Reddit post classifier

Rahul Kaw

What is the goal ?

To correctly predict or classify a Reddit post sub-reddit using one or more machine learning classification models

How ?

Gather reddit posts from two or more subreddits
Reddit posts are in free form text which includes emojis,links etc
Select attribute(s) of post to use for prediction
Clean up post body text
Tokenize
Lemmatize tokens
Conversion of each post's text into numerical vectors
Use classification models to predict category of each post

Selecting features

	title	selftext	subreddit	created_utc	author	num_comments	score	is_self
0	TIL that between 76 and 92% of Americans males...	NaN	todayilearned	1587576858	SpongebabeTwitch	0	1	False
1	TIL that 81% of American males (aged 14-59) ar...	NaN	todayilearned	1587577083	SpongebabeTwitch	0	1	False
2	TIL that the owo_bot account is rated more who...	[deleted]	todayilearned	1587577152	[deleted]	1	1	False
3	TIL that snakes can help predict earthquakes.	NaN	todayilearned	1587577341	robboss3000	2	2	False
4	Cortana is named after a HALO character, she i...	NaN	todayilearned	1587577425	potatomandude12345	0	1	False

Subreddits in data set;
todayilearned and *motorcycles*

	combinedtext	subreddit
0	TIL that between 76 and 92% of Americans males...	todayilearned
1	TIL that 81% of American males (aged 14-59) ar...	todayilearned
2	TIL that the owo_bot account is rated more who...	todayilearned
3	TIL that snakes can help predict earthquakes.	todayilearned
4	Cortana is named after a HALO character, she i...	todayilearned

60-40 distribution of 54,600 posts between two sub-reddits for modeling

	combinedtext	subreddit
0	5.11 Shield Boot as Riding Shoes?	motorcycles
1	Stolen Bike	motorcycles
2	Motorcycle Grudge racing edit	motorcycles
3	I got my entire vintage motorcycle collection ...	motorcycles
4	What is this? [pic1](https://ibb.co/WxYSyQg)\n...	motorcycles

Pre processing

Iterate over each Reddit post
Apply regular expression to strip out symbols,punctuations,urls,line breaks,tabs etc

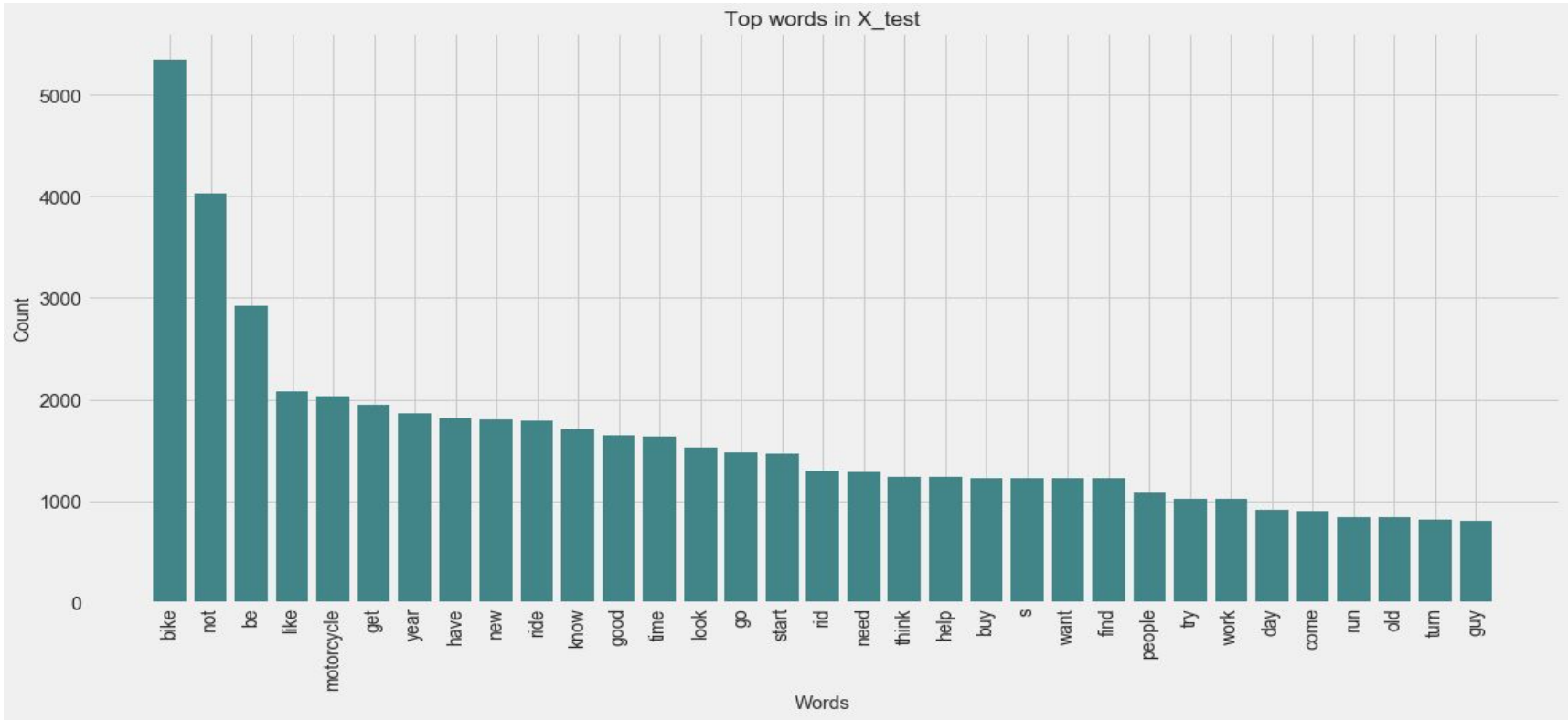
```
: body = 'TIL In 1959, nine Russian hikers died mysteriously in the Ural Mountains. 🚜 🕉 www.yahoo.com '\
  ' https://www.rahulkaw.com !$%^åå°Δμ¬^ΔΔ†¥ concluded 🛌 that 🇧🇩 the of a \"compelling natural force\"'

  regex = r"(.\\n|.\\t|.\\r)|(^TIL)|([^a-zA-Z0-9\s]+)|(ht[tp?s://a-zA-Z0-9.com]+)|(ww[wa-zA-Z0-9.com]+)"

  output = reg.sub(regex, '', body)
  output

: ' In 1959 nine Russian hikers died mysteriously in the Ural Mountains       concluded  that  the of a compelling natu
  ral force'
```

