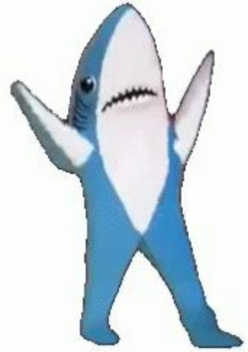
FAKE NEWS

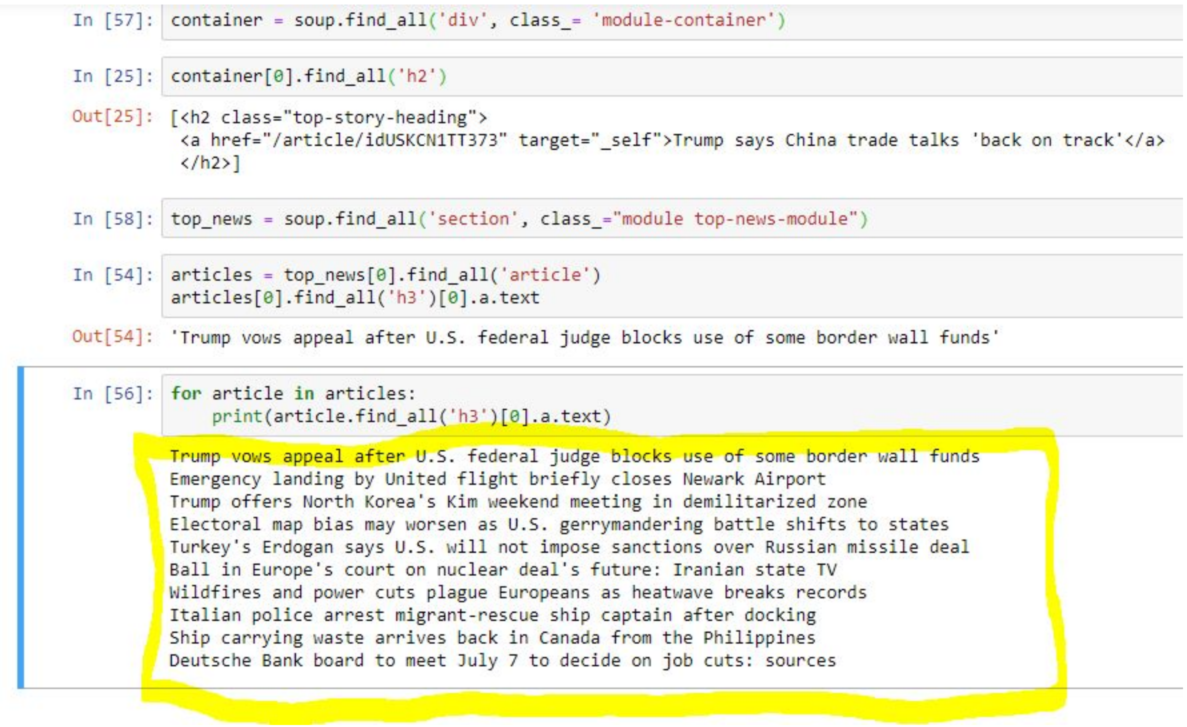
By: Team Left Shark



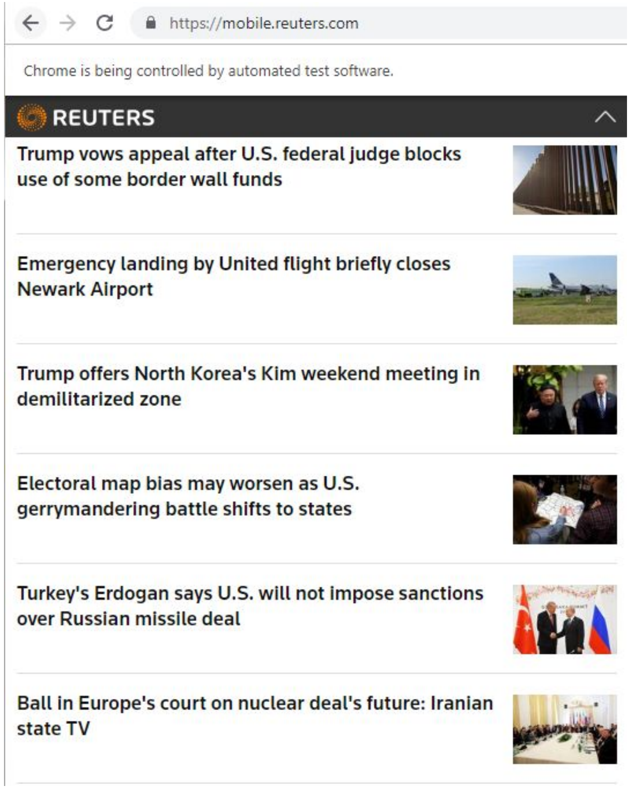
Step 1: Data Collection

Web Scraping Reuters

Using mobile version, much easier!....



```
In [57]: container = soup.find_all('div', class_= 'module-container')

In [25]: container[0].find_all('h2')

Out[25]: [<h2 class="top-story-heading">
          <a href="/article/idUSKCN1TT373" target="_self">Trump says China trade talks 'back on track'</a>
          </h2>]

In [58]: top_news = soup.find_all('section', class_="module top-news-module")

In [54]: articles = top_news[0].find_all('article')
         articles[0].find_all('h3')[0].a.text

Out[54]: 'Trump vows appeal after U.S. federal judge blocks use of some border wall funds'

In [56]: for article in articles:
             print(article.find_all('h3')[0].a.text)

         Trump vows appeal after U.S. federal judge blocks use of some border wall funds
         Emergency landing by United flight briefly closes Newark Airport
         Trump offers North Korea's Kim weekend meeting in demilitarized zone
         Electoral map bias may worsen as U.S. gerrymandering battle shifts to states
         Turkey's Erdogan says U.S. will not impose sanctions over Russian missile deal
         Ball in Europe's court on nuclear deal's future: Iranian state TV
         Wildfires and power cuts plague Europeans as heatwave breaks records
         Italian police arrest migrant-rescue ship captain after docking
         Ship carrying waste arrives back in Canada from the Philippines
         Deutsche Bank board to meet July 7 to decide on job cuts: sources
```



https://mobile.reuters.com

Chrome is being controlled by automated test software.

REUTERS

Trump vows appeal after U.S. federal judge blocks use of some border wall funds

Emergency landing by United flight briefly closes Newark Airport

Trump offers North Korea's Kim weekend meeting in demilitarized zone

Electoral map bias may worsen as U.S. gerrymandering battle shifts to states

Turkey's Erdogan says U.S. will not impose sanctions over Russian missile deal

Ball in Europe's court on nuclear deal's future: Iranian state TV

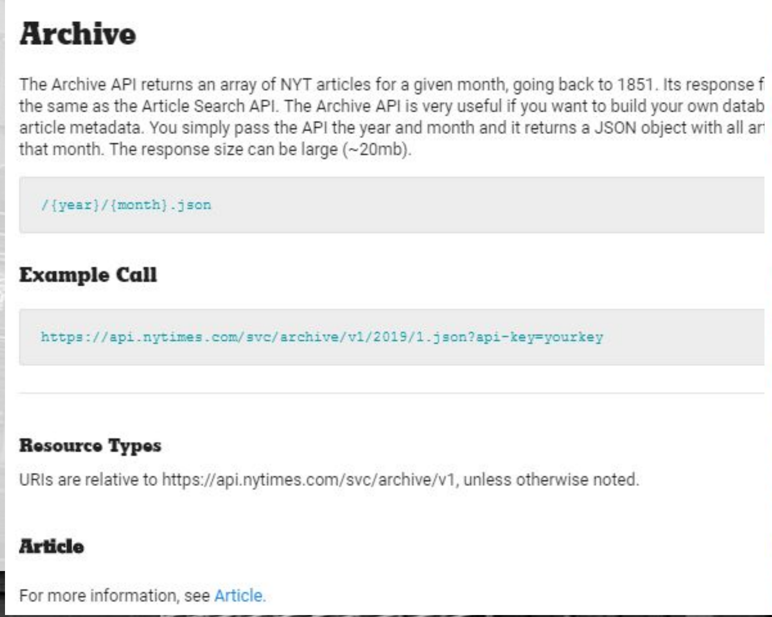
….but this only gives headlines from today =(

So to train our model we must move to LARGE data sets….

“Real News”: NY Times API, Reuters (from Kaggle)

Historical news articles archives. Able to set parameters for month and year.

Create for loop in Jupyter Notebook to collect all articles from Jan 2019 - current



Archive

The Archive API returns an array of NYT articles for a given month, going back to 1851. Its response f the same as the Article Search API. The Archive API is very useful if you want to build your own datab article metadata. You simply pass the API the year and month and it returns a JSON object with all ar that month. The response size can be large (~20mb).

```
/{year}/{month}.json
```

Example Call

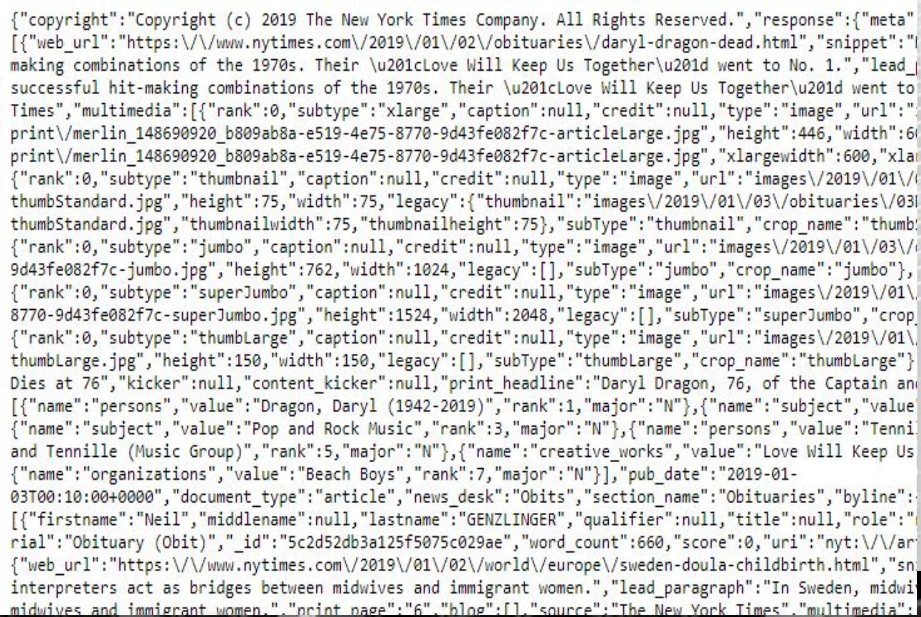
```
https://api.nytimes.com/svc/archive/v1/2019/1.json?api-key=yourkey
```

Resource Types

URIs are relative to https://api.nytimes.com/svc/archive/v1, unless otherwise noted.

