r/CFB | r/NFL

Ryan Mackie

Project Overview

Problem

- Training a model to classify a post based on a subreddit.

Content

- Data Collection
- Data Cleaning
- Pre-processing
- Modeling
- Evaluation

Data Collection

1. Using the pushshift.io we created our base URL

```
base_url = 'https://api.pushshift.io/reddit/search/submission'
```

2. Then our function to scrape our subreddits and store our data frames

```
def get_json(subreddit):
    # list of scrapes
    subs = []
    for i in range(20):
        print(f'Scrape {i+1} of 20.')
        # get data using requests library
        res = requests.get(url = base_url,
            params = {
                'subreddit': subreddit,
                'size': 100,
                'before': str(2*i)+"d"
            })
        # add each scrape to our list of scrapes
        subs.extend(res.json()['data'])
        # take it easy on the server
        time.sleep(5)
    df = pd.DataFrame.from_dict(subs)
    df.to_csv(subreddit + '.csv')
    return df
```

Data Cleaning

1. Checked for any null values in our target column

```
cfb['title'].isnull().sum()
```

2. Checked to make sure there were no duplicates

```
cfb.drop_duplicates(subset = 'title')
print(f'There are {len(cfb)} values in cfb')
```

3. Set up our data frame to only include the columns we are interested in

```
cfb = cfb[['subreddit','title']]
```

4. Combined our data frames and binarized our target variable

```
data['subreddit'] = (data['subreddit']=='CFB').astype(int)
```