Pre-processing

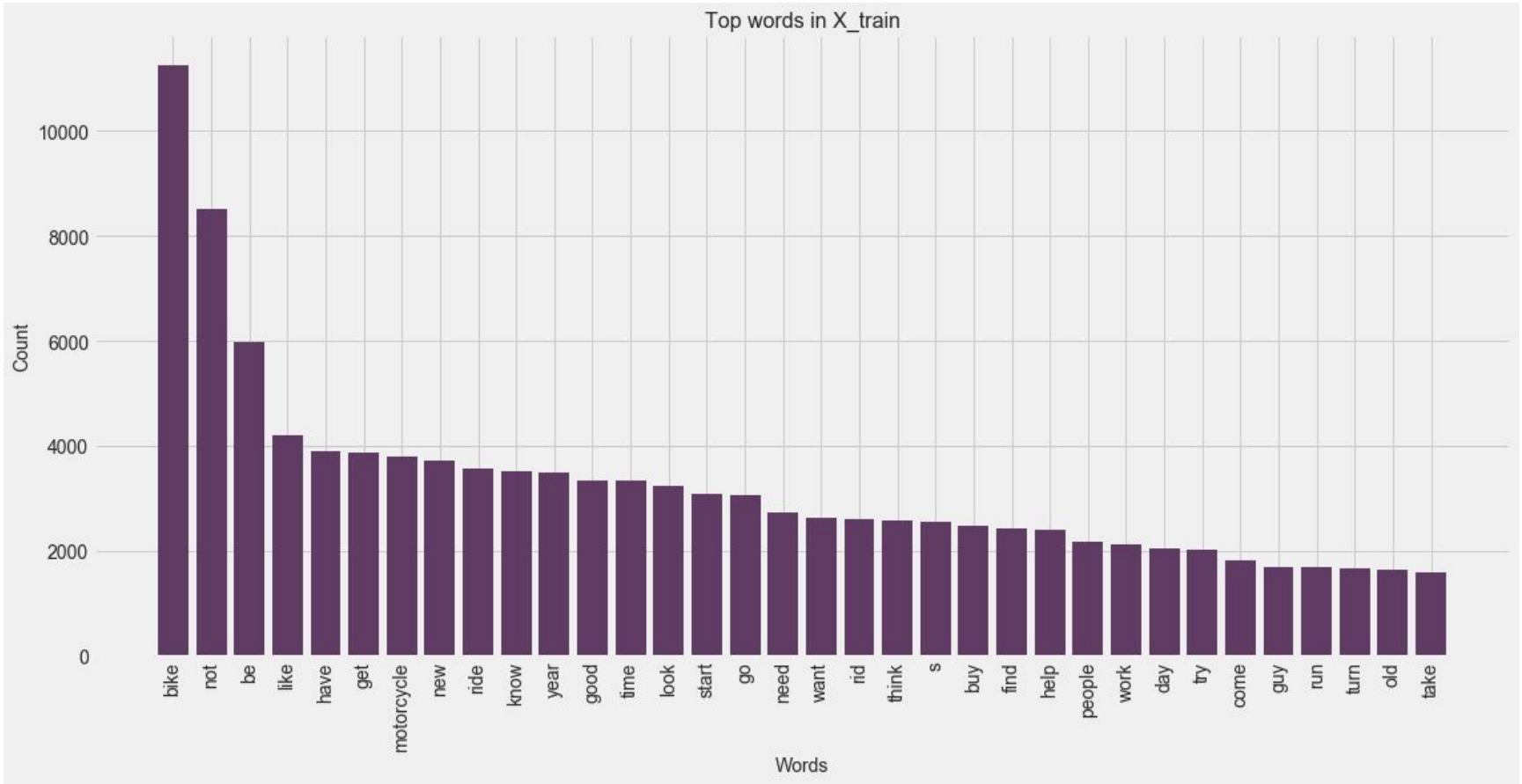
NLTK or spaCy?
spaCy: 179 vs 305 stop words. Well documented API. spaCy results in a compact document

