SVM 实验

SVM 算法原理

SVM 是一个用于分类问题的模型，其原理是要找到一个最大化最小间隔(样本点到分类超平面的间隔)的分类超平面。

实验步骤

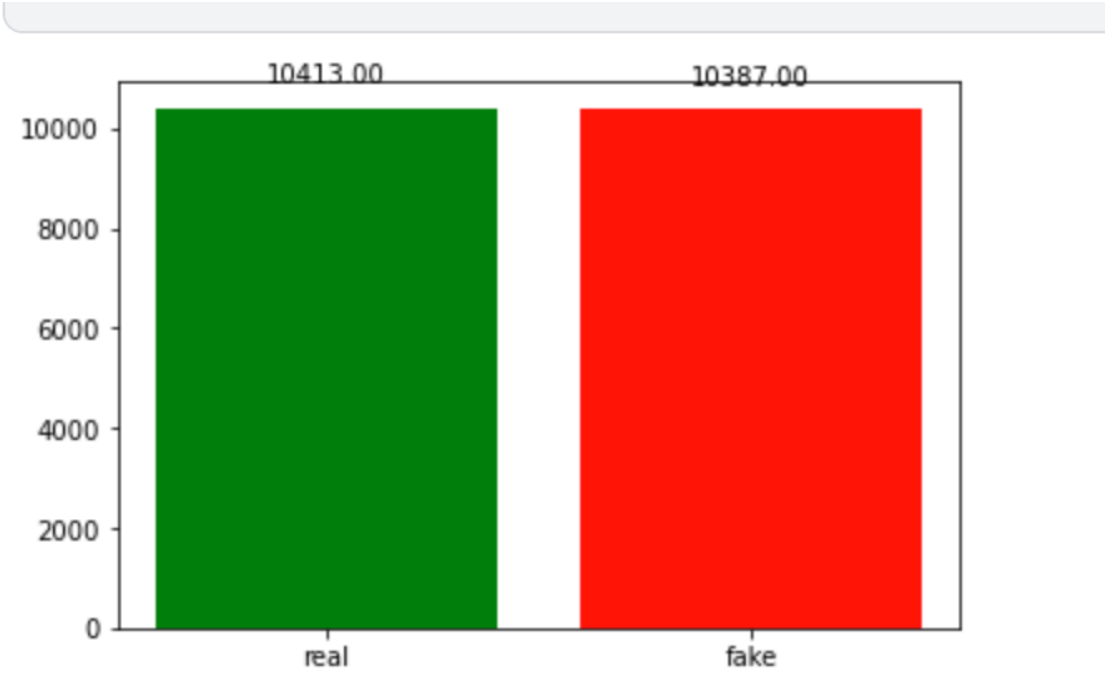
数据集概览: Fake News Detection

每个样本有五个特征:

- id: 每个文章的唯一 id
- title: 每个文章的标题
- author: 文章作者
- text: 文章内容
- label: 是否是 fake news (1 for yes, 0 for no)

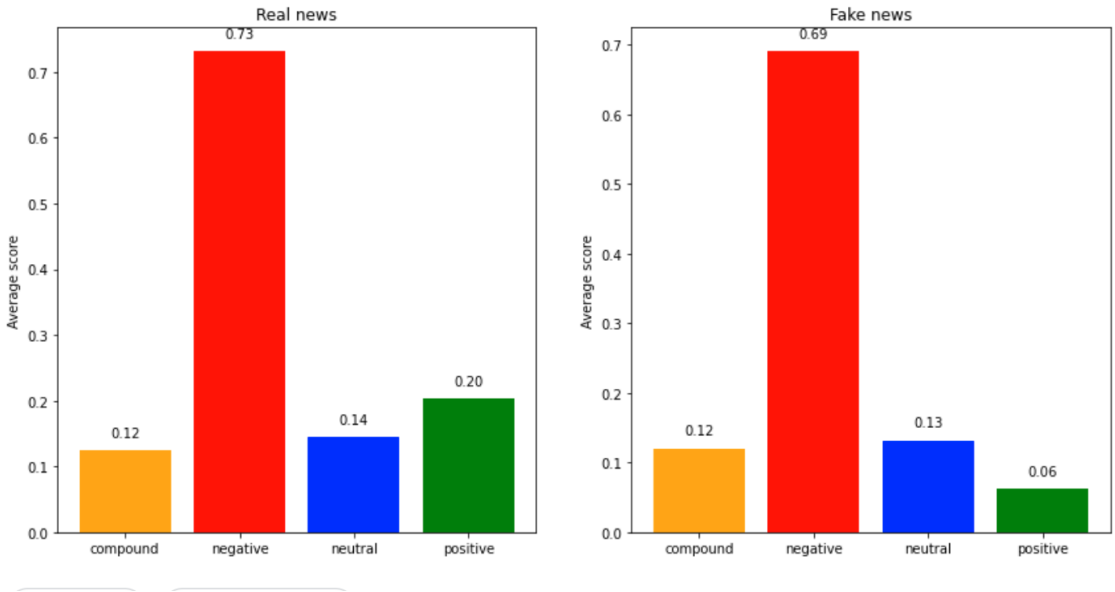
数据预处理和可视化

训练集所有文章在 reliable 和 unreliable 上的分布。



- 可以看出数据的分布是比较均衡的。
- 接下来使用 nltk 对文本进行一些预处理: 正则表达式去除标点并转换成小写 -> 分词 -> 去除停用词 -> lemmatize.

- ```
  def preprocessArticle(article):
      #Clean sentence to remove any punctuations, convert to lower case
      cleaned_sentence = re.sub(r'[^\w\s]', '', str(article).lower())
      #Tokenize sentence into words
      words = nltk.word_tokenize(cleaned_sentence)
      #Remove stop words and words with length less than equal to 3
      filtered_words = [word for word in words if not word in stop_words and
  len(word) > 3]
      #Lemmatize
      output_sentence = ''
      for word in filtered_words:
          output_sentence = output_sentence  + ' ' +
  str(lemmatizer.lemmatize(word))

      return output_sentence
  ```
- 之后用 nltk 的情感分析模块对预处理后的文本做一个粗略的情感分析，reliable 和 unreliable 的 article 中的情感分布如下图:
- 

