

9_fake_news_detecting

吴清柳

August 3, 2023

1 检测假新闻

将借助 Sklearn 在数据集上建立一个 TfidfVectorizer, 然后初始化一个 PassiveAggressive Classifier 并将拟合. 在最后, 准确度分数和相关性矩阵将表明模型的效果.

2 TfidfVectorizer

TF(Term Frequency): 是一个单词在文章中出现的次数. 越高意味着一个单词相比其他单词出现的次数更多, 因此当这个单词是搜索词的一部分的时候, 这篇文章将是良好的搜索结果;
IDF(Inverse Document Freeuency): 在一篇文章中出现了很多次的单词也在其他的中出现了很多次, 可能是不相关的. IDF 就是一种判断一个单词在整个群中的重要性的方法;

TfidfVectorizer 可以将一系列的原文档转为 TF-IDF 特征矩阵.

3 PassiveAggressiveClassifier

Passive Aggressive 算法 (被动攻击性算法) 是一组在线学习算法. 这样的算法对于一个正确的分类输出保留保持, 对于错误的分类变得激进, 进行更新和调整. 不像大多数其他的算法, 该算法不收敛. 他的目的是造成更新来修复损失, 在权重向量中引起非常小的改变.

4 假新闻数据集介绍

本次的数据集为 news.csv, 其形状为 7796×4 . 第一列标识新闻, 第二盒第三列为标题和内容, 第四列是标识新闻是 REAL 还是 FAKE 的标签. 这个数据集大小为 29.2MB.

```
[1]: # import libraries
import numpy as np
import pandas as pd
import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

4.1 Read the data into a DataFrame

```
[4]: # read the data
df = pd.read_csv("../dataset/news.csv")

# get shape and head
print(df.shape)
df.head()
```

(6335, 4)

```
[4]: Unnamed: 0                                title \
0      8476                                You Can Smell Hillary' s Fear
1    10294  Watch The Exact Moment Paul Ryan Committed Pol...
2     3608                Kerry to go to Paris in gesture of sympathy
3    10142  Bernie supporters on Twitter erupt in anger ag...
4       875   The Battle of New York: Why This Primary Matters

                                text label
0  Daniel Greenfield, a Shillman Journalism Fello...  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu...  FAKE
2  U.S. Secretary of State John F. Kerry said Mon...  REAL
3  — Kaydee King (@KaydeeKing) November 9, 2016 T...  FAKE
4  It's primary day in New York and front-runners...  REAL
```

```
[3]: # get labels from the DataFrame
labels=df.label
labels.head()
```

```
[3]: 0    FAKE
1    FAKE
2    REAL
3    FAKE
4    REAL
Name: label, dtype: object
```

```
[5]: # split the dataset into training and testing sets
X_train,X_test,y_train,y_test=train_test_split(df['text'],labels,test_size=0.
↪2,random_state=7)
```

4.2 初始化一个 TfidfVectorizer

使用英语中的停止单词和最大的文章频率为 0.7(有更高的文章频率的单词会被丢弃). 停止词是在语言中在处理自然语言数据之前需要被拆选出去的最频繁的单词. 一个 TfidfVectorizer 可以把一组文章转变成 TF-IDF 特征矩阵.

```
[6]: # Initialize a TfidfVectorizer
tfidf_vectorizer=TfidfVectorizer(stop_words='english',max_df=0.7)
```

```
# Fit and transform train set, transform test set
tfidf_train=tfidf_vectorizer.fit_transform(X_train)
tfidf_test=tfidf_vectorizer.transform(X_test)
```

5 初始化一个 PassiveAggressiveClassifier

将其在 tfidf_train 和 y_train 上拟合

```
[8]: # initialize a Passive Aggressive Classifier
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train, y_train)

# predict on the test set and calculate accuracy
y_pred = pac.predict(tfidf_test)
score = accuracy_score(y_test, y_pred=y_pred)
print("Accuracy: {:.2f}%".format(round(score * 100, 2)))
```

Accuracy: 92.74%

最后, 打印一个混淆矩阵来查看真假阳性和阴性的数量

```
[9]: # build confusion matrix
confusion_matrix(y_test,y_pred,labels=['FAKE','REAL'])
```

```
[9]: array([[588,  50],
          [ 42, 587]])
```

在这个模型中, 有 588 个真阳性, 587 个真阴性, 42 个假阴性, 50 个假阳性.

6 总结

借助 TfidfVectorizer, PassiveAggressiveClassifier 来在新闻数据集上拟合模型, 获取了 92.74% 的准确度.