	CountVectorizer with -No custom pre processing -CV object called with defaults *conversion was blinking fast*	CountVectorizer with -Custom functions for pre process, tokenize etc -CV object called with defaults *conversion was slower*	CountVectorizer with: -Overridden preprocessor and tokenizer for CV object call *conversion was slowest of all*
20,000 posts	20,200 features	9141 features	
54,600 posts	40,428 features	32,172 features	31,660 features
98,000 posts	45,343 features	37,450 features	

Word frequency



Word frequency



Estimator,transformer parameters

No pre-processing

CountVectorizer,[LogisticRegression,MultiNomialNB]
TfifdVectorizer,[LogisticRegression,MultiNomialNB]

min_df: 5 and 10
max_df: 0.7 and 0.9
stop_words: english and none
n_gram: (1,2) and (1,3)
max_features: 3000,4000,5000

Accuracy as compared to pre-processed data was bit lower

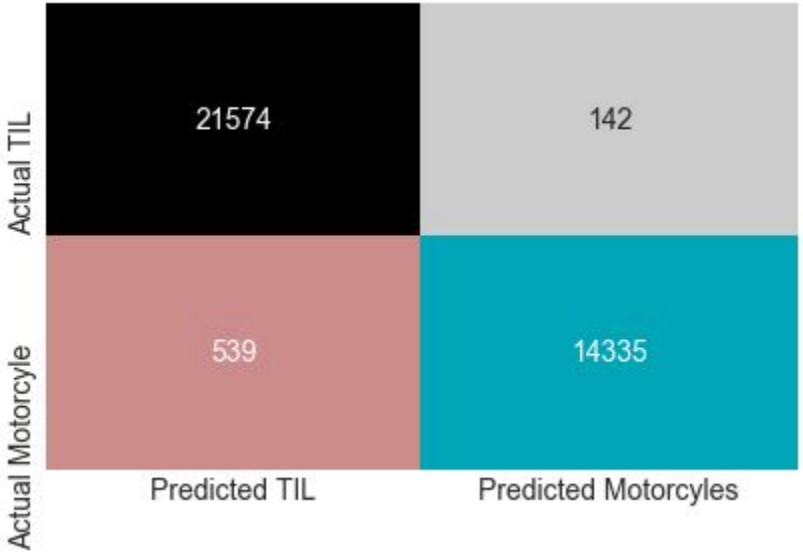
Pre-processed

CountVectorizer,[LogisticRegression,MultiNomialNB]
TfifdVectorizer,[LogisticRegression,MultiNomialNB]