Article

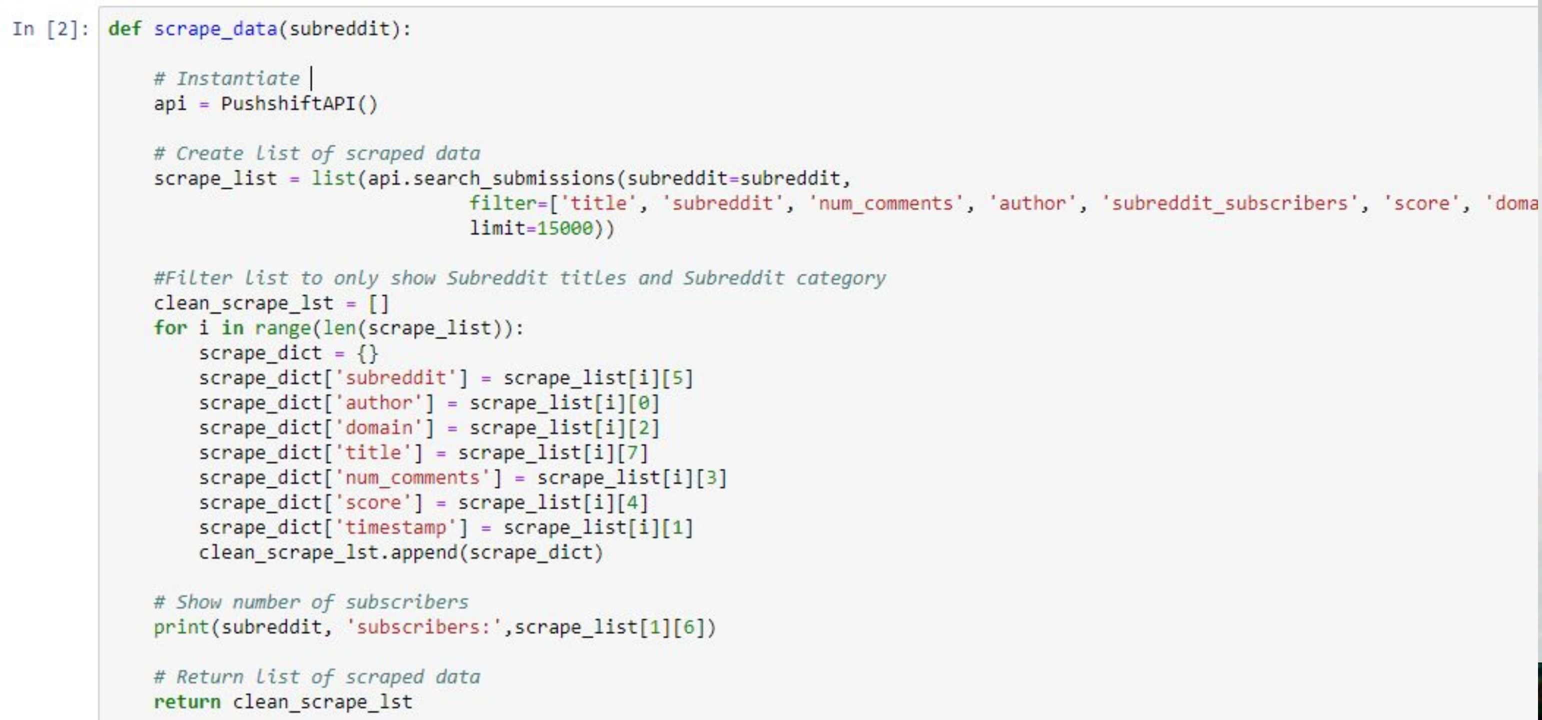
For more information, see Article.



```
{"copyright":"Copyright (c) 2019 The New York Times Company. All Rights Reserved.","response":{"meta"
[{"web_url":"https:\/\/www.nytimes.com\/2019\/01\/02\/obituaries\/daryl-dragon-dead.html","snippet":"
making combinations of the 1970s. Their \u201cLove Will Keep Us Together\u201d went to No. 1.","lead_
successful hit-making combinations of the 1970s. Their \u201cLove Will Keep Us Together\u201d went to
Times","multimedia":[{"rank":0,"subtype":"xlarge","caption":null,"credit":null,"type":"image","url":"
print\/merlin_148690920_b809ab8a-e519-4e75-8770-9d43fe082f7c-articleLarge.jpg","height":446,"width":6
print\/merlin_148690920_b809ab8a-e519-4e75-8770-9d43fe082f7c-articleLarge.jpg","xlargewidth":600,"xla
{"rank":0,"subtype":"thumbnail","caption":null,"credit":null,"type":"image","url":"images\/2019\/01\/
thumbStandard.jpg","height":75,"width":75,"legacy":{"thumbnail":"images\/2019\/01\/03\/obituaries\/03
thumbStandard.jpg","thumbnailwidth":75,"thumbnailheight":75},"subType":"thumbnail","crop_name":"thumb
{"rank":0,"subtype":"jumbo","caption":null,"credit":null,"type":"image","url":"images\/2019\/01\/03\/
9d43fe082f7c-jumbo.jpg","height":762,"width":1024,"legacy":[],"subType":"jumbo","crop_name":"jumbo"},
{"rank":0,"subtype":"superJumbo","caption":null,"credit":null,"type":"image","url":"images\/2019\/01\
8770-9d43fe082f7c-superJumbo.jpg","height":1524,"width":2048,"legacy":[],"subType":"superJumbo","crop
{"rank":0,"subtype":"thumbLarge","caption":null,"credit":null,"type":"image","url":"images\/2019\/01\
thumbLarge.jpg","height":150,"width":150,"legacy":[],"subType":"thumbLarge","crop_name":"thumbLarge"}
Dies at 76","kicker":null,"content_kicker":null,"print_headline":"Daryl Dragon, 76, of the Captain an
[{"name":"persons","value":"Dragon, Daryl (1942-2019)","rank":1,"major":"N"},{"name":"subject","value
{"name":"subject","value":"Pop and Rock Music","rank":3,"major":"N"},{"name":"persons","value":"Tenni
and Tennille (Music Group)","rank":5,"major":"N"},{"name":"creative_works","value":"Love Will Keep Us
{"name":"organizations","value":"Beach Boys","rank":7,"major":"N"}],"pub_date":"2019-01-
03T00:10:00+0000","document_type":"article","news_desk":"Obits","section_name":"Obituaries","byline":
[{"firstname":"Neil","middlename":null,"lastname":"GENZLINGER","qualifier":null,"title":null,"role":"
rial":"Obituary (Obit)","_id":"5c2d52db3a125f5075c029ae","word_count":660,"score":0,"uri":"nyt:\/\/ar
{"web_url":"https:\/\/www.nytimes.com\/2019\/01\/02\/world\/europe\/sweden-doula-childbirth.html","sn
interpreters act as bridges between midwives and immigrant women.","lead_paragraph":"In Sweden, midwi
midwives and immigrant women.","print_page":"6","blog":[],"source":"The New York Times","multimedia":
```

“Fake News”: r/The Onion API, Kaggle (many sources)

Using PSAW - A wrapper for searching public Reddit comments via pushift.io API



```
In [2]: def scrape_data(subreddit):

            # Instantiate
            api = PushshiftAPI()

            # Create list of scraped data
            scrape_list = list(api.search_submissions(subreddit=subreddit,
                                        filter=['title', 'subreddit', 'num_comments', 'author', 'subreddit_subscribers', 'score', 'doma
                                        limit=15000))

            #Filter list to only show Subreddit titles and Subreddit category
            clean_scrape_lst = []
            for i in range(len(scrape_list)):
                scrape_dict = {}
                scrape_dict['subreddit'] = scrape_list[i][5]
                scrape_dict['author'] = scrape_list[i][0]
                scrape_dict['domain'] = scrape_list[i][2]
                scrape_dict['title'] = scrape_list[i][7]
                scrape_dict['num_comments'] = scrape_list[i][3]
                scrape_dict['score'] = scrape_list[i][4]
                scrape_dict['timestamp'] = scrape_list[i][1]
                clean_scrape_lst.append(scrape_dict)

            # Show number of subscribers
            print(subreddit, 'subscribers:',scrape_list[1][6])

            # Return list of scraped data
            return clean_scrape_lst
```

Final Data:

REAL NEWS:

- New York Times
  - Jan & Feb 2016
  - Removed opinions and blogs
  - ~13,500 records
- Reuters
  - 2016 through 2018
  - Removed incorrect headlines
  - ~40,000 records

TOTAL HEADLINES: 39,790
58 % Real, 42% Fake

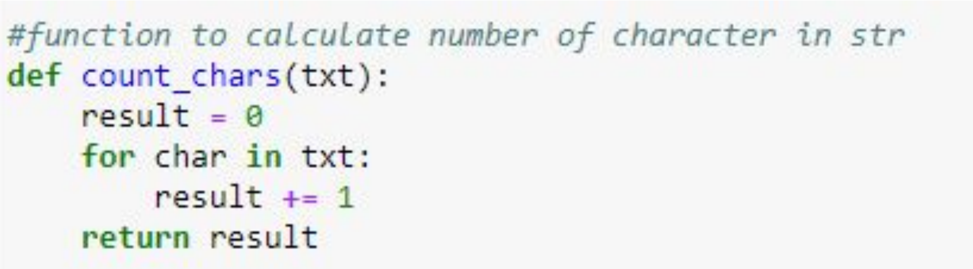
FAKE NEWS:

- r/TheOnion
  - 2018 - 2019
  - Removed fortnite ads, duplicates
  - ~6,300 records
  - Removed headlines < 2 words (not actual headlines but user post subject title - potential issue with larger titles)
- Kaggle (many sources)
  - Removed incorrect headlines/rows that weren’t delimited accurately
  - ~10,900

Step 2: Headline Metric Functions

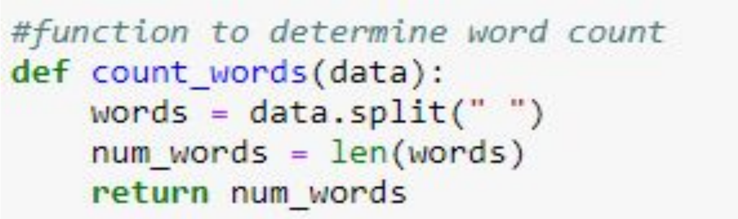
Create functions to analyze headlines (python)

Character count



```python
#function to calculate number of character in str
def count_chars(txt):
    result = 0
    for char in txt:
        result += 1
    return result
```

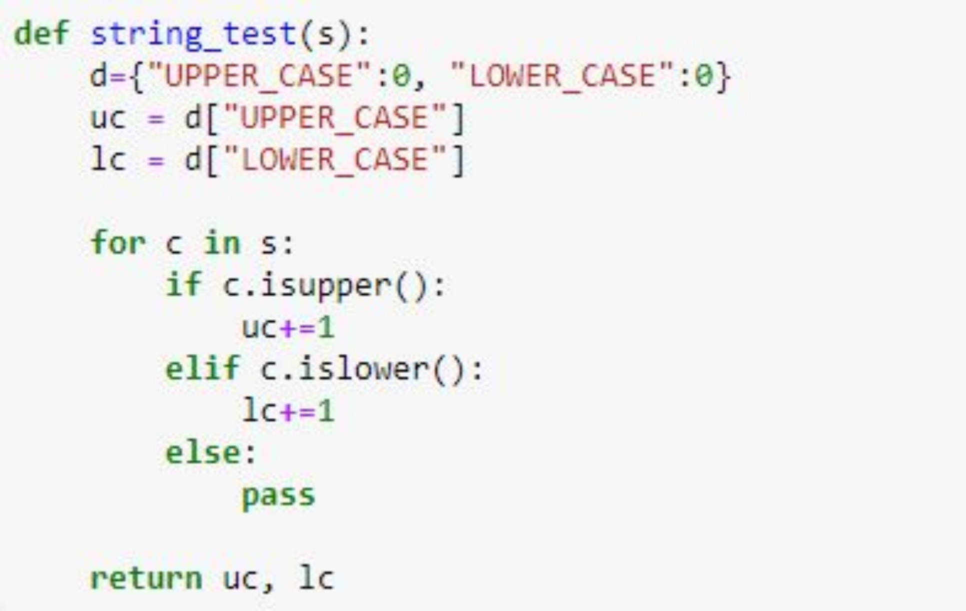
Word count (with .split)



```python
#function to determine word count
def count_words(data):
    words = data.split(" ")
    num_words = len(words)
    return num_words
```

Create functions to analyze headlines (python)

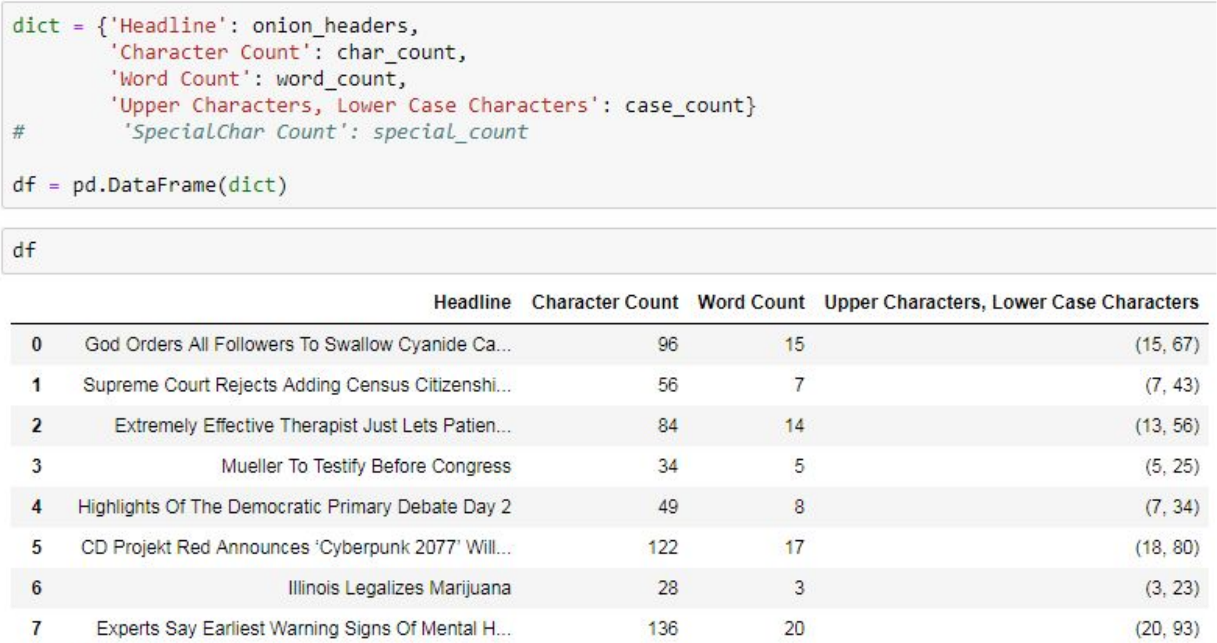
Case count (using .isupper(), .islower())



```python
def string_test(s):
    d={"UPPER_CASE":0, "LOWER_CASE":0}
    uc = d["UPPER_CASE"]
    lc = d["LOWER_CASE"]

    for c in s:
        if c.isupper():
            uc+=1
        elif c.islower():
            lc+=1
        else:
            pass

    return uc, lc
```

