07_Covid_Fake_News_Classification

May 1, 2022

0.1 Covid Fake New Classification Using Logistic Regression

In this session, we are going to create a simple logistic regression model to classify COVID news to be either true or fake.

The process is surprisingly simple and easy. We will clean and pre-process the text data, perform feature extraction using NLTK library, build and deploy a logistic regression classifier using Scikit-Learn library, and evaluate the model's accuracy at the end.

The data set contains 586 true news and 578 fake news, almost 50/50 split.

[1]:
```
# import libraries
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
import pickle
from sklearn.linear_model import LogisticRegressionCV
import re
import pandas as pd
```

[2]:
```
df = pd.read_csv('data/corona_fake.csv')
```

[3]:
```
df.head()
```

[3]:

	title	text	source	label
0	Due to the recent outbreak for the Coronavirus…	You just need to add water, and the drugs and …		
1	NaN	Hydroxychloroquine has been shown to have a 10…		
2	NaN	Fact: Hydroxychloroquine has been shown to hav…		
3	NaN	The Corona virus is a man made virus created i…		
4	NaN	Doesn't @BillGates finance research at the Wuh…		

```
0  coronavirusmedicalkit.com  Fake
1               RudyGiuliani  Fake
2                CharlieKirk  Fake
3    JoanneWrightForCongress  Fake
4    JoanneWrightForCongress  Fake
```

[4]:
```
df.shape
```

[4]:
```
(1164, 4)
```

[5]:
```
df['label'].value_counts()
```

[5]:
```
TRUE    584
Fake    345
fake    230
Name: label, dtype: int64
```

[6]:
```
df.loc[5:15]
```

[6]:
```
                                                title  \
5   CORONA UNMASKED: Chinese Intelligence Officer …
6                                                 NaN
7                                                 NaN
8                                                 NaN
9   Basic protective measures against the new coro…
10                                                NaN
11                                                NaN
12                                                NaN
13                                                NaN
14  Exposing yourself to the sun or to temperature…
15  You can recover from the coronavirus disease (…

                                                text  \
5                                                 NaN
6   Urgent: Health Bulletin to the Public. Ministr…
7   Pls tell ur families, relatives and friendsMOH…
8   SERIOUS EXCELLENT ADVICE by Japanese doctors t…
9   Stay aware of the latest information on the CO…
10  The new Coronavirus may not show signs of infe…
11  A vaccine meant for cattle can be used to figh…
12  Using a hair dryer to breathe in hot air can c…
13  Corona virus before it reaches the lungs it re…
14  You can catch COVID-19, no matter how sunny or…
15  Most of the people who catch COVID-19 can reco…

                                               source label
5                                                 NaN   NaN
```

```
6                                  Ministry of Health  Fake
7                                               NWLLAB  Fake
8           Japanese doctors treating COVID-19 cases  Fake
9   https://www.who.int/emergencies/diseases/novel…  TRUE
10                                      Taiwan Experts  Fake
11                                            facebook  Fake
12                                             Youtube  Fake
13                                             twitter  Fake
14  https://www.who.int/emergencies/diseases/novel…  TRUE
15  https://www.who.int/emergencies/diseases/novel…   NaN
```

[7]:
```
df.isna().sum()
```

[7]:
```
title     82
text      10
source    20
label      5
dtype: int64
```

Because the data collection bias, lets not to use `source` as one of the features, instead, will combine `title` and `text` into one feature `title_text`.

[8]:
```
df.loc[df['label'] == 'Fake', ['label']] = 'FAKE'
df.loc[df['label'] == 'fake', ['label']] = 'FAKE'
df.loc[df['source'] == 'facebook', ['source']] = 'Facebook'
df.text.fillna(df.title, inplace=True)

df.loc[5]['label'] = 'FAKE'
df.loc[15]['label'] = 'TRUE'
df.loc[43]['label'] = 'FAKE'
df.loc[131]['label'] = 'TRUE'
df.loc[242]['label'] = 'FAKE'

#df = df.sample(frac=1).reset_index(drop=True)
df.title.fillna('missing', inplace=True)
df.source.fillna('missing', inplace=True)

df['title_text'] = df['title'] + ' ' + df['text']
```