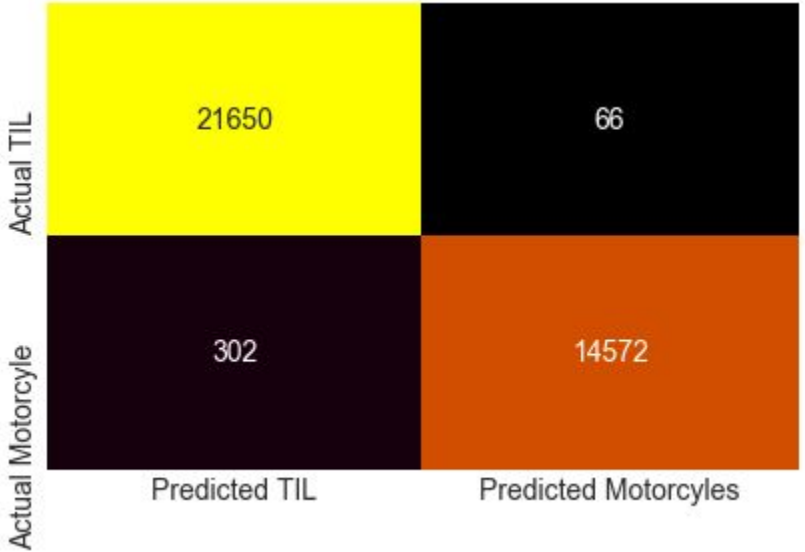
min_df: 5 and 10
max_df: 0.7 and 0.9
stop_words: none
n_gram: (1,3) and (1,2)
max_features: 3000,4000,5000