Create functions to analyze headlines (python)

Output Dataframe, to .csv



```
dict = {'Headline': onion_headers,
        'Character Count': char_count,
        'Word Count': word_count,
        'Upper Characters, Lower Case Characters': case_count}
#         'SpecialChar Count': special_count

df = pd.DataFrame(dict)
```

```
df
```

	Headline	Character Count	Word Count	Upper Characters, Lower Case Characters
0	God Orders All Followers To Swallow Cyanide Ca...	96	15	(15, 67)
1	Supreme Court Rejects Adding Census Citizenshi...	56	7	(7, 43)
2	Extremely Effective Therapist Just Lets Patien...	84	14	(13, 56)
3	Mueller To Testify Before Congress	34	5	(5, 25)
4	Highlights Of The Democratic Primary Debate Day 2	49	8	(7, 34)
5	CD Projekt Red Announces ‘Cyberpunk 2077’ Will...	122	17	(18, 80)
6	Illinois Legalizes Marijuana	28	3	(3, 23)
7	Experts Say Earliest Warning Signs Of Mental H...	136	20	(20, 93)

Step 3: Machine Learning-Train/Test

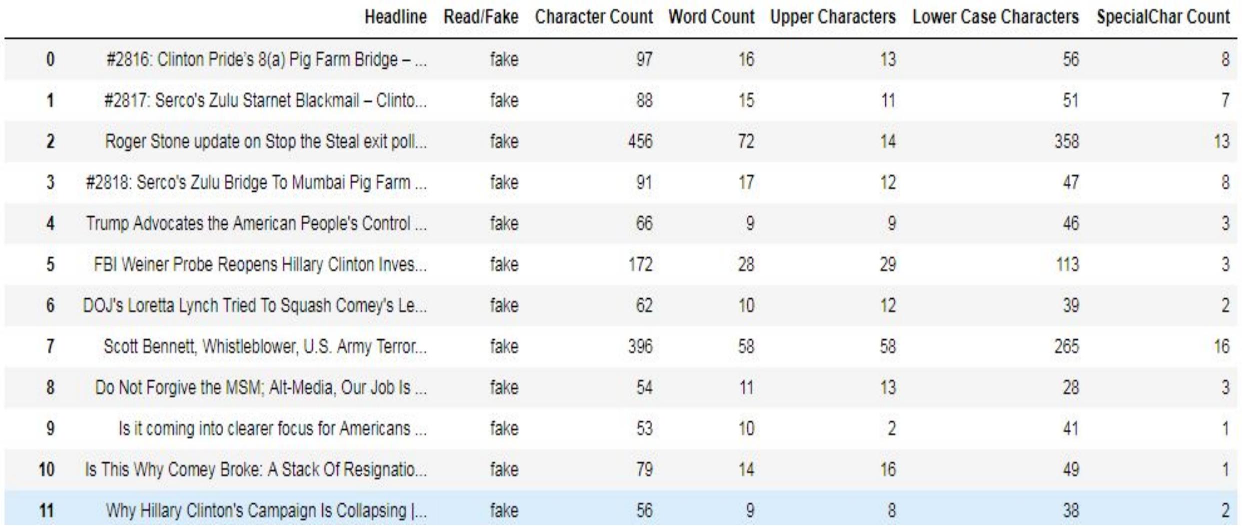
1) Logistic Regression on built on Manual Metrics

Columns:

- Character count
- Word Count
- Upper Case Characters
- Lower Case Characters
- Special Character Count
- Sentiment Analysis

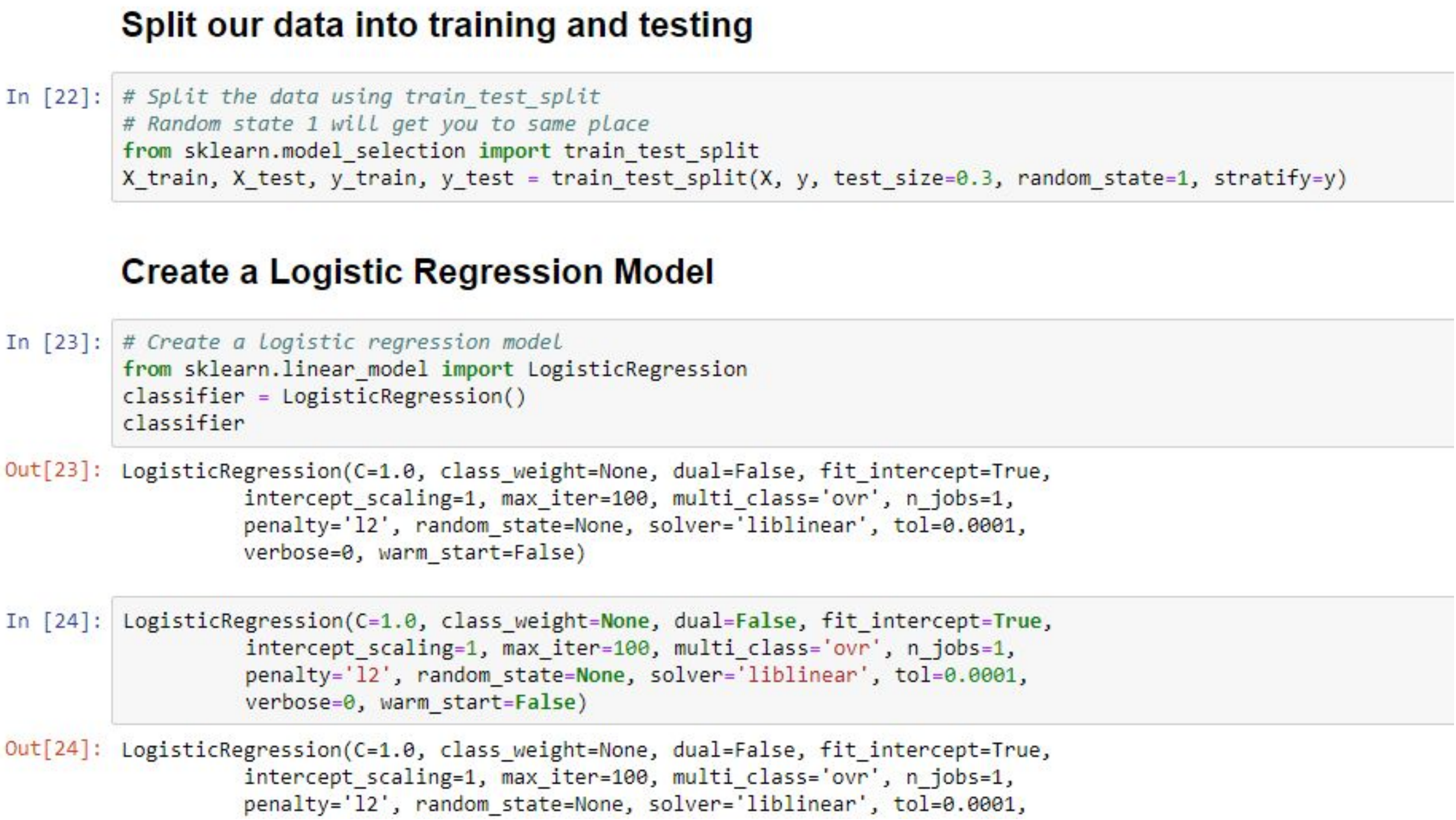
Removed “Headlines” column

X = dropped the Real/Fake column



	Headline	Read/Fake	Character Count	Word Count	Upper Characters	Lower Case Characters	SpecialChar Count
0	#2816: Clinton Pride's 8(a) Pig Farm Bridge – ...	fake	97	16	13	56	8
1	#2817: Serco's Zulu Starnet Blackmail – Clinto...	fake	88	15	11	51	7
2	Roger Stone update on Stop the Steal exit poll...	fake	456	72	14	358	13
3	#2818: Serco's Zulu Bridge To Mumbai Pig Farm ...	fake	91	17	12	47	8
4	Trump Advocates the American People's Control ...	fake	66	9	9	46	3
5	FBI Weiner Probe Reopens Hillary Clinton Inves...	fake	172	28	29	113	3
6	DOJ's Loretta Lynch Tried To Squash Comey's Le...	fake	62	10	12	39	2
7	Scott Bennett, Whistleblower, U.S. Army Terror...	fake	396	58	58	265	16
8	Do Not Forgive the MSM; Alt-Media, Our Job Is ...	fake	54	11	13	28	3
9	Is it coming into clearer focus for Americans ...	fake	53	10	2	41	1
10	Is This Why Comey Broke: A Stack Of Resignatio...	fake	79	14	16	49	1
11	Why Hillary Clinton's Campaign Is Collapsing |...	fake	56	9	8	38	2

1) Logistic Regression on built on Manual Metrics



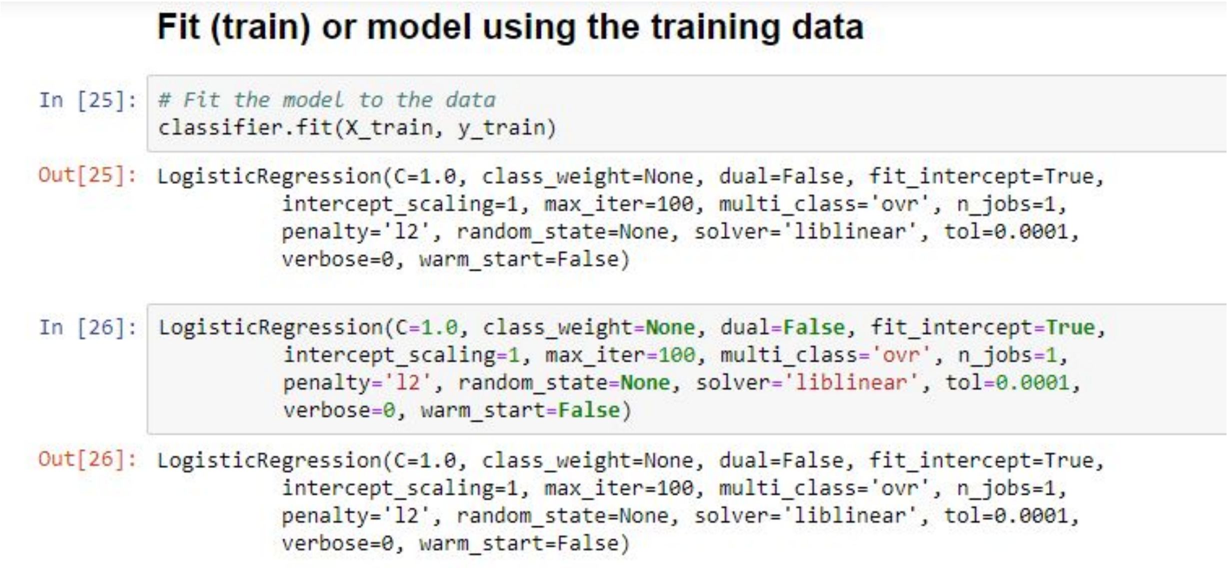
Split our data into training and testing

```
In [22]: # Split the data using train_test_split
         # Random state 1 will get you to same place
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
```

Create a Logistic Regression Model

```
In [23]: # Create a logistic regression model
         from sklearn.linear_model import LogisticRegression
         classifier = LogisticRegression()
         classifier

Out[23]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)

In [24]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)

Out[24]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
```

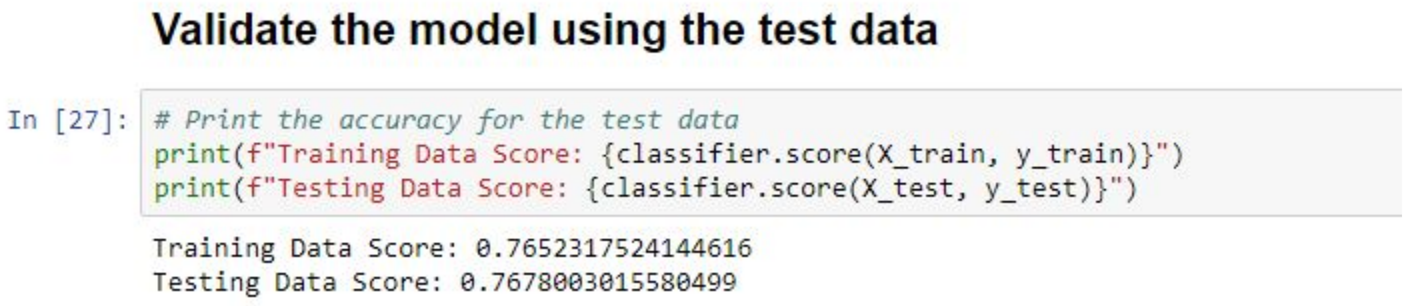
1) Logistic Regression on built on Manual Metrics