[9]:
```
df.head(15)
```

[9]:
```
                                               title  \
0   Due to the recent outbreak for the Coronavirus…
1                                           missing
2                                           missing
3                                           missing
4                                           missing
```

```
5   CORONA UNMASKED: Chinese Intelligence Officer …
6                                                missing
7                                                missing
8                                                missing
9   Basic protective measures against the new coro…
10                                               missing
11                                               missing
12                                               missing
13                                               missing
14  Exposing yourself to the sun or to temperature…

                                                   text  \
0   You just need to add water, and the drugs and …
1   Hydroxychloroquine has been shown to have a 10…
2   Fact: Hydroxychloroquine has been shown to hav…
3   The Corona virus is a man made virus created i…
4   Doesn't @BillGates finance research at the Wuh…
5   CORONA UNMASKED: Chinese Intelligence Officer …
6   Urgent: Health Bulletin to the Public. Ministr…
7   Pls tell ur families, relatives and friendsMOH…
8   SERIOUS EXCELLENT ADVICE by Japanese doctors t…
9   Stay aware of the latest information on the CO…
10  The new Coronavirus may not show signs of infe…
11  A vaccine meant for cattle can be used to figh…
12  Using a hair dryer to breathe in hot air can c…
13  Corona virus before it reaches the lungs it re…
14  You can catch COVID-19, no matter how sunny or…

                                              source label  \
0                          coronavirusmedicalkit.com  FAKE
1                                       RudyGiuliani  FAKE
2                                        CharlieKirk  FAKE
3                          JoanneWrightForCongress  FAKE
4                          JoanneWrightForCongress  FAKE
5                                           missing  FAKE
6                                Ministry of Health  FAKE
7                                             NWLLAB  FAKE
8          Japanese doctors treating COVID-19 cases  FAKE
9   https://www.who.int/emergencies/diseases/novel…  TRUE
10                                    Taiwan Experts  FAKE
11                                          Facebook  FAKE
12                                           Youtube  FAKE
13                                           twitter  FAKE
14  https://www.who.int/emergencies/diseases/novel…  TRUE

                                         title_text
0   Due to the recent outbreak for the Coronavirus…
```

```
1   missing Hydroxychloroquine has been shown to h…
2   missing Fact: Hydroxychloroquine has been show…
3   missing The Corona virus is a man made virus c…
4   missing Doesn't @BillGates finance research at…
5   CORONA UNMASKED: Chinese Intelligence Officer …
6   missing Urgent: Health Bulletin to the Public…
7   missing Pls tell ur families, relatives and fr…
8   missing SERIOUS EXCELLENT ADVICE by Japanese d…
9   Basic protective measures against the new coro…
10  missing The new Coronavirus may not show signs…
11  missing A vaccine meant for cattle can be used…
12  missing Using a hair dryer to breathe in hot a…
13  missing Corona virus before it reaches the lun…
14  Exposing yourself to the sun or to temperature…
```

[10]:
```
df.shape
```

[10]:
```
(1164, 5)
```

Pre-processing Let's have a look an example of the title text combination:

[11]:
```
df['title_text'][50]
```

[11]:
```
"missing And what's next, everyone will swallow and sit in silence, only Russia
and China will press, and the rest will be swallowed."
```

Looking at the above example of title and text, they are pretty clean, a simple text pre-processing would do the job. So, we will strip off any html tags, punctuation, and make them lower case.

[12]:
```
def preprocessor(text):
    text = re.sub('<[^>]*>', '', text)
    text = re.sub(r'[^\w\s]','', text)
    text = re.sub(r'[\n]', "", text)
    text = text.lower()
    return text

df['title_text'] = df['title_text'].apply(preprocessor)
df['title_text'][50]
```

[12]:
```
'missing and whats next everyone will swallow and sit in silence only russia and
china will press and the rest will be swallowed'
```

The following code combines tokenization and stemming techniques together, and then apply the techniques on `title_text` later.

[13]:
```
porter = PorterStemmer()
def tokenizer_porter(text):
```

```
    return [porter.stem(word) for word in text.split()]
```

0.1.1 TF-IDF

Here we transform `title_text` feature into TF-IDF vectors. - Because we have already convert `title_text` to lowercase earlier, here we set `lowercase=False`. - Because we have taken care of and applied preprocessing on `title_text`, here we set `preprocessor=None`. - We override the string tokenization step with our combination of tokenization and stemming we defined earlier. - Set `use_idf=True` to enable inverse-document-frequency reweighting. - Set `smooth_idf=True` to avoid zero divisions.

[14]:
```
tfidf = TfidfVectorizer(strip_accents=None,
                        lowercase=False,
                        preprocessor=None,
                        tokenizer=tokenizer_porter,
                        use_idf=True,
                        norm='l2',
                        smooth_idf=True)
X = tfidf.fit_transform(df['title_text'])
y = df.label.values
```

[15]:
```
TfidfVectorizer?
```

0.1.2 Logistic Regression for Document Classification

- Instead of tuning C parameter manually, we can use an estimator which is LogisticRegressionCV.
- We specify the number of cross validation folds `cv=5` to tune this hyperparameter.
- The measurement of the model is the accuracy of the classification.
- By setting `n_jobs=-1`, we dedicate all the CPU cores to solve the problem.
- We maximize the number of iterations of the optimization algorithm.
- We use pickle to save the model.

[16]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, \
                                                    test_size=0.3,␣
↪shuffle=False)

clf = LogisticRegressionCV(cv=5, scoring='accuracy', random_state=0, n_jobs=-1,␣
↪\
                           verbose=3, max_iter=300)
clf.fit(X_train, y_train)

fake_news_model = open('fake_news_model.sav', 'wb')
pickle.dump(clf, fake_news_model)
fake_news_model.close()
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 out of   5 | elapsed:   16.1s remaining:   24.2s
```

```
[Parallel(n_jobs=-1)]: Done   5 out of   5 | elapsed:   18.6s finished
```

0.1.3 Model Evaluation

- Use pickle to load our saved model.
- Use the model to look at the accuracy score on the data it has never seen before.

[17]:
```
filename = 'fake_news_model.sav'
saved_clf = pickle.load(open(filename, 'rb'))

saved_clf.score(X_test, y_test)
```

[17]:
```
0.9257142857142857
```

[18]:
```
from sklearn.metrics import classification_report, accuracy_score
y_pred = clf.predict(X_test)
print("---Test Set Results---")
#print("Accuracy with logreg: {}".format(accuracy_score(y_test, y_pred)))
print(classification_report(y_test, y_pred))
```

```
---Test Set Results---
              precision    recall  f1-score   support

        FAKE       0.91      0.89      0.90       132
        TRUE       0.93      0.95      0.94       218

    accuracy                           0.93       350
   macro avg       0.92      0.92      0.92       350
weighted avg       0.93      0.93      0.93       350
```

[ ]:
```

```