Accuracy was higher with pre-processed data

Results

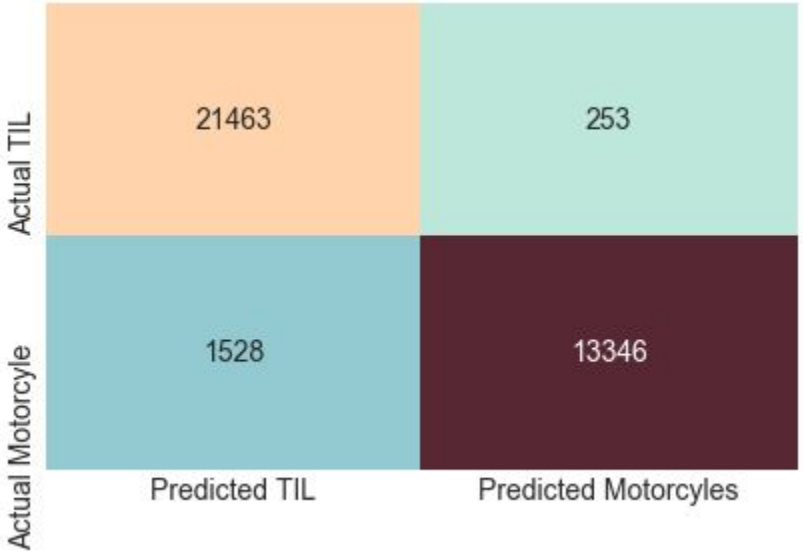


TfifdVectorizer+LogisticRegression

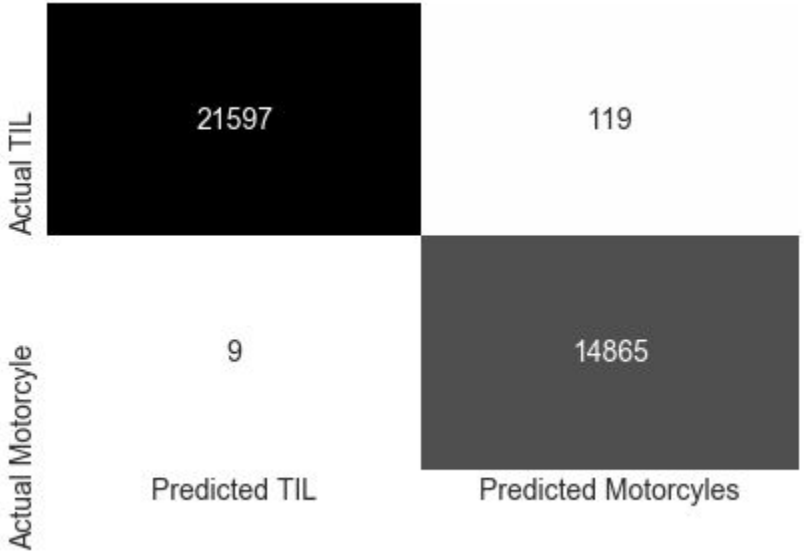


CountVectorizer+LogisticRegression

Results

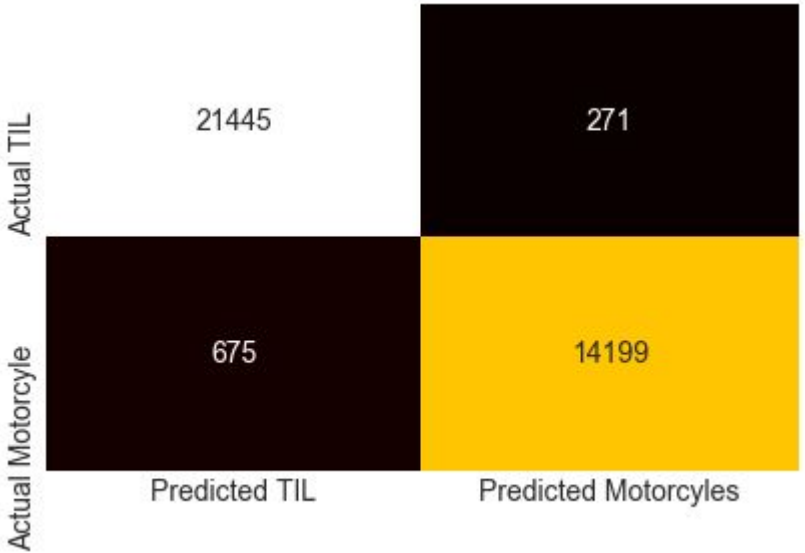


TfifdVectorizer+LogisticRegression

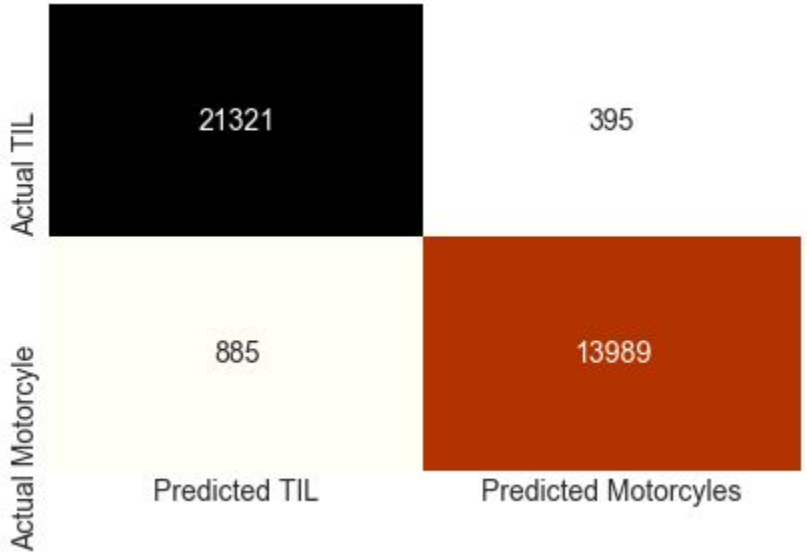


CountVectorizer+LogisticRegression

Results



TfifdVectorizer+MultinomialNB



CountVectorizer+MultinomialNB

Results


0.9925
0.9900
0.9875
0.9850
0.9825
0.9800
0.9775
0.9750
0.9725
Mean test score
30000
40000
50000
Max features

In the end

Model was fairly accurate in predicting the sub-reddit of a reddit post

Pre-processed data had higher accuracy

Model using one of the tree based models

Would same model be accurate on a very large data set?

```
y_train.value_counts()
y_train.value_counts(normalize = True)
```

```
0    21716
1    14874
Name: subreddit, dtype: int64

0    0.593495
1    0.406505
Name: subreddit, dtype: float64
```

```
actual_v_predicted_train.head()
actual_v_predicted_test.head()
```

	Actual	Predicted	Text
54421	1	1	royal jordanian go cinema portrait biker
41640	1	1	probably go say remember blindly trust rider l...
2881	0	0	streetbillionairementorcom visit save free shi...
26213	0	0	man asperger start successful company help asp...
12355	0	0	wwii woodenhaul devil boat boat operate pack l...

CountVectorizer vocabulary

```
[31]: count_vectorizer.vocabulary_

:[31]: {'royal': 24440,
        'jordanian': 14938,
        'go': 11636,
        'cinema': 5053,
        'portrait': 21980,
        'biker': 2829,
        'probably': 22414,
        'say': 24920,
        'remember': 23684,
        'blindly': 3051,
        'trust': 29453,
        'rider': 24104,
        'lead': 15982,
        'group': 12063,
        'streetbillionairementorcom': 27406,
        'visit': 30767,
        'save': 24894,
        'free': 10815,
        'shipping': 25754,
        'corner': 6138,
        'resource': 23865,
        'modify': 18380,
        'discount': 7797,
        'link': 16371,
        'enjoy': 9114,
        'school': 25026,
        'supply': 27787,
        'manufacture': 17205,
        'superstore': 27768,
        'service': 25499,
        'cbd': 4508,
        'cosmetic': 6222,
        'accessory': 150,
        'training': 29189,
        'program': 22474,
        'travel': 29263,
        'agency': 490,
        'booking': 3290,
        'play': 21703,
        'music': 18919,
        'man': 17112,
        'asperger': 1668,
```

Numbers are the position of the word in the sparse vector not how many times the word occurs in input data