Fit (train) or model using the training data

```
In [25]: # Fit the model to the data
         classifier.fit(X_train, y_train)

Out[25]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)

In [26]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)

Out[26]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)
```

1) Logistic Regression on built on Manual Metrics



Validate the model using the test data

```
In [27]: # Print the accuracy for the test data
         print(f"Training Data Score: {classifier.score(X_train, y_train)}")
         print(f"Testing Data Score: {classifier.score(X_test, y_test)}")
```

```
Training Data Score: 0.7652317524144616
Testing Data Score: 0.7678003015580499
```

Training Data Score: 0.7652317524144616
Testing Data Score: 0.7678003015580499

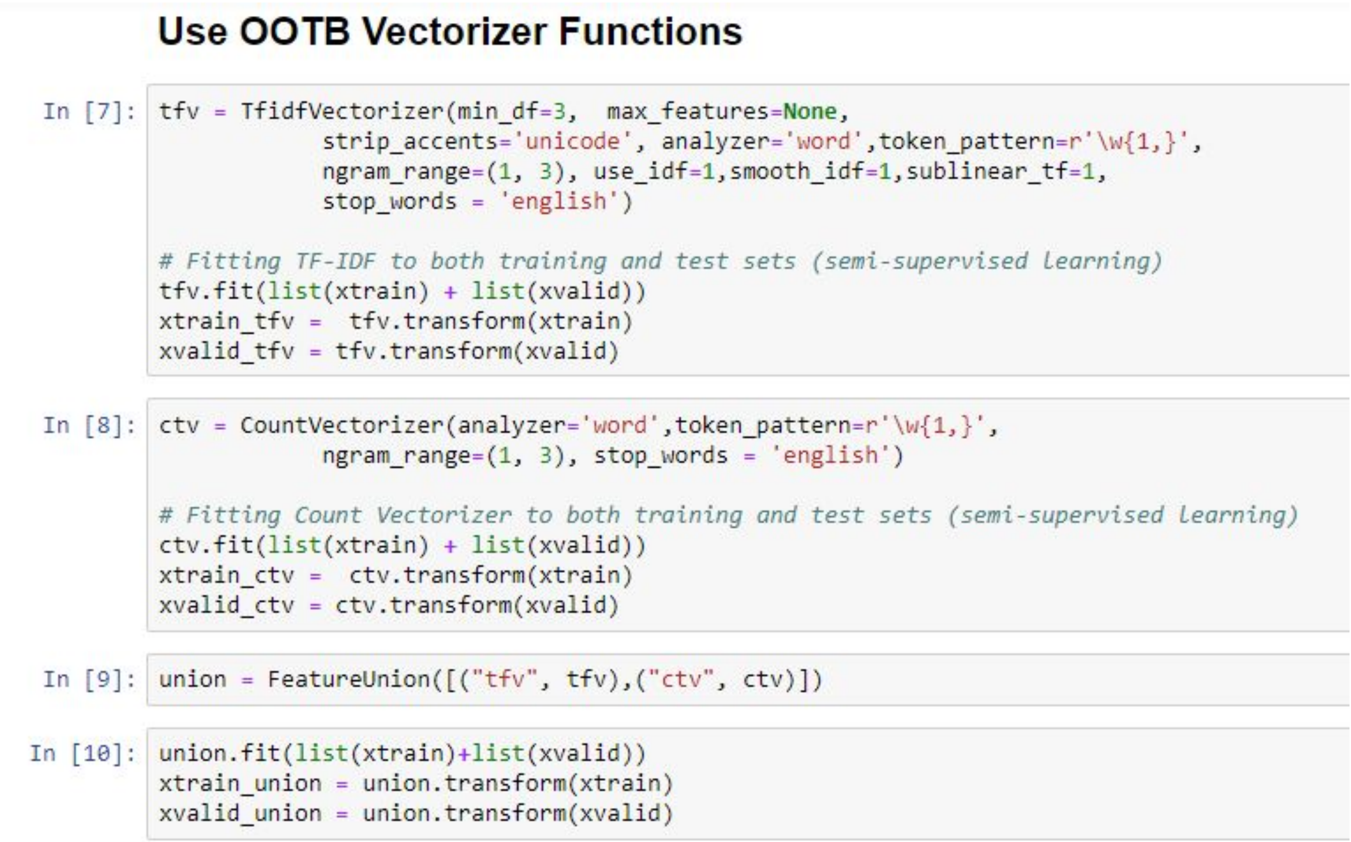
"...But we can do better!"

2) ML with SKLearn TF-IDF Vectorizer / Models

Tools Used:

- Term Frequency/ Inverse Document Frequency Vectorizer
- Logistic Regression
- Naive Bayes Model
- Random Forest
- XG Boost

2) ML with SKLearn TF-IDF Vectorizers



Use OOTB Vectorizer Functions

In [7]:

```
tfv = TfidfVectorizer(min_df=3,  max_features=None,
            strip_accents='unicode', analyzer='word',token_pattern=r'\w{1,}',
            ngram_range=(1, 3), use_idf=1,smooth_idf=1,sublinear_tf=1,
            stop_words = 'english')

# Fitting TF-IDF to both training and test sets (semi-supervised learning)
tfv.fit(list(xtrain) + list(xvalid))
xtrain_tfv =  tfv.transform(xtrain)
xvalid_tfv = tfv.transform(xvalid)
```

In [8]:

```
ctv = CountVectorizer(analyzer='word',token_pattern=r'\w{1,}',
            ngram_range=(1, 3), stop_words = 'english')

# Fitting Count Vectorizer to both training and test sets (semi-supervised learning)
ctv.fit(list(xtrain) + list(xvalid))
xtrain_ctv =  ctv.transform(xtrain)
xvalid_ctv = ctv.transform(xvalid)
```

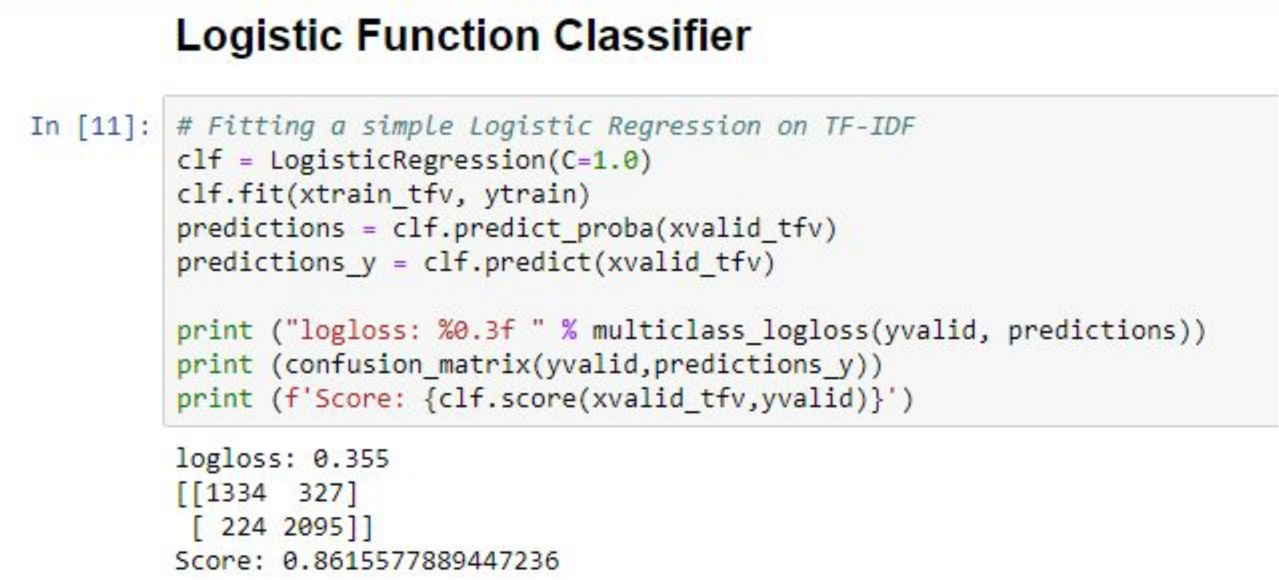
In [9]:

```
union = FeatureUnion([("tfv", tfv),("ctv", ctv)])
```

In [10]:

```
union.fit(list(xtrain)+list(xvalid))
xtrain_union = union.transform(xtrain)
xvalid_union = union.transform(xvalid)
```

2) ML with SKLearn TF-IDF Vectorizers



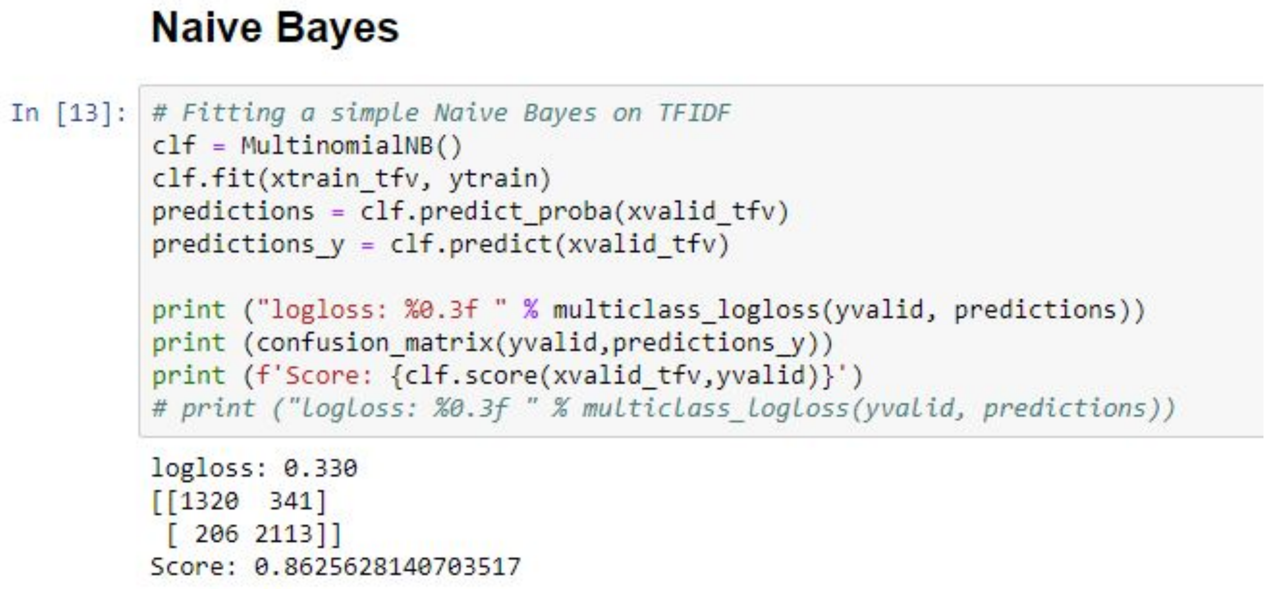
Logistic Function Classifier

In [11]:
```python
# Fitting a simple Logistic Regression on TF-IDF
clf = LogisticRegression(C=1.0)
clf.fit(xtrain_tfv, ytrain)
predictions = clf.predict_proba(xvalid_tfv)
predictions_y = clf.predict(xvalid_tfv)

print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
print (confusion_matrix(yvalid,predictions_y))
print (f'Score: {clf.score(xvalid_tfv,yvalid)}')
```

```
logloss: 0.355
[[1334  327]
 [ 224 2095]]
Score: 0.8615577889447236
```