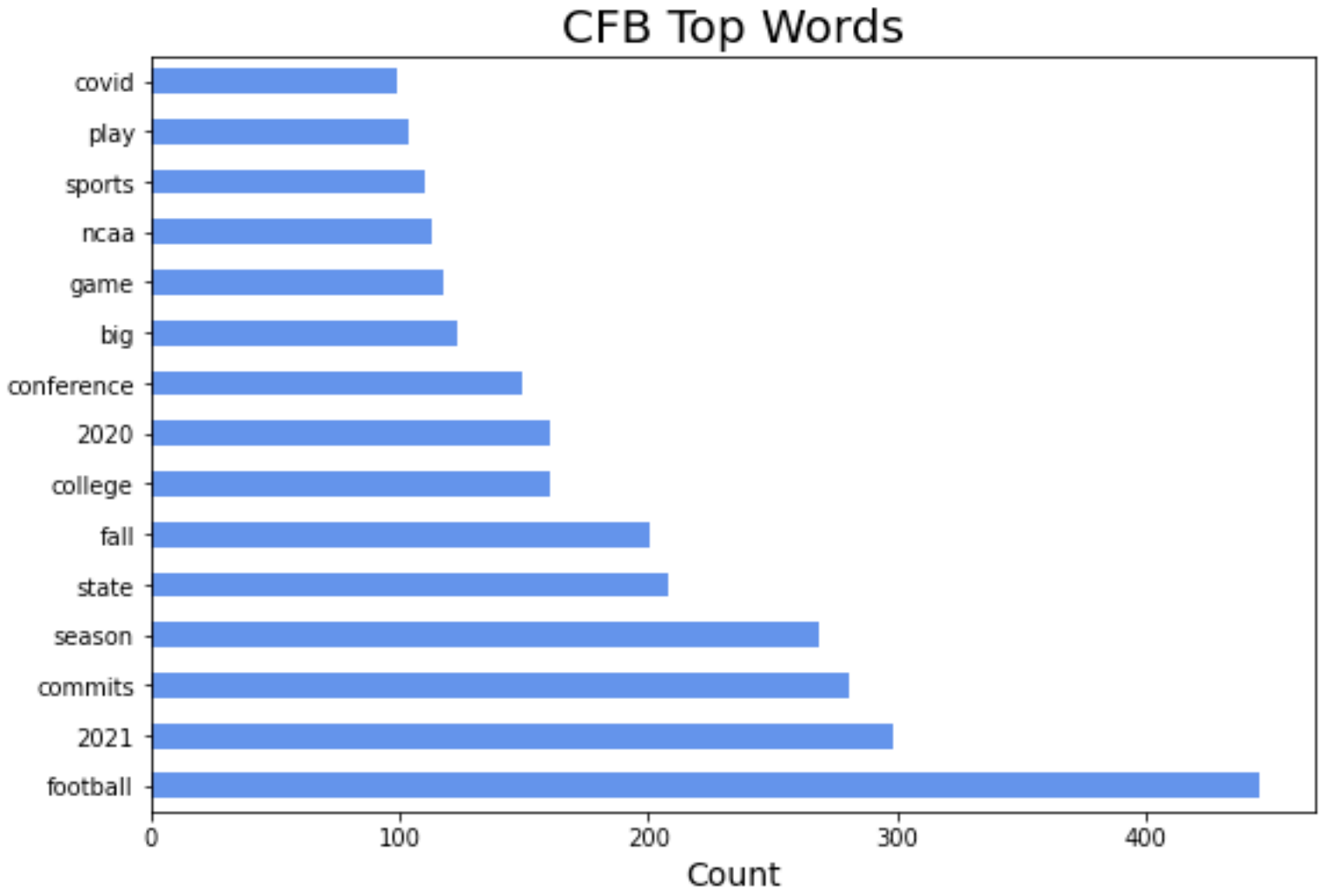
Pre-Processing

Using Count/Tfidf Vectorizer

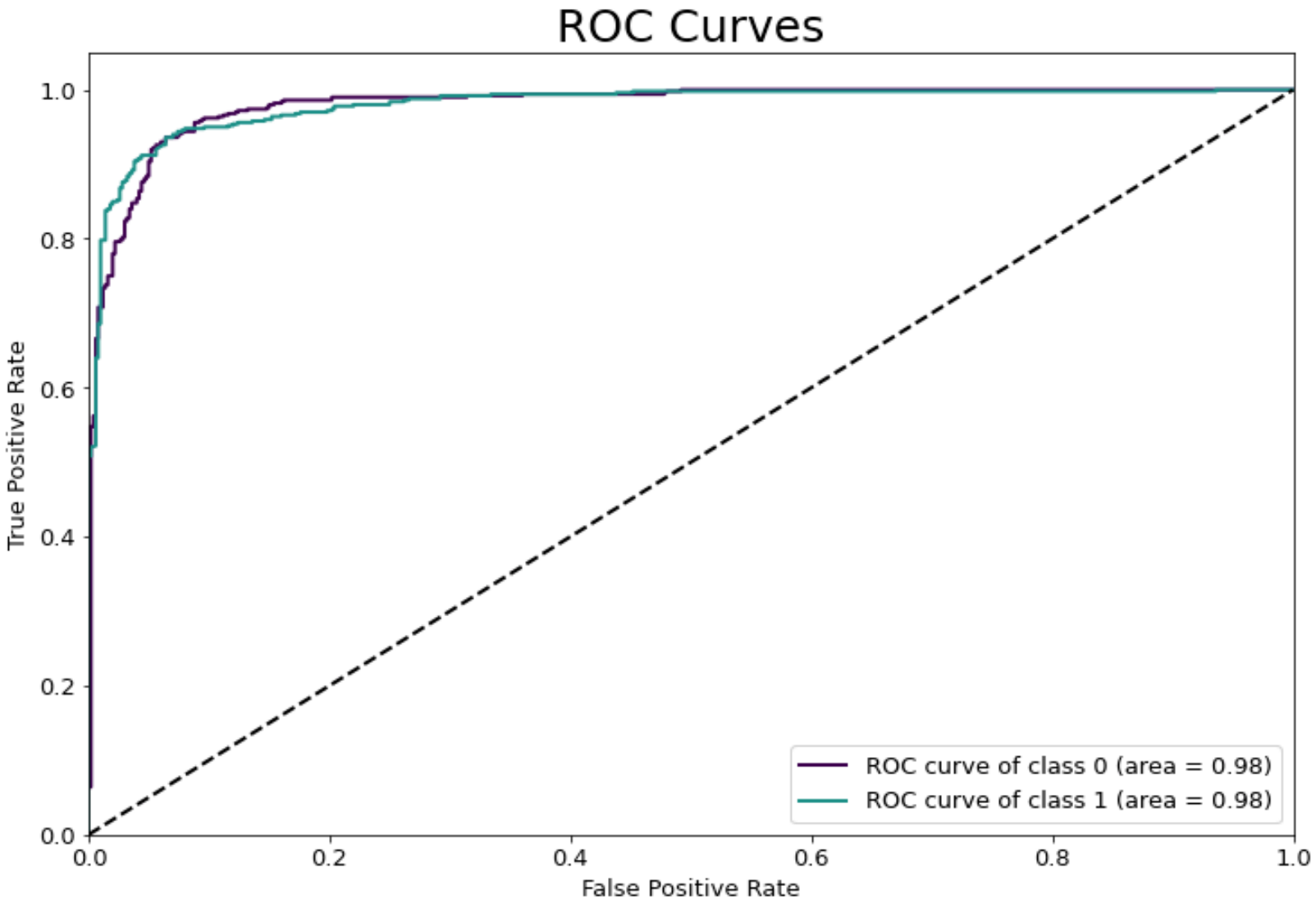
1. Tokenized our data
2. Removed special characters
3. Added stop words
4. Used our transformer to create some simple visualizations

