他两个都低。在500个特征时模型基本收敛于96%的正确率。

BernoulliNB会有特征数量增加时，正确率和精确率降低的情况，在100个特征左右时，正确率可以达到最大值约为92%。

4. 优化

下面将从邮件头信息和TF-IDF两方面优化

```
data['header'][0]
```

```
'Received: from hp-5e1fe6310264 ([218.79.188.136]) \tby spam-gw.ccert.edu.cn (MIMEDefang) with ESMTP id j7CAoGvt02324
7 \tfor <lu@ccert.edu.cn>; Sun, 14 Aug 2005 09:59:04 +0800 (CST) Message-ID: <200508121850.j7CAoGvt023247@spam-gw.cce
rt.edu.cn> From: "yan"<(8月27-28,上海)培训课程> Reply-To: yan@vip.163.com"<b4a7r0h0@vip.163.com> To: lu@ccert.edu.cn Su
bject: =?gb2312?B?t8eyxs7xvq3A7bXEssbO8bncwO0to6jJs8XMxKPE4qOp?= Date: Tue, 30 Aug 2005 10:08:15 +0800 MIME-Version:
1.0 Content-type: multipart/related;     type="multipart/alternative";     boundary="----=_NextPart_000_004A_2531AAA
C.6F950005" X-Priority: 3 X-MSMail-Priority: Normal X-Mailer: Microsoft Outlook Express 6.00.2800.1158 X-MimeOLE: Pro
duced By Microsoft MimeOLE V6.00.2800.1441'
```

邮件头中有用的信息主要是发送人，接受人，还有邮件标题。可以用正则表达式提取这些数据。

```
import re
import base64
import jieba
def parse_sender(header):
    try:
        sender = re.search('From: ([^\s]*)', header).group(1)
        try:
            gbd = re.search('=\?GB2312\?B\?(.*)\?=', sender, re.I).group(1)
            sender = base64.b64decode(gbd).decode('gb2312')
        except:
            pass
    except:
        sender = ''
    return ' '+sender
def parse_receiver(header):
    try:
        receiver = re.search('To: ([^\s]*)', header).group(1)
    except:
        receiver = ''
    return ' '+receiver
def parse_subject(header):
    try:
        subject = re.search('Subject: ([^\s]*)', header).group(1)
        try:
            gbd = re.search('=\?GB2312\?B\?(.*)\?=', subject, re.I).group(1)
            subject = base64.b64decode(gbd).decode('gb2312')
        except:
            pass
    except:
        subject = ''
    return ' ' + " ".join(jieba.cut(subject))
```

```
data['sender'] = data['header'].map(parse_sender)
data['receiver'] = data['header'].map(parse_receiver)
data['subject'] = data['header'].map(parse_subject)
```

```
data.head()
```

	label	header	body	sender	receiver	subject
0	1	Received: from hp-5e1fe6310264 ([218.79.188.13...	[课 程 背 景] 每 一位 管理 和 技术人员 都 清楚 地 懂得， 单...	"yan"<(8月27-28,上海)培训课程>	yan@vip.163.com" <b4a7r0h0@vip.163.com>	非 财务经理 的 财务管理 - (沙盘 模拟)
1	0	Received: from jdl.ac.cn ([159.226.42.8]) \tby...	讲 的 是 孔子 后人 的 故事 。 一个 老 领导 回到 家乡， 跟 儿子 感情 不 和...	"pan"	shi@ccert.edu.cn	• 问 一部 魏宗万 的 电影 名称
2	1	Received: from 163.con ([61.141.165.252]) \tby...	尊敬 的 贵 公司 (财务 / 经理) 负责人 您好 ！ 我 是 深圳 金海 实业 有限...	张海南	xing@ccert.edu.cn	公司 业务 . 代 开发票 ！
3	1	Received: from 12.com ([222.50.6.150]) \tby sp...	贵 公司 负责人 (经理 / 财务) 您好 ： 深圳市 华龙 公司 受 多家 公司 委托...	代开发票	ling@ccert.edu.cn	低点 代 开发票 ！
4	1	Received: from dghhkjk.com ([59.36.183.208]) \...	这是 一封 HTML 格式 信件 ！ ---------- ------...	"mei"	tang@ccert.edu.cn	一边 上网冲浪 ， 一边 赚钱 ， 何乐而不为 ？

将这些数据组合在一起，并重新拆分数据集

```
data['body'] = data['body'] + data['sender'] + data['receiver'] + data['subject']
```

```
X = data['body']
Y = data['label']
x_train, x_test, y_train, y_test = split(X, Y)
```

使用TFIDF提取文本特征，预测结果为：

```
tfidf = TfidfVectorizer(min_df = 0.01)
tfidf.fit(x_train)
train_x_fit = tfidf.transform(x_train).toarray()
test_x_fit = tfidf.transform(x_test).toarray()
model_predict(train_x_fit, y_train, test_x_fit, y_test)
```

	model	accuracy	precision	recall
0	MultinomialNB	0.973228	0.973333	0.986019
1	BernoulliNB	0.928582	0.910292	0.980350
2	ComplementNB	0.965568	0.959064	0.988547

可以看到所有模型的准确率、精准率、召回率均高于0.9，MultinomialNB模型的整体预测效果依旧是最好的，融合了邮件头且用TFIDF提取特征，提升了1.5%左右。

5. 实验总结

本次实验使用了朴素贝叶斯进行垃圾邮件识别。一般来说，如果样本特征的分布大部分是连续值，使用ComplementNB会比较好。如果样本特征的分布大部分是多元离散值，使用MultinomialNB比较合适。而如果样本特征是二元离散值或者很稀疏的多元离散值，应该使用BernoulliNB。