LogLoss: 0.355
Score: 0.8615577889447236

2) ML with SKLearn TF-IDF Vectorizers



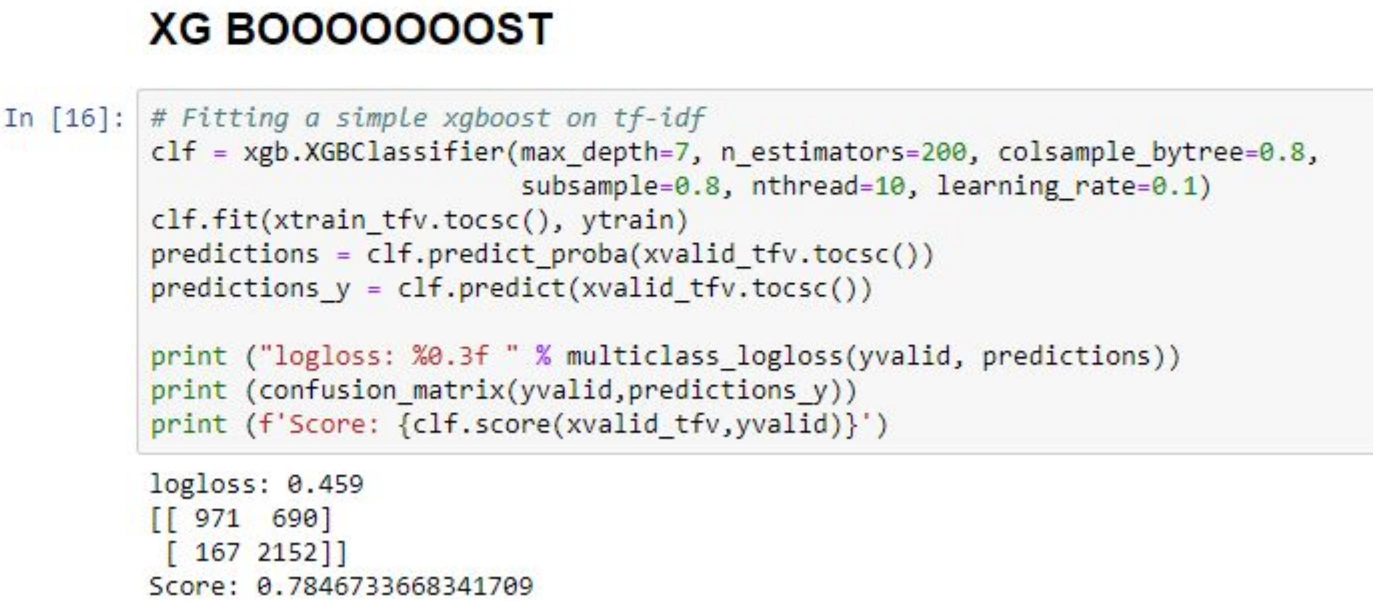
Naive Bayes

```
In [13]: # Fitting a simple Naive Bayes on TFIDF
         clf = MultinomialNB()
         clf.fit(xtrain_tfv, ytrain)
         predictions = clf.predict_proba(xvalid_tfv)
         predictions_y = clf.predict(xvalid_tfv)

         print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
         print (confusion_matrix(yvalid,predictions_y))
         print (f'Score: {clf.score(xvalid_tfv,yvalid)}')
         # print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
```

```
logloss: 0.330
[[1320  341]
 [ 206 2113]]
Score: 0.8625628140703517
```

LogLoss: 0.330
Score: 0.8625628140703517

2) ML with SKLearn TF-IDF Vectorizers



XG BOOOOOOOST

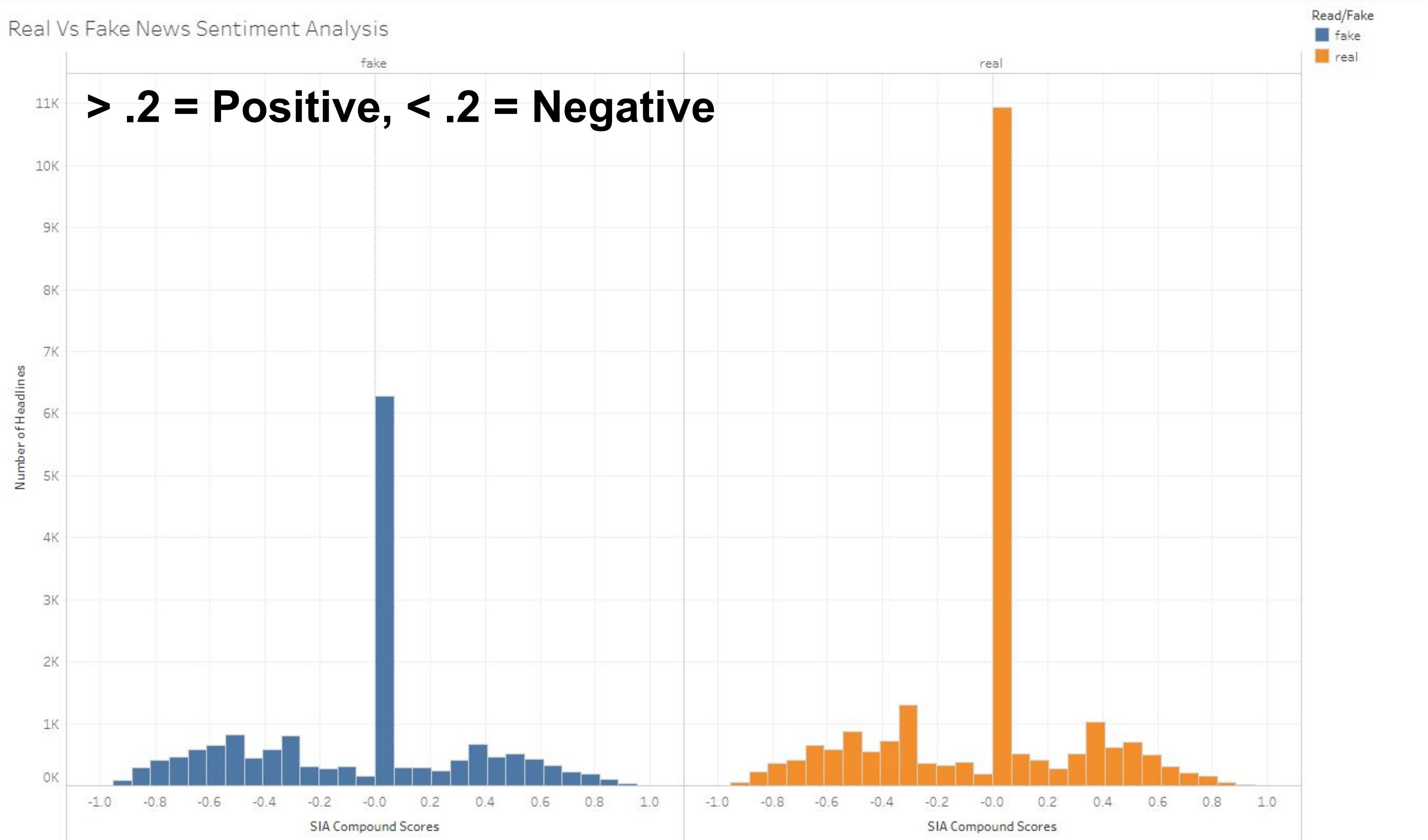
```
In [16]: # Fitting a simple xgboost on tf-idf
         clf = xgb.XGBClassifier(max_depth=7, n_estimators=200, colsample_bytree=0.8,
                                 subsample=0.8, nthread=10, learning_rate=0.1)
         clf.fit(xtrain_tfv.tocsc(), ytrain)
         predictions = clf.predict_proba(xvalid_tfv.tocsc())
         predictions_y = clf.predict(xvalid_tfv.tocsc())

         print ("logloss: %0.3f " % multiclass_logloss(yvalid, predictions))
         print (confusion_matrix(yvalid,predictions_y))
         print (f'Score: {clf.score(xvalid_tfv,yvalid)}')
```

```
logloss: 0.459
[[ 971  690]
 [ 167 2152]]
Score: 0.7846733668341709
```

LogLoss: 0.459
Score: 0.7846733668341709

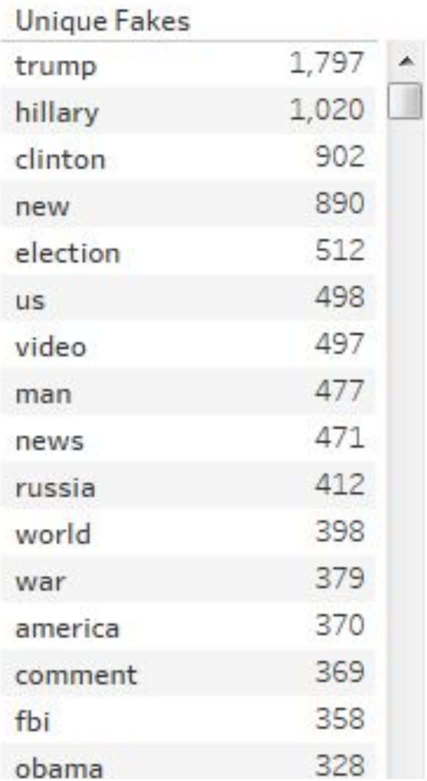
Step 4: Further Analysis

Tableau - Sentiment Analysis

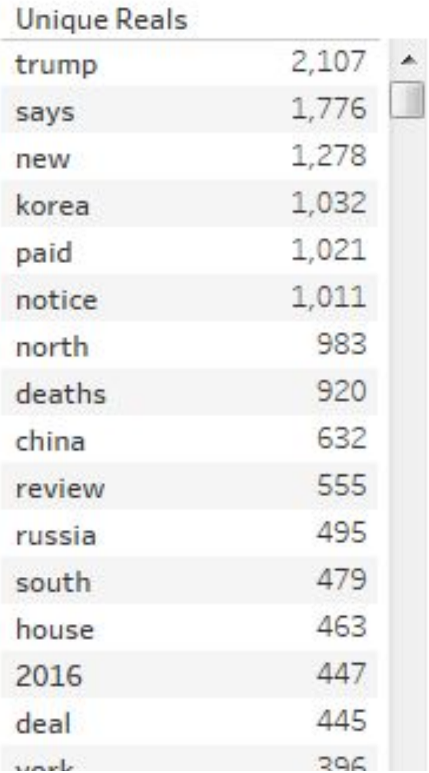


> .2 = Positive, < .2 = Negative

Tableau - Word Count



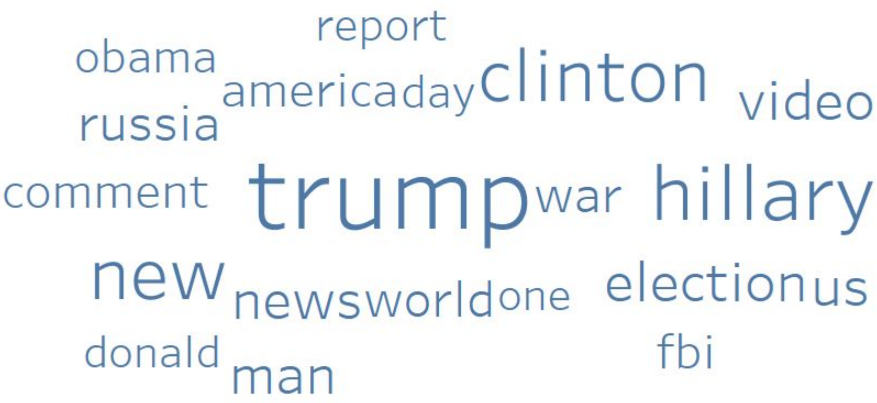
Unique Fakes	
trump	1,797
hillary	1,020
clinton	902
new	890
election	512
us	498
video	497
man	477
news	471
russia	412
world	398
war	379
america	370
comment	369
fbi	358
obama	328



Unique Reals	
trump	2,107
says	1,776
new	1,278
korea	1,032
paid	1,021
notice	1,011
north	983
deaths	920
china	632
review	555
russia	495
south	479
house	463
2016	447
deal	445
york	396