- 可以看出无论在 real news 还是 fake news 中，negative 都是占多数的，这符合西方媒体的一般情况。但是在 fake news 中 positive 占比更少，这也符合 fake news 的一般套路，制造骇人听闻的消极新闻。

特征提取

此处尝试了几种策略，首先对 article 部分进行分词，使用了 unigram tfidf, unigram and bigram tfidf, unigram, bigram and trigram tfidf 作为特征分别做了实验。结果将在最后一个小节呈现。
  ```

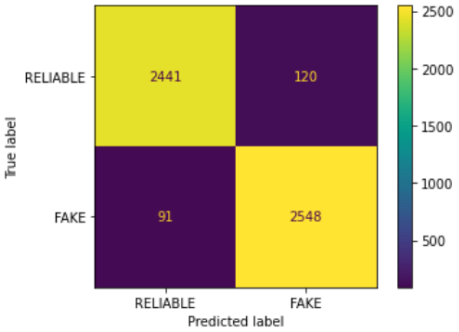
超参数设置

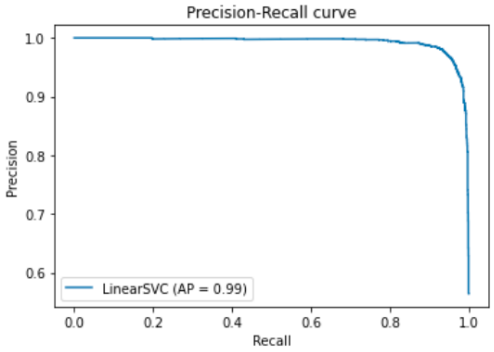
尝试了 C 为 1 和 0.001, max_iter 为 1000

实验结果

使用 tfidf-unigram, C = 1.0 的结果:

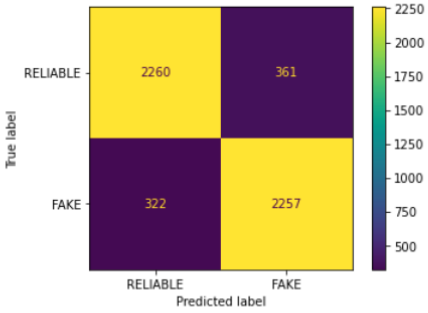
```
With linear kernel, standard regularization (inversely proportional to C) set to 1.0, using unigram tf-idf:
Accuracy: 95.94%
Precision (macro): 0.960
Precision (micro): 0.959
Recall (macro): 0.959
Recall (micro): 0.959
F1Score (macro): 0.959
F1Score (micro): 0.959
```

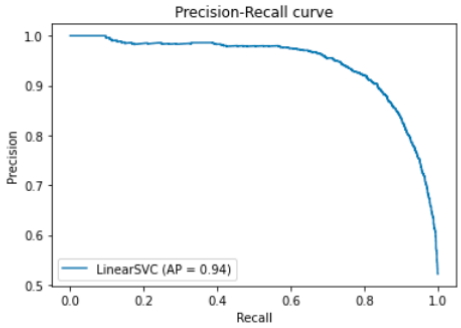




使用 tfidf-unigram, C = 1e-3 的结果:

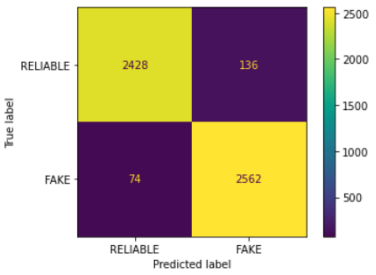
```
With linear kernel, more regularization (inversely proportional to C) set to 1.0e-3, using unigram tf-idf:
Accuracy: 86.87%
Precision (macro): 0.869
Precision (micro): 0.869
Recall (macro): 0.869
Recall (micro): 0.869
F1Score (macro): 0.869
F1Score (micro): 0.869
```

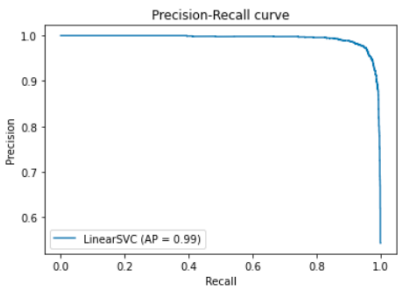




使用 tfidf-unigram and bigram, C = 1.0 的结果:

```
With linear kernel, standard regularization (inversely proportional to C) set to 1.0, using unigram and bigram tf-idf:
Accuracy: 95.96%
Precision (macro): 0.960
Precision (micro): 0.960
Recall (macro): 0.959
Recall (micro): 0.960
F1Score (macro): 0.960
F1Score (micro): 0.960
```





使用 tfidf-unigram, bigram and trigram, C = 1.0 的结果:

```
With linear kernel, standard regularization (inversely proportional to C) set to 1.0, using unigram, bigram and trigram tf-idf:
Accuracy: 95.52%
Precision (macro): 0.955
Precision (micro): 0.955
Recall (macro): 0.955
Recall (micro): 0.955
F1Score (macro): 0.955
F1Score (micro): 0.955
```