Updated Stop Words

1. Created a new list of stop words
2. Updated the old list with our new list



Modeling



1. MultinomialNB
2. Random Forest
3. Support Vector Machines

Multinomial Naive Bayes

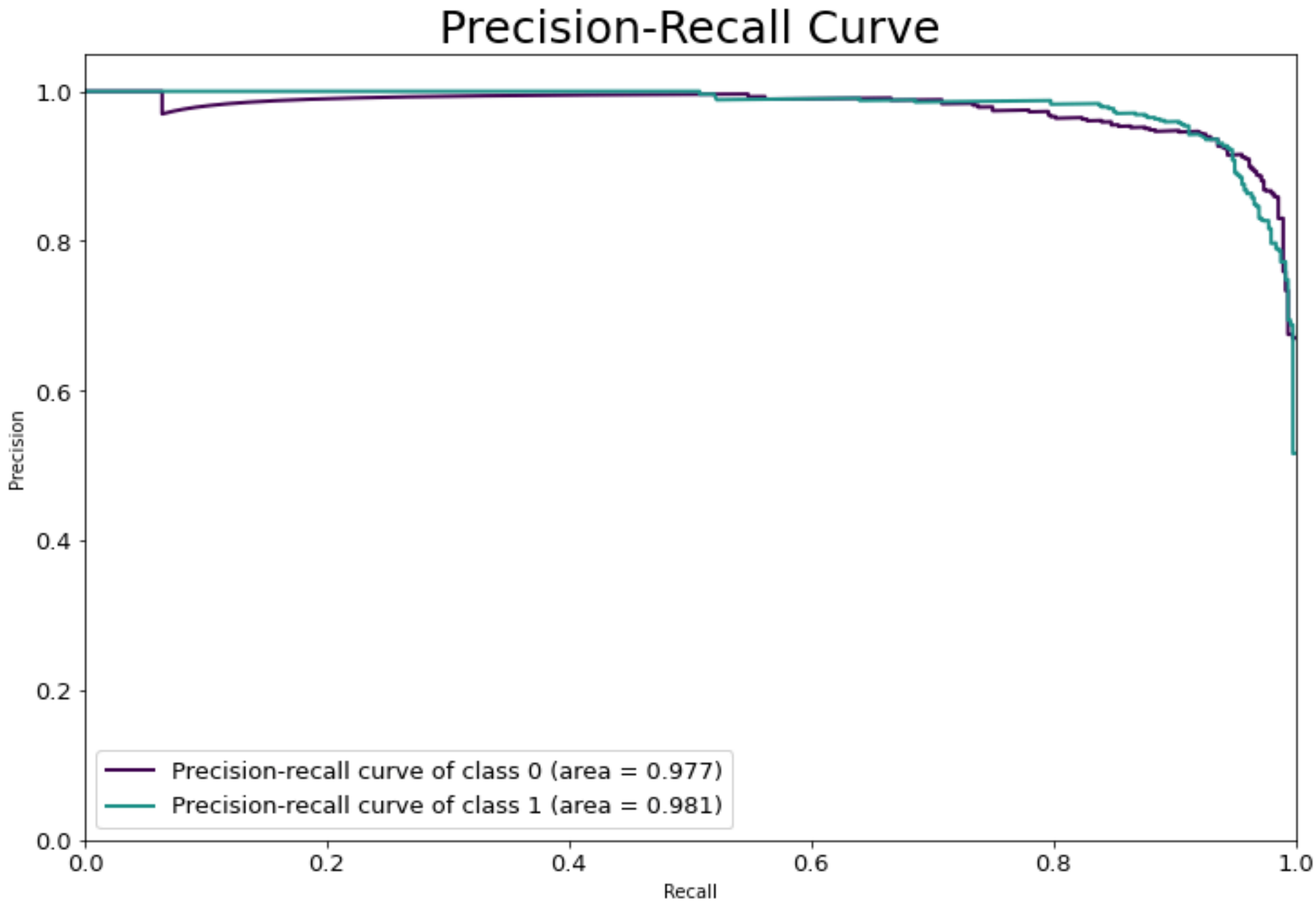
1. Fitted using both CountVectorizer and TfidfVectorizer
2. First model - overall best scores
3. Train/Test Scores: **0.974/0.943**

Random Forest

1. Train/Test Scores: **0.995/0.921**

Support Vector Machines

1. Train/Test Scores: **0.997/0.928**


Precision-Recall Curve
1.0
0.8
0.6
0.4
0.2
0.0
Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
Precision-recall curve of class 0 (area = 0.977)
Precision-recall curve of class 1 (area = 0.981)

Conclusions

1. Produced a fairly decent model using MultinomialNB - a bit overfit, but not bad

2. Collecting more data should help to improve the model performance

3. All of the special characters were removed - keeping certain values in our data should help the model as well

4. Working to customize the more complex models (Random Forest, SVM) would likely push them ahead of the NB model

Questions?

https://clutchpoints.com/wp-content/uploads/2020/08/Mahomes-Lamar-Lawrence-Fields.jpg

https://www.google.com/imgres?imgurl=https%3A%2F%2Fimg.bleacherreport.net%2Fimg%2Fimages%2Fphotos%2F002%2F752%2F606%2F06bfd9d1e14ea3c44472d8f466c48b9e_crop_north.jpg%3Fh%3D533%26w%3D800%26q%3D70%26crop_x%3Dcenter%26crop_y%3Dtop&imgrefurl=https%3A%2F%2Fbleacherreport.com%2Farticles%2F1958203-ncaa-proposes-changes-to-ejection-for-targeting-rule-defensive-substitutions&tbnid=ZuY6QIjmRKWR5M&vet=12ahUKEwiyvJ2j-7zrAhXGfK0KHadZC9sQMygKegUIARDDAQ..i&docid=HHdukzzd3M-T9M&w=800&h=533&q=targeting%20football&ved=2ahUKEwiyvJ2j-7zrAhXGfK0KHadZC9sQMygKegUIARDDAQ

https://sportsqandadotcom1.files.wordpress.com/2015/10/flag.jpg