Tableau - WordCloud

FAKE_cloud



REAL_cloud

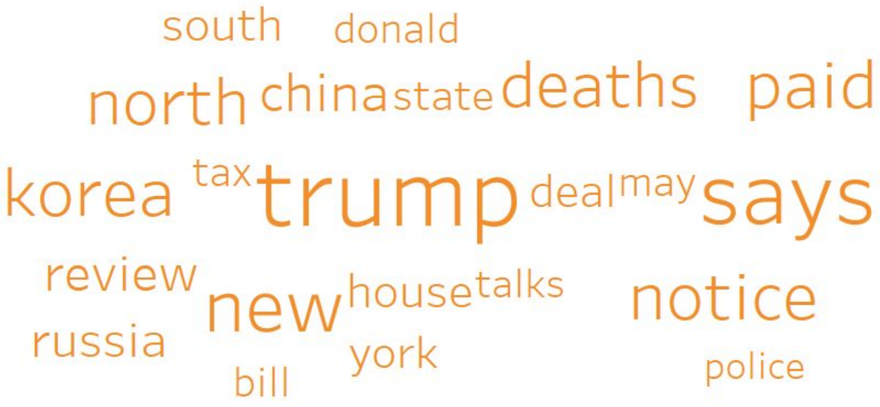
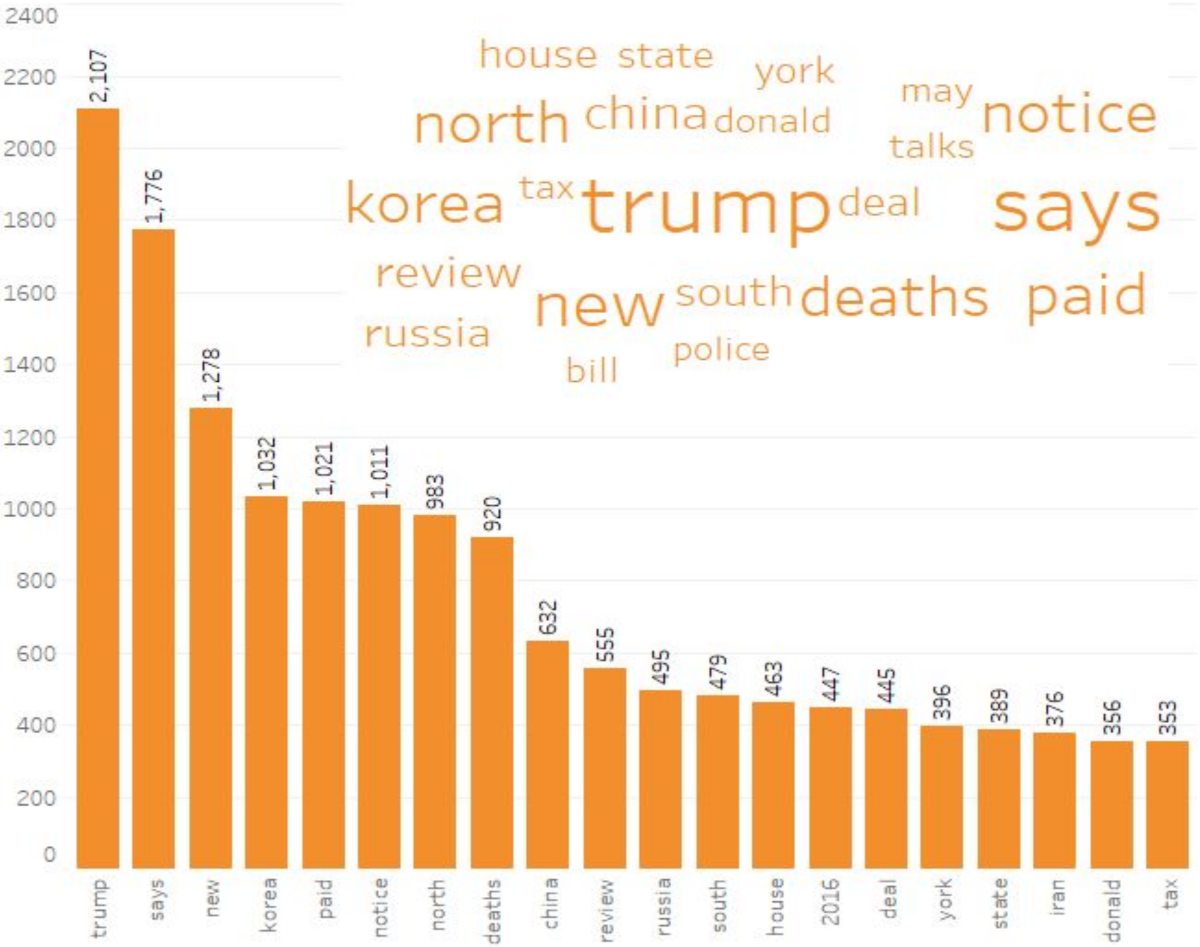
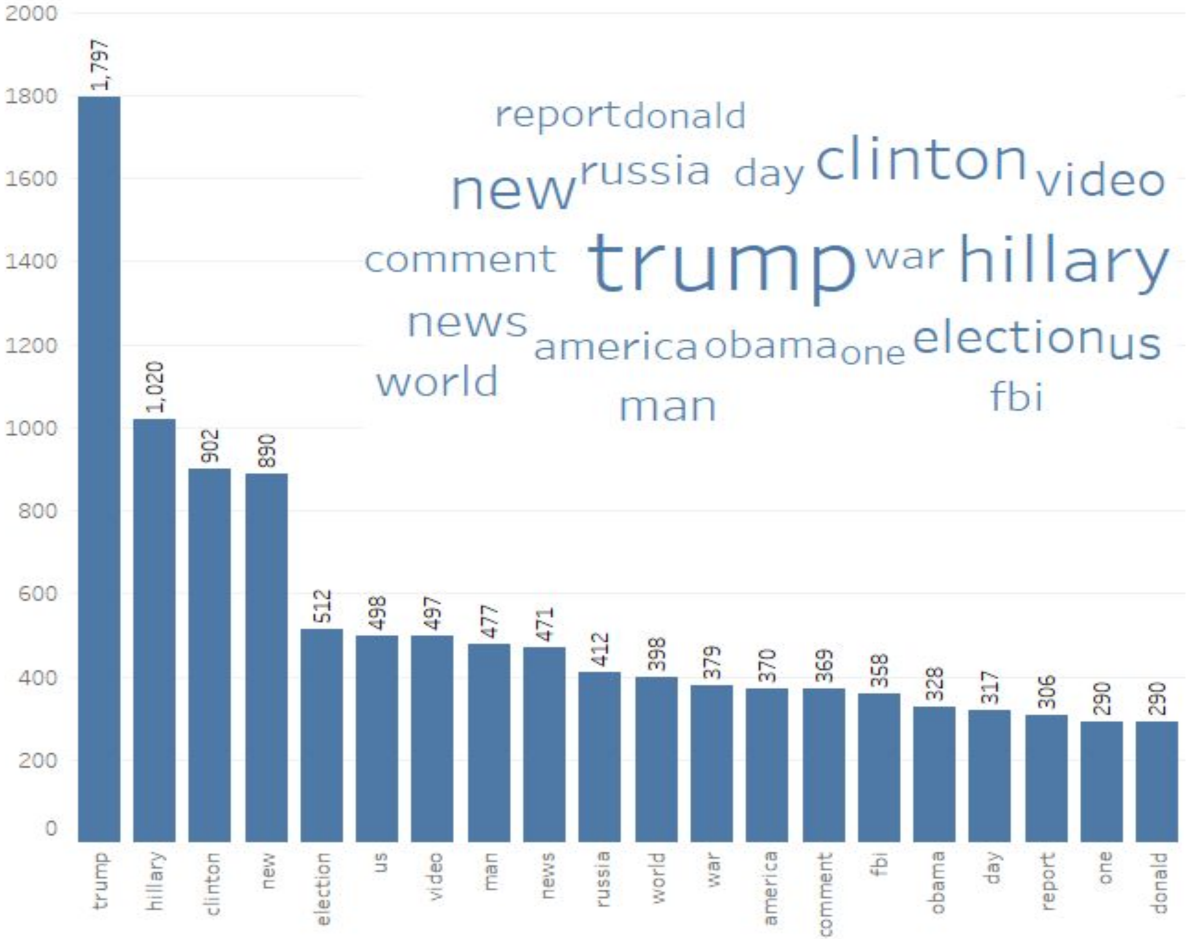


Tableau - Dashboard

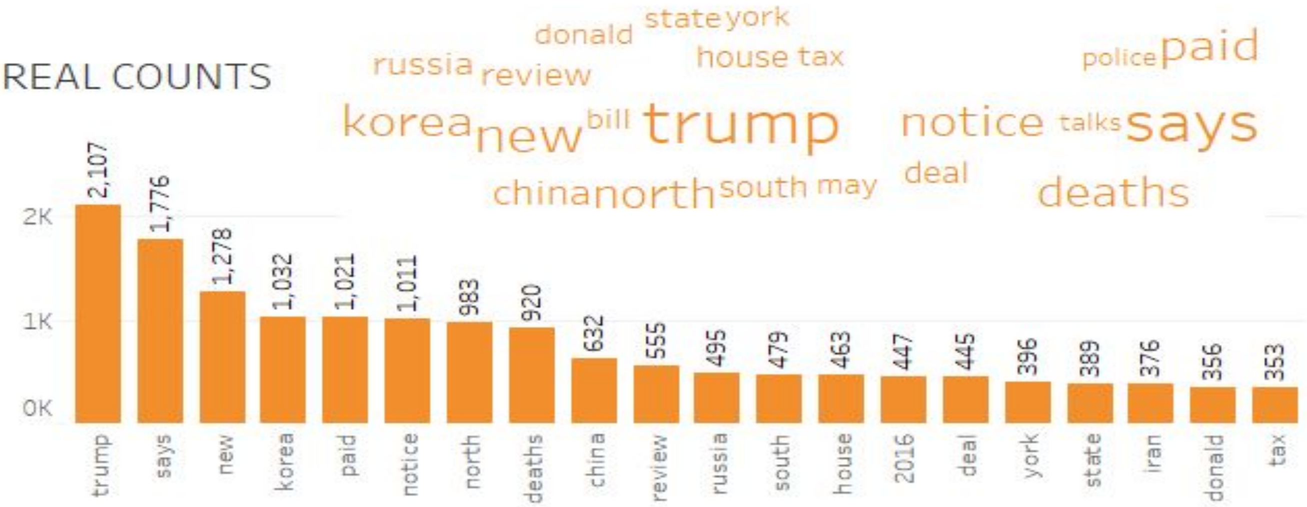
REAL COUNTS


2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
trump 2,107
says 1,776
new 1,278
korea 1,032
paid 1,021
notice 1,011
north 983
deaths 920
china 632
review 555
russia 495
south 479
house 463
2016 447
deal 445
york 396
state 389
iran 376
donald 356
tax 353
house
state
york
may
north
china
donald
notice
talks
korea
tax
trump
deal
says
review
new
south
deaths
paid
russia
bill
police

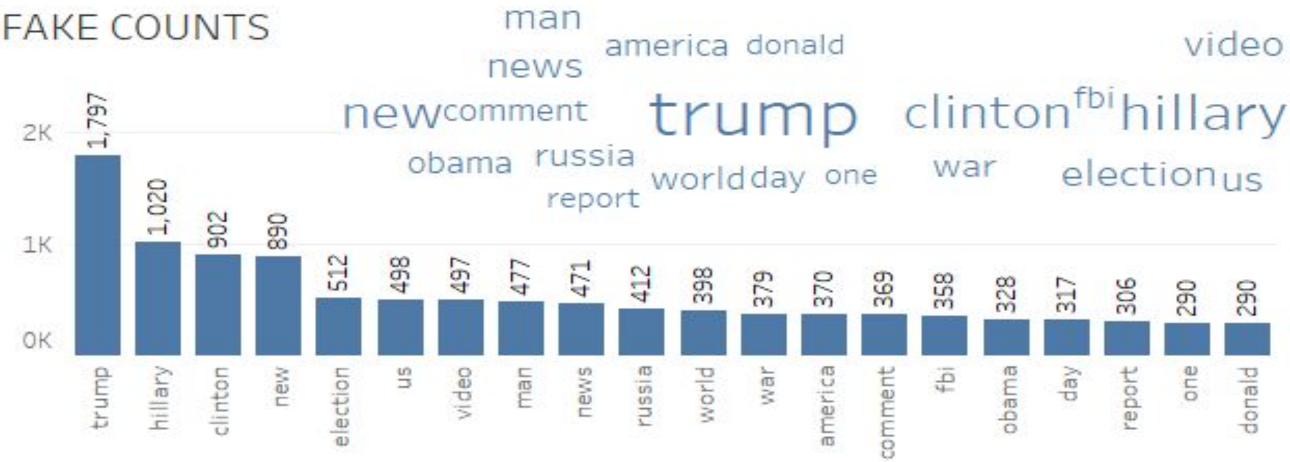
FAKE COUNTS


2000
1800
1600
1400
1200
1000
800
600
400
200
0
1,797
1,020
902
890
512
498
497
477
471
412
398
379
370
369
358
328
317
306
290
290
trump
hillary
clinton
new
election
us
video
man
news
russia
world
war
america
comment
fbi
obama
day
report
one
donald
report
donald
new
russia
day
clinton
video
comment
trump
war
hillary
news
america
obama
one
election
us
world
man
fbi

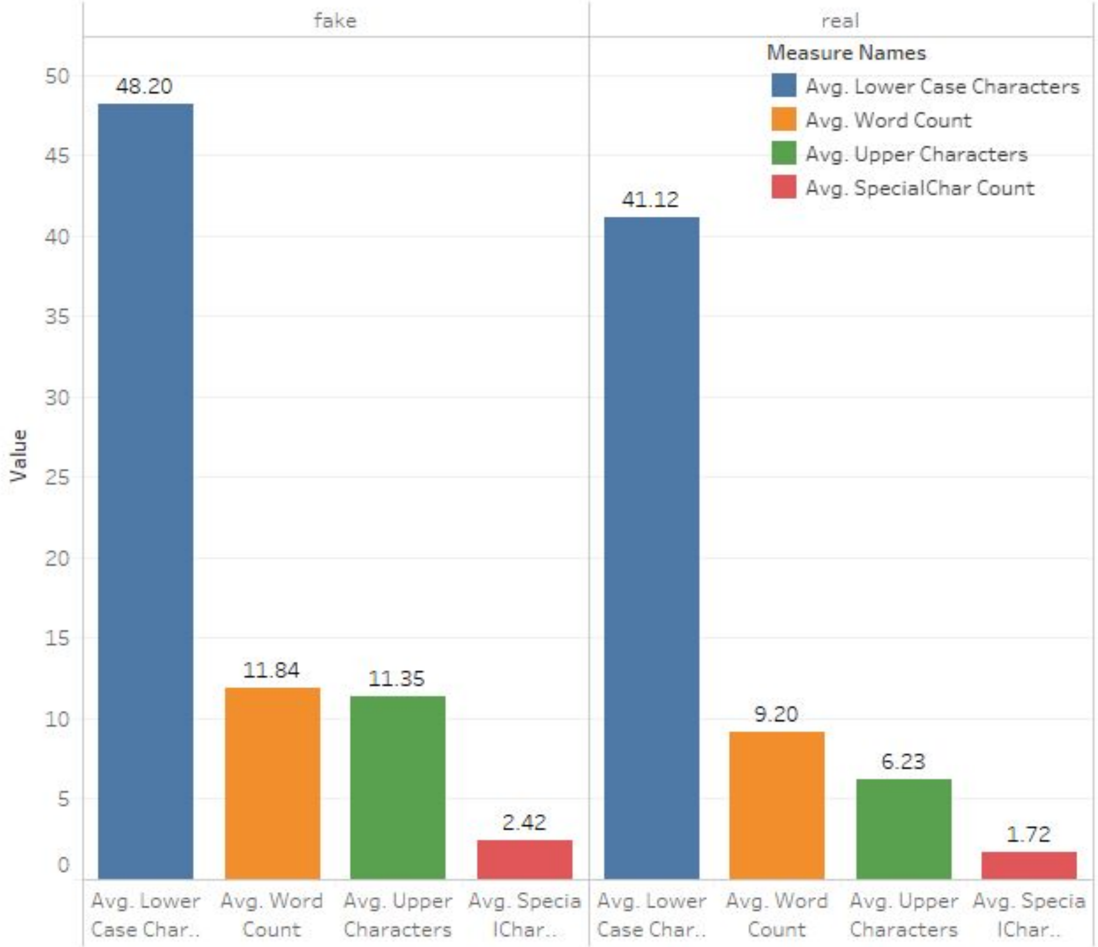
REAL COUNTS



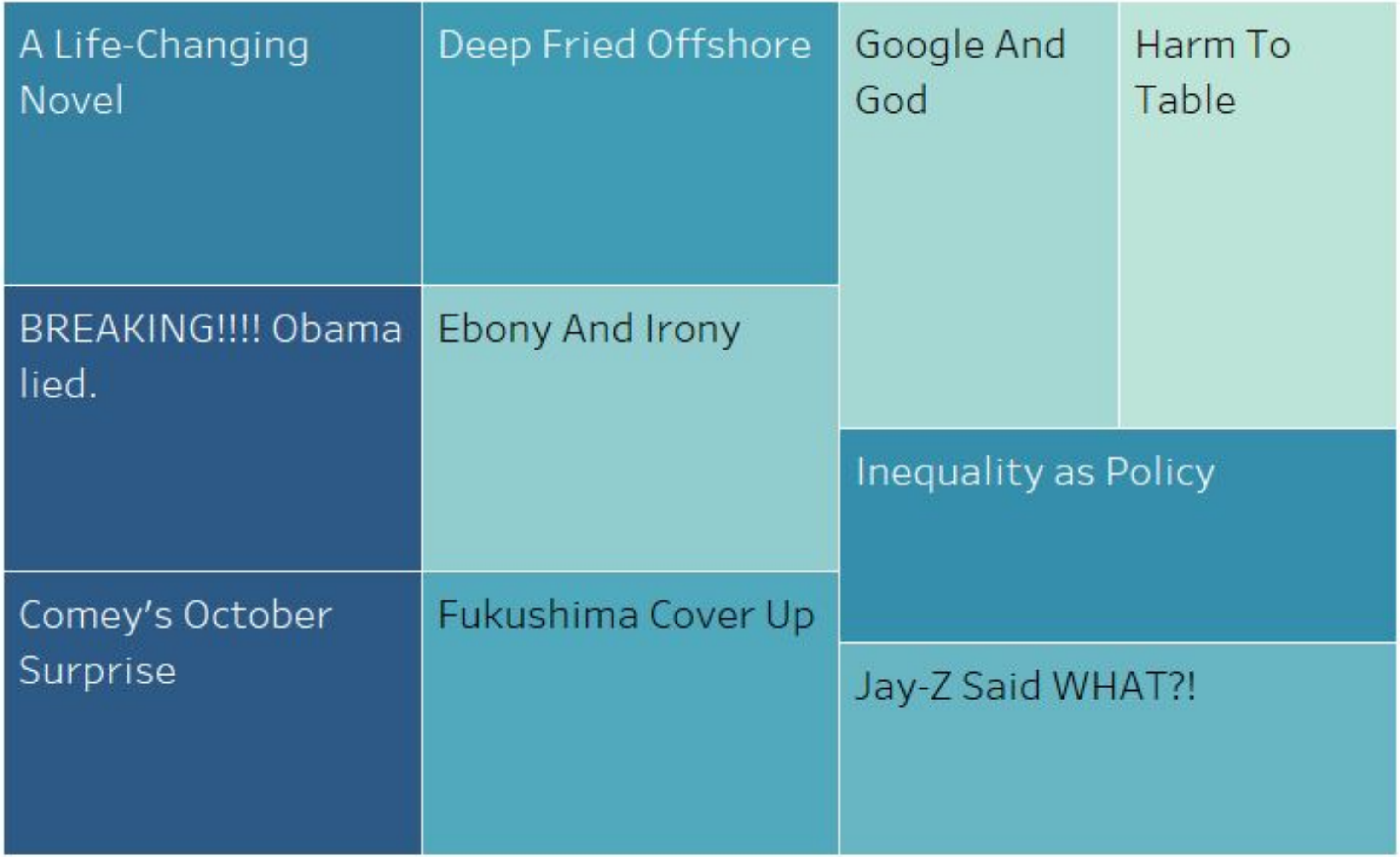
FAKE COUNTS



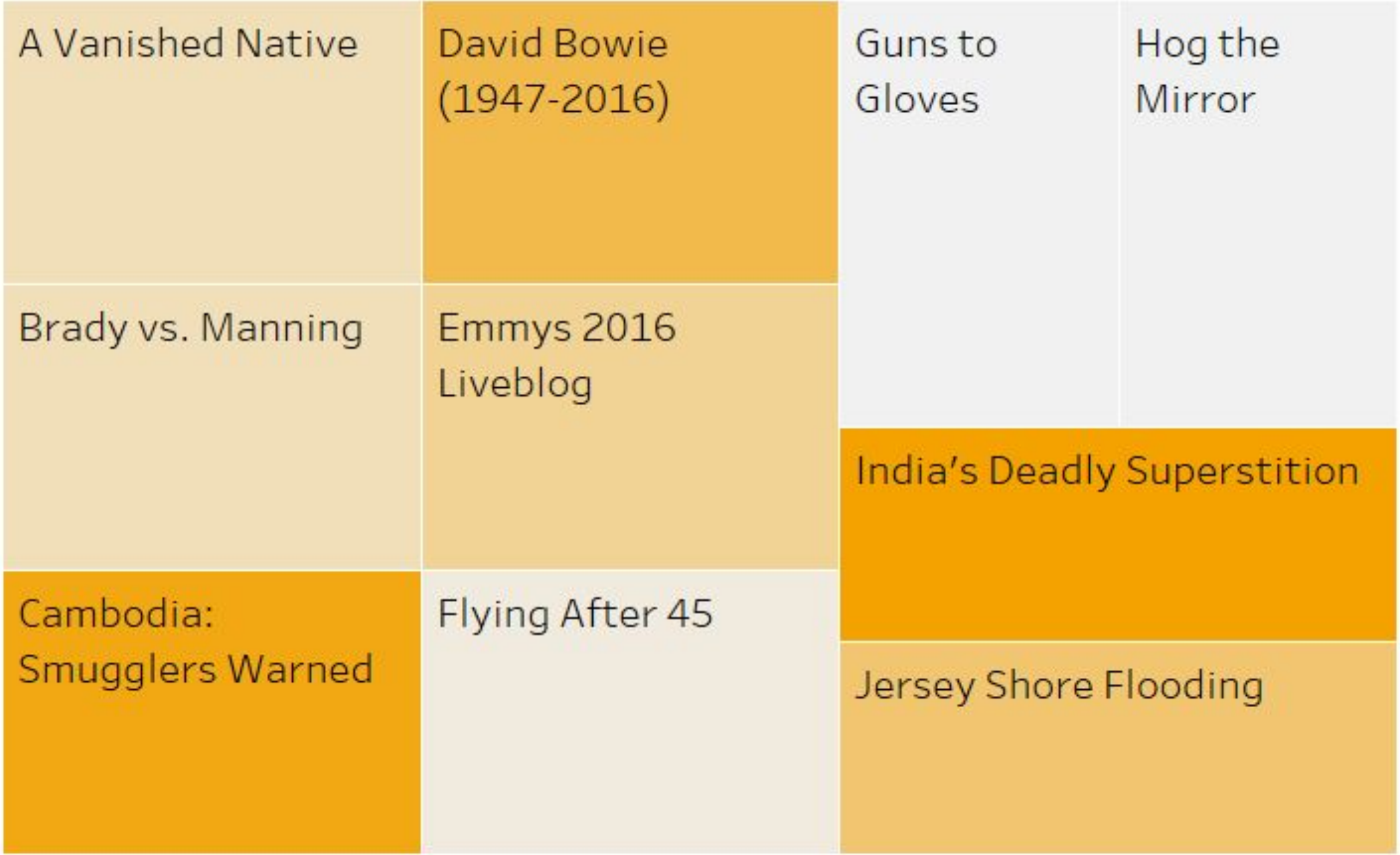
Character_Counts



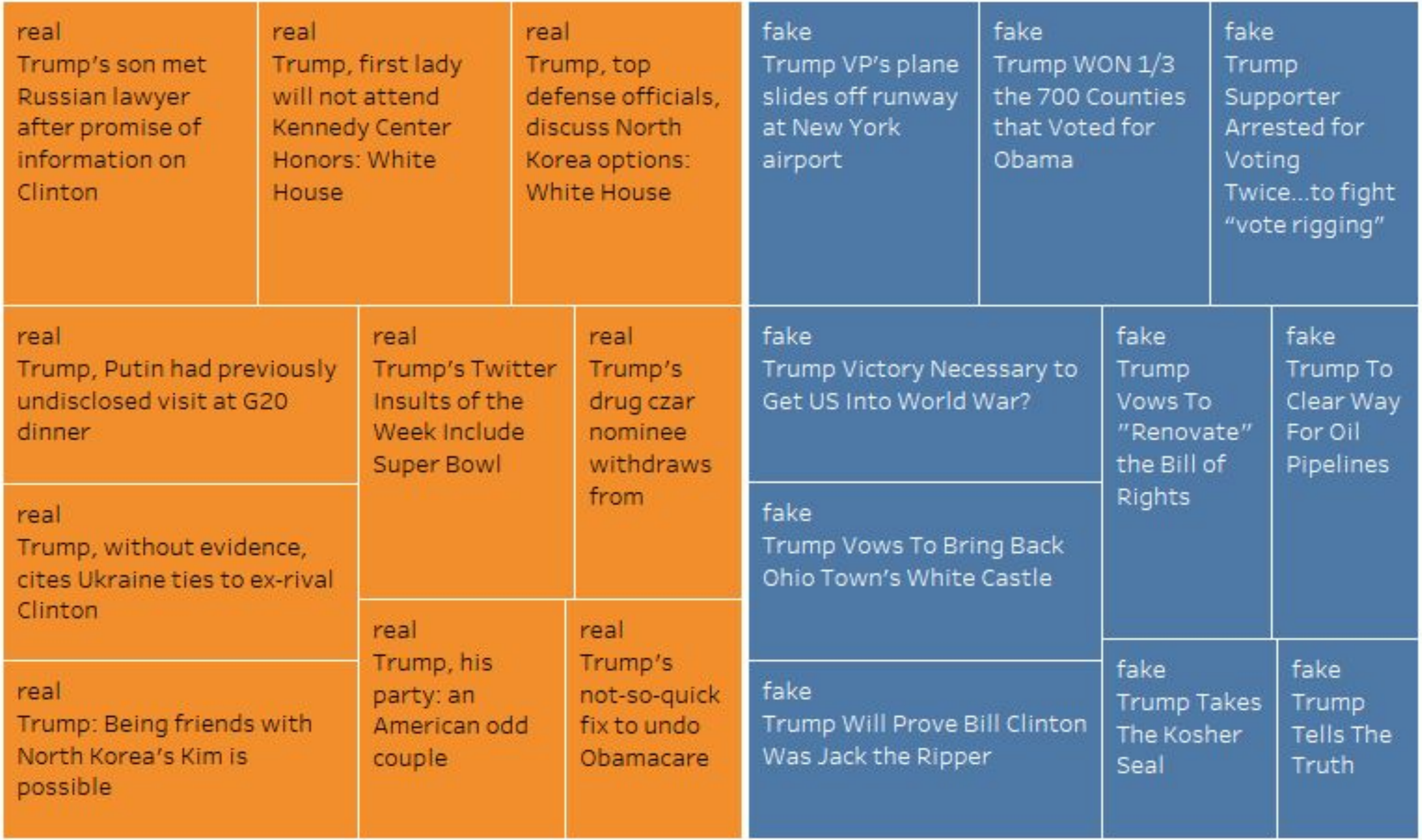
FAKE 3-Word Headlines



REAL 3 Word Headlines



Real vs Fake Trump Headlines



real
Trump's son met Russian lawyer after promise of information on Clinton

real
Trump, first lady will not attend Kennedy Center Honors: White House

real
Trump, top defense officials, discuss North Korea options: White House

real
Trump, Putin had previously undisclosed visit at G20 dinner

real
Trump's Twitter Insults of the Week Include Super Bowl

real
Trump's drug czar nominee withdraws from

real
Trump, without evidence, cites Ukraine ties to ex-rival Clinton

real
Trump: Being friends with North Korea's Kim is possible

real
Trump, his party: an American odd couple

real
Trump's not-so-quick fix to undo Obamacare

fake
Trump VP's plane slides off runway at New York airport

fake
Trump WON 1/3 the 700 Counties that Voted for Obama

fake
Trump Supporter Arrested for Voting Twice...to fight "vote rigging"

fake
Trump Victory Necessary to Get US Into World War?

fake
Trump Vows To "Renovate" the Bill of Rights

fake
Trump To Clear Way For Oil Pipelines

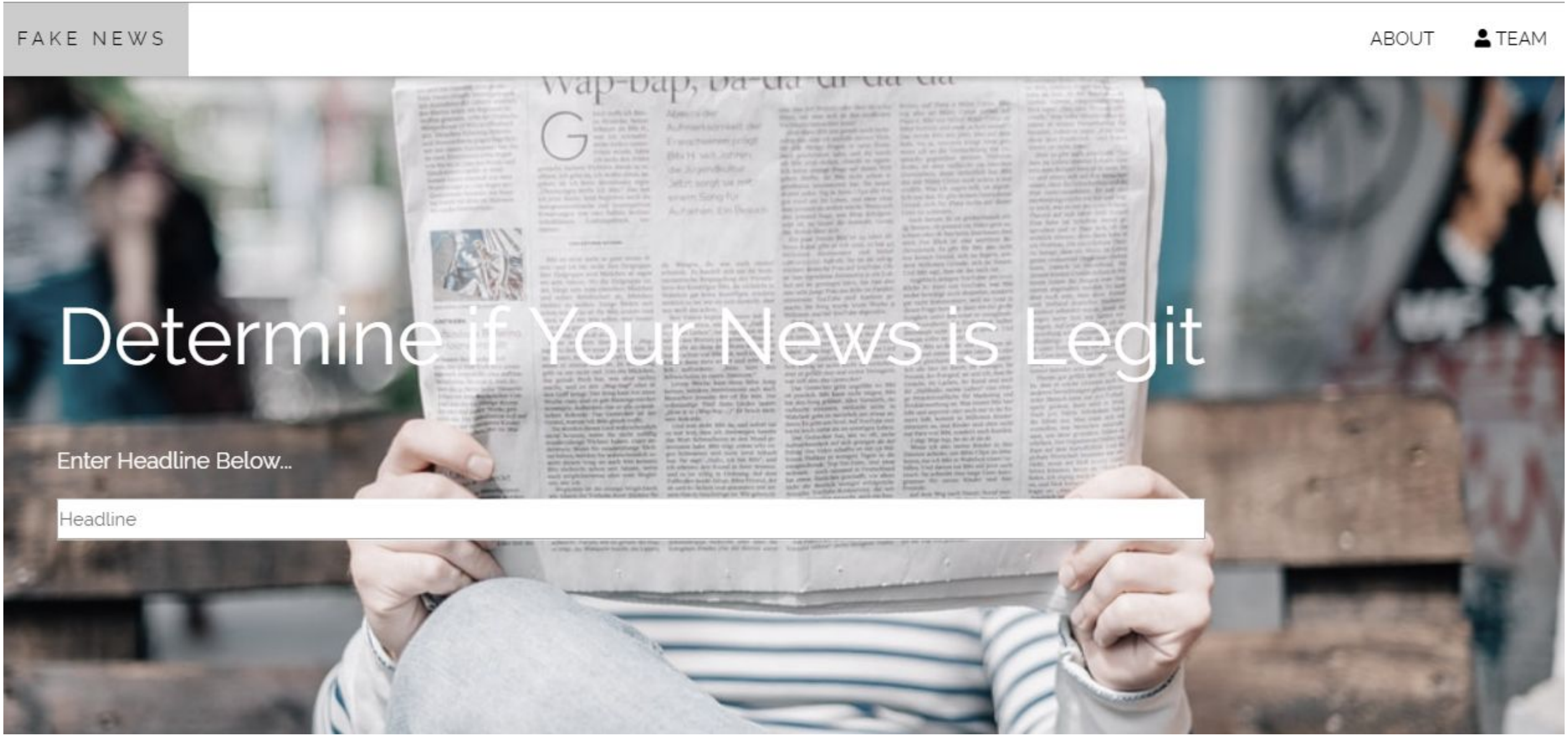
fake
Trump Vows To Bring Back Ohio Town's White Castle

fake
Trump Will Prove Bill Clinton Was Jack the Ripper

fake
Trump Takes The Kosher Seal

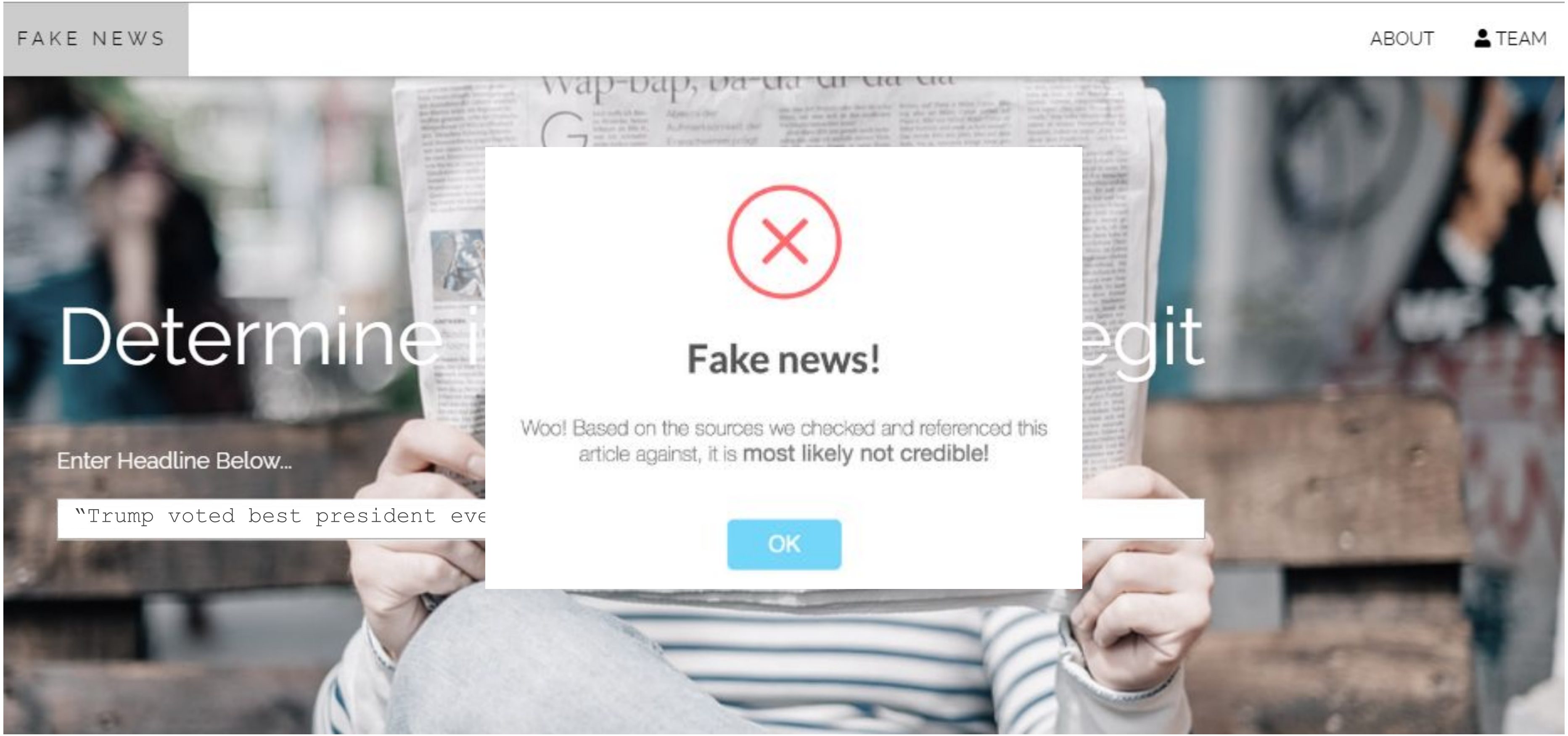
fake
Trump Tells The Truth

Step 5: Web Application



Determine if Your News is Legit

Enter Headline Below...



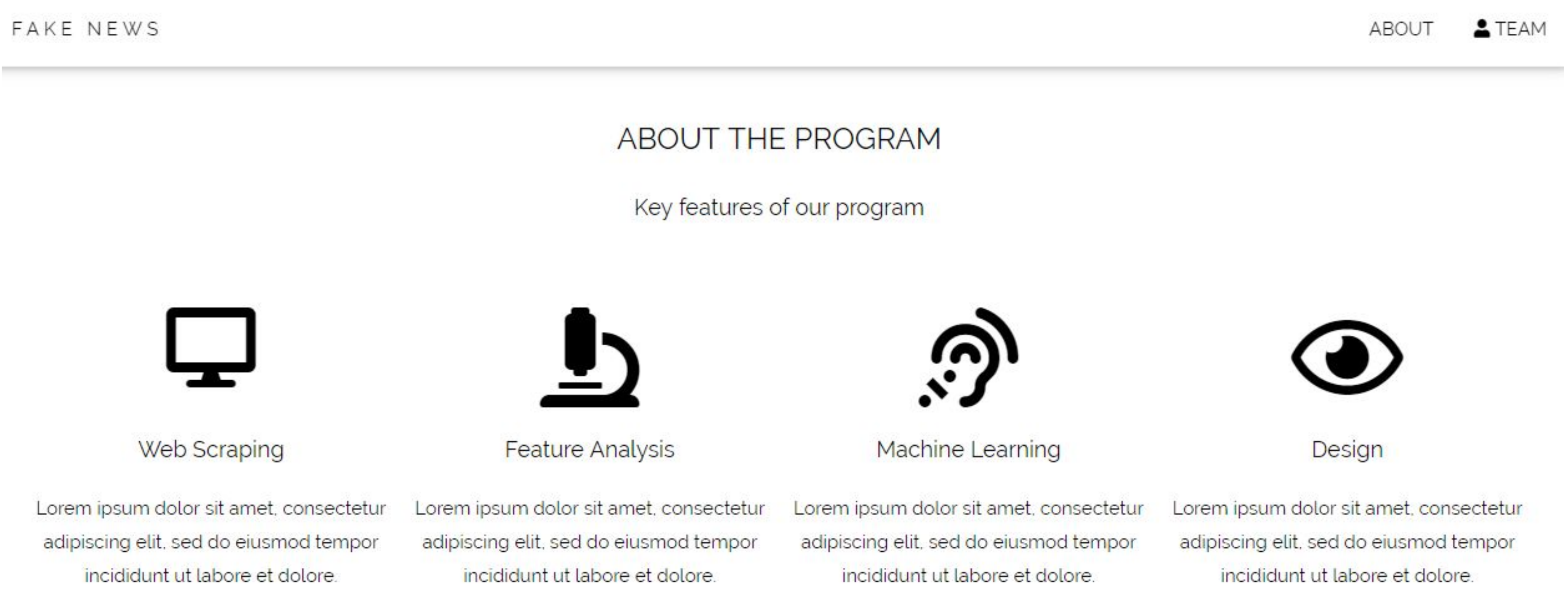
Determine i egit

Enter Headline Below...

“Trump voted best president eve

Fake news!

Woo! Based on the sources we checked and referenced this article against, it is **most likely not credible!**



ABOUT THE PROGRAM

Key features of our program

Web Scraping

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore.

Feature Analysis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore.

Machine Learning

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore.

Design

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore.