

Imports and Reading Data

```
In [1]: # libraries
import os, sys
import pandas as pd
import pickle
import re, string
import spacy, nltk
import sklearn, gensim, tweepy, pyLDAvis
from wordcloud import WordCloud
```

/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/gensim/similarities/_init_.py:15: UserWarning: The gensim.similarities.levenshtein submodule is disabled, because the optional Levenshtein package <<https://pypi.org/project/python-Levenshtein/>> is unavailable. Install Levenshtein (e.g. `pip install python-Levenshtein`) to suppress this warning.

```
warnings.warn(msg)
```

/Users/aprilyang/.local/lib/python3.8/site-packages/sklearn/decomposition/_lda.py:28: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
EPS = np.finfo(np.float).eps
```

```
In [2]: # reference source: https://medium.datadriveninvestor.com/trump-tweets-topic-modeling-1234567890
df=pd.read_csv("../data/tweets_01-08-2021.csv")
df.head()
```

/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during the transform in `preprocessing_exc_tuple` in IPython 7.17 and above.

```
and should_run_async(code)
```

```
Out[2]:
```

	id	text	isRetweet	isDeleted	device	favorites	retweets
0	98454970654916608	Republicans and Democrats have both created ou...	f	f	TweetDeck	49	255
1	1234653427789070336	I was thrilled to be back in the Great city of...	f	f	Twitter for iPhone	73748	17404
2	1218010753434820614	RT @CBS_Herridge: READ: Letter to surveillance...	t	f	Twitter for iPhone	0	7396
3	1304875170860015617	The Unsolicited Mail In Ballot Scam is a major...	f	f	Twitter for iPhone	80527	23502
4	1218159531554897920	RT @MZHemingway: Very friendly telling of even...	t	f	Twitter for iPhone	0	9081

In [3]:

```
tweets_df=df.loc[:,['text']]
tweets_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56571 entries, 0 to 56570
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  ---
0    text      56571 non-null    object
dtypes: object(1)
memory usage: 442.1+ KB
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
    and should_run_async(code)
```

In [4]:

```
# quick glance at the text
a = 56500
for i in range(a,a+10):
    print(tweets_df.text[i])
    print()
```

Great support coming from all sides for Border Security (including Wall) on our very dangerous Southern Border. Teams negotiating this weekend! Washington Post and NBC reporting of events, including Fake sources, has been very inaccurate (to put it mildly)!

Thank you to Kanye West for your nice words. Criminal Justice Reform is now law - passed in a very bipartisan way!

Great new book by Dr. Robert Jeffress, "Choosing the Extraordinary Life." Get it and enjoy! @LouDobbs

The story in the New York Times regarding Jim Webb being considered as the next Secretary of Defense is FAKE NEWS. I'm sure he is a fine man, but I don't know Jim, and never met him. Patrick Shanahan, who is Acting Secretary of Defense, is doing a great job!

GREAT JOBS NUMBERS JUST ANNOUNCED!

How do you impeach a president who has won perhaps the greatest election of all time, done nothing wrong (no Collusion with Russia, it was the Dems that Colluded), had the most successful first two years of any president, and is the most popular Republican in party history 93%?

As I have stated many times, if the Democrats take over the House or Senate, there will be disruption to the Financial Markets. We won the Senate, they won the House. Things will settle down. They only want to impeach me because they know they can't win in 2020, too much success!

....President Trump deserves a lot of credit, but again, you have the anti-Trump people who are not going to give him a lot of credit."

Michael Pillsbury interviewed by @cvpayne: "They have the motive of making the President look bad - instead of President Trump being portrayed as a HERO. The first President to take China on, it's 20 years overdue...."

<https://t.co/jsOrDtwdEa>

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

Data Cleaning -- Using code from Mark

In [5]:

```
# Data cleaning:
# Lowercase text
# Remove brackets using regular expressions
# remove punctuation and numbers using regular expressions

def clean_text(text):
    text = text.lower()
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r'\[.*?\]', '', text)

    text = re.sub(r'^\w\s', '', text)
    text = re.sub("https?://([\s]+)", ' ', text) # links
    text = re.sub(r'^https?:\/\/\.[^\s]*$', '', text)

    text = re.sub("rt", ' ', text) # RT :
    text = re.sub("&", ' ', text) # &
    text = re.sub("\n|\r|\t", ' ', text) # new line, tabs, etc
    text = re.sub('[!,-;:"\'"'\[\]\{\}]', ' ', text) # punct
    text = re.sub('\s{2,}', ' ', text) # 2+ whitespaces

    return text

tweets_df_clean = pd.DataFrame(tweets_df.text.apply(lambda x: clean_text(x)))
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

```
<>:12: DeprecationWarning: invalid escape sequence \s
<>:18: DeprecationWarning: invalid escape sequence \[
<>:19: DeprecationWarning: invalid escape sequence \s
<>:12: DeprecationWarning: invalid escape sequence \s
<>:18: DeprecationWarning: invalid escape sequence \[
<>:19: DeprecationWarning: invalid escape sequence \s
<ipython-input-5-17f1bc126e7b>:12: DeprecationWarning: invalid escape sequence
\s
    text = re.sub("https?://([\s]+)", ' ', text) # links
<ipython-input-5-17f1bc126e7b>:18: DeprecationWarning: invalid escape sequence \
[
    text = re.sub('[!,-;:"\'"'\[\]\{\}]', ' ', text) # punct
<ipython-input-5-17f1bc126e7b>:19: DeprecationWarning: invalid escape sequence
\s
    text = re.sub('\s{2,}', ' ', text) # 2+ whitespaces
```

In [6]:

```
tweets_df_clean.head()
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
```

```
n IPython 7.17 and above.
    and should_run_async(code)
```

```
Out[6]:
```

	text
0	republicans and democrats have both created ou...
1	i was thrilled to be back in the great city of...
2	cbs_herridge read letter to surveillance cou ...
3	the unsolicited mail in ballot scam is a major...
4	mzhemingway very friendly telling of events h...

```
In [7]: from nltk.corpus import wordnet
import nltk
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
# nltk.download('wordnet') ##### need this line or will get erro
import string
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
    and should_run_async(code)
```

```
In [8]: # Perform Lemmatization to reduce inflected words to their root words.----cannot
# nlp = spacy.load("en_core_web_sm")
# def lemmatizer(text):
#     sent = []
#     doc = nlp(text)
#     for word in doc:
#         sent.append(word.lemma_)
#     return " ".join(sent)

from nltk.corpus import wordnet

stop = set(stopwords.words('english'))
# manually adding stopwords
overused = ['thank','thanks','president','new','big','nice','like','time','year'
            'want','good','little','never','wants','want','thing','follow','foll
            'see','high','low','says','day','today','different','realdonaldtrump
for i in overused:
    stop.add(i)

def lemmatizer(doc):
    lemma = WordNetLemmatizer()
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
    # punc_free = ''.join(ch for ch in stop_free if ch not in exclude)
    normalized = " ".join(lemma.lemmatize(word) for word in stop_free.split())
    return normalized

tweets_df_clean = pd.DataFrame(tweets_df_clean.text.apply(lambda x: lemmatizer(x
tweets_df_clean['text'] = tweets_df_clean['text'].str.replace('-PRON-', ''))
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
```

omatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.

and should_run_async(code)

Easy EDAs

In [9]:

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

plt.figure(figsize=(10,6))
doc_lens = [len(d) for d in tweets_df_clean.text]
plt.hist(doc_lens, bins = 100)
plt.title('Distribution of Tweets character length')
plt.ylabel('Number of Tweets')
plt.xlabel('Tweets character length')
sns.despine();
```

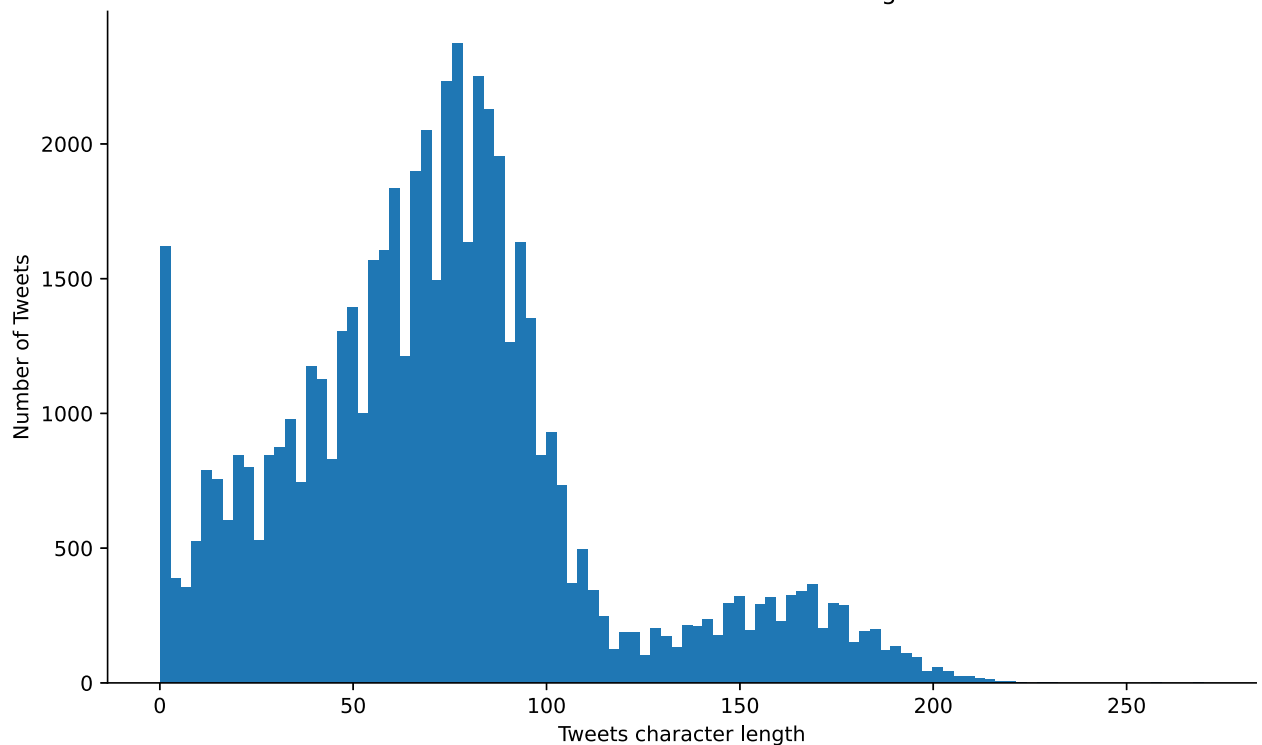
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.

and should_run_async(code)

/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/pylab/config.py:70: DeprecationWarning: InlineBackend.figure_formats_changed is deprecated in traitlets 4.1: use @observe and @unobserve instead.

def _figure_formats_changed(self, name, old, new):

Distribution of Tweets character length



In [10]:

```
import matplotlib as mpl
from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS

mpl.rcParams['figure.figsize']=(12.0,12.0)
```

```
mpl.rcParams['font.size']=12
mpl.rcParams['savefig.dpi']=100
mpl.rcParams['figure.subplot.bottom']=.1
stopwords = set(STOPWORDS)
wordcloud = WordCloud(
    background_color='white',
    stopwords=stopwords,
    max_words=500,
    max_font_size=40,
    random_state=100
).generate(str(tweets_df_clean.text))
print(wordcloud)
fig = plt.figure(1)
plt.imshow(wordcloud)
plt.axis('off')
plt.show();
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.
```

```
and should_run_async(code)
```

```
<wordcloud.wordcloud.WordCloud object at 0x7f831cdb9f10>
```



Unigrams: remove all stop words to get unigrams

```
In [11]: from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

def get_top_n_words(corpus, n=None):
    vec = CountVectorizer(stop_words='english').fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
    words_freq = sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]

common_words = get_top_n_words(tweets_df_clean.text, 10)
unigram = pd.DataFrame(common_words, columns = ['unigram', 'count'])
unigram
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

Out[11]:

	unigram	count
0	great	7578
1	people	3507
2	country	2750
3	america	2441
4	democrat	2403
5	job	2324
6	state	2103
7	make	2061
8	news	2056
9	american	2050

Trigrams

In [12]:

```
def get_top_n_trigram(corpus, n=None):
    vec = CountVectorizer(ngram_range=(3,3),stop_words='english').fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]

common_words = get_top_n_trigram(tweets_df_clean.text, 10)
trigram = pd.DataFrame(common_words, columns = ['trigram' , 'count'])
trigram
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

Out[12]:

	trigram	count
0	make america great	595
1	fake news medium	259
2	complete total endorsement	249
3	happy bi hday	149
4	crooked hillary clinton	137
5	radical left democrat	114
6	sleepy joe biden	89

	trigram	count
7	let make america	84
8	strong crime border	72
9	fake news cnn	71

Topic modeling with LDA

```
In [13]: from sklearn.decomposition import LatentDirichletAllocation
vectorizer = CountVectorizer(
    analyzer='word',
    min_df=3, # minimum required occurrences of a word
    stop_words='english', # remove stop words
    lowercase=True, # convert all words to lowercase
    token_pattern='[a-zA-Z0-9]{3,}', # num chars > 3
    max_features=5000, # max number of unique words
)

data_matrix = vectorizer.fit_transform(tweets_df_clean.text)
data_matrix
```

/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during the transform in `preprocessing_exc_tuple` in IPython 7.17 and above.

and should_run_async(code)

```
Out[13]: <56571x5000 sparse matrix of type '<class 'numpy.int64'>'
with 422455 stored elements in Compressed Sparse Row format>
```

```
In [14]: lda_model = LatentDirichletAllocation(
    n_components=10, # Number of topics
    learning_method='online',
    random_state=20,
    n_jobs = -1 # Use all available CPUs
)

lda_output = lda_model.fit_transform(data_matrix)
```

/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during the transform in `preprocessing_exc_tuple` in IPython 7.17 and above.

and should_run_async(code)

```
In [15]: # !pip install pyLDAvis
import pyLDAvis
import pyLDAvis.sklearn
pyLDAvis.enable_notebook()
p=pyLDAvis.sklearn.prepare(lda_model, data_matrix, vectorizer, mds='tsne')
p=pyLDAvis.save_html(p, '../output/lda_original_v2.html')
```

/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` argument

and any exception that happen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and above.

```
and should_run_async(code)
/Users/aprilyang/.local/lib/python3.8/site-packages/sklearn/metrics/pairwise.py:
58: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`
`. To silence this warning, use `float` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use `np.
float64` here.
```

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
dtype = np.float
/Users/aprilyang/.local/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:3
49: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`
`. To silence this warning, use `float` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use `np.
float64` here.
```

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
error = np.finfo(np.float).max
/Users/aprilyang/.local/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:3
50: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`
`. To silence this warning, use `float` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use `np.
float64` here.
```

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
best_error = np.finfo(np.float).max
/Users/aprilyang/.local/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:3
49: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`
`. To silence this warning, use `float` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use `np.
float64` here.
```

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
error = np.finfo(np.float).max
/Users/aprilyang/.local/lib/python3.8/site-packages/sklearn/manifold/_t_sne.py:3
50: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`
`. To silence this warning, use `float` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use `np.
float64` here.
```

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
best_error = np.finfo(np.float).max
```

In [16]:

```
for i,topic in enumerate(lda_model.components_):
    print(f'Top 10 words for topic #{i}:')
    print([vectorizer.get_feature_names()[i] for i in topic.argsort()[-10:]])
    print('\n')
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

```
Top 10 words for topic #0:
```

```
['doesnt', 'white', 'nation', 'whitehouse', 'sta', 'people', 'better', 'watch',
'look', 'house']
```

```
Top 10 words for topic #1:
```

```
['senator', 'cou', 'witch', 'hunt', 'rating', 'book', 'congratulation', 'tonigh
t', 'best', 'great']
```

Top 10 words for topic #2:

```
['iran', 'time', 'course', 'wall', 'border', 'fbi', 'security', 'story', 'record', 'national']
```

Top 10 words for topic #3:

```
['democrat', 'foxnews', 'united', 'world', 'win', 'really', 'state', 'country', 'people', 'job']
```

Top 10 words for topic #4:

```
['tomorrow', 'forward', 'man', 'people', 'got', 'work', 'make', 'donald', 'america', 'great']
```

Top 10 words for topic #5:

```
['election', 'democrat', 'let', 'poll', 'republican', 'vote', 'repo', 'medium', 'fake', 'news']
```

Top 10 words for topic #6:

```
['looking', 'law', 'campaign', 'clinton', 'country', 'love', 'hillary', 'obama', 'great', 'american']
```

Top 10 words for topic #7:

```
['word', 'change', 'join', 'mexico', 'debate', 'thing', 'cnn', 'going', 'true', 'year']
```

Top 10 words for topic #8:

```
['trump', 'yesterday', 'mueller', 'enjoy', 'obamacare', 'morning', 'collusion', 'meeting', 'getting', 'foxandfriends']
```

Top 10 words for topic #9:

```
['total', 'vote', 'democrat', 'tax', 'border', 'suppo', 'china', 'deal', 'run', 'need']
```

```
In [18]: # get top 30 words in each topic and generate a csv file
df_topic=pd.DataFrame(columns=['Topic', 'TopWords'])
word_list=[]

for i,topic in enumerate(lda_model.components_):
    sub_list=[]
    print(f'Top 30 words for topic #{i}:')
    print([vectorizer.get_feature_names()[i] for i in topic.argsort()[-30:]])
    print('\n')
    sub_list.append([vectorizer.get_feature_names()[i] for i in topic.argsort()[-30:]])
    word_list.append(sub_list)

word_list

df_topic['Topic']=[1,2,3,4,5,6,7,8,9,10]
df_topic['TopWords']=word_list

df_topic.head()
df_topic.to_csv('../output/topic top words v2.csv',index=None)
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
    and should_run_async(code)
```

```
Top 30 words for topic #0:
```

```
['wait', 'teamtrump', 'service', 'building', 'press', 'million', 'war', 'energ
y', 'truly', 'dont', 'lot', 'candidate', 'remember', 'open', 'presidential', 'en
d', 'problem', 'live', 'place', 'soon', 'doesnt', 'white', 'nation', 'whitehous
e', 'sta', 'people', 'better', 'watch', 'look', 'house']
```

```
Top 30 words for topic #1:
```

```
['absolutely', 'highest', 'given', 'robe', 'mark', 'miss', 'save', 'celebrity',
'win', 'russian', 'apprenticenbc', 'god', 'wow', 'ivankatrump', 'happen', 'stan
d', 'celebapprentice', 'apprentice', 'justice', 'case', 'senator', 'cou', 'witec
h', 'hunt', 'rating', 'book', 'congratulation', 'tonight', 'best', 'great']
```

```
Top 30 words for topic #2:
```

```
['failing', 'success', 'texas', 'going', 'price', 'south', 'truth', 'cruz', 'tar
iff', 'drug', 'korea', 'federal', 'company', 'lie', 'schiff', 'lost', 'happy',
'york', 'hit', 'fantastic', 'iran', 'time', 'course', 'wall', 'border', 'fbi',
'security', 'story', 'record', 'national']
```

```
Top 30 words for topic #3:
```

```
['election', 'history', 'administration', 'didnt', 'far', 'dont', 'thats', 'bes
t', 'usa', 'republican', 'russia', 'bad', 'right', 'night', 'said', 'impeachmen
t', 'say', 'working', 'real', 'hard', 'democrat', 'foxnews', 'united', 'world',
'win', 'really', 'state', 'country', 'people', 'job']
```

```
Top 30 words for topic #4:
```

```
['tower', 'criminal', 'sign', 'allowed', 'stock', 'comey', 'ive', 'voting', 'fut
ure', 'hotel', 'ready', 'market', 'potus', 'rally', 'government', 'golf', 'flori
da', 'john', 'makeamericagreatagain', 'billion', 'tomorrow', 'forward', 'man',
'people', 'got', 'work', 'make', 'donald', 'america', 'great']
```

```
Top 30 words for topic #5:
```

```
['fraud', 'force', 'southern', 'phony', 'country', 'stay', 'ing', 'jim', 'disast
er', 'voter', 'iowa', 'major', 'fox', 'dems', 'point', 'number', 'office', 'hono
r', 'interview', 'stop', 'election', 'democrat', 'let', 'poll', 'republican', 'v
ote', 'repo', 'medium', 'fake', 'news']
```

```
Top 30 words for topic #6:
```

```
['year', 'guy', 'carolina', 'city', 'million', 'political', 'ant', 'order', 'lon
g', 'making', 'illegal', 'wonderful', 'governor', 'woman', 'state', 'coming', 'c
rooked', 'maga', 'impo', 'life', 'looking', 'law', 'campaign', 'clinton', 'count
ry', 'love', 'hillary', 'obama', 'great', 'american']
```

```
Top 30 words for topic #7:
```

```
['depa', 'crazy', 'bad', 'donaldjtrumpjr', 'mean', 'believe', 'night', 'race',
'fighting', 'taking', 'entrepreneur', 'rate', 'seen', 'month', 'ago', 'tremendou
s', 'sad', 'friend', 'home', 'trying', 'word', 'change', 'join', 'mexico', 'deba
te', 'thing', 'cnn', 'going', 'true', 'year']
```

```
Top 30 words for topic #8:
```

```
['politics', 'information', 'special', 'highly', 'approval', 'investigation', 'l
oser', 'used', 'interviewed', 'bush', 'youre', 'cont', 'told', 'joe', 'mike', 'g
```

```
eneral', 'corrupt', 'person', 'agree', 'team', 'trump', 'yesterday', 'mueller',
'enjoy', 'obamacare', 'morning', 'collusion', 'meeting', 'getting', 'foxandfrien
ds']
```

Top 30 words for topic #9:

```
['biden', 'dollar', 'sma', 'money', 'gop', 'leader', 'make', 'trade', 'week', 'c
rime', 'senate', 'america', 'amazing', 'congress', 'fact', 'family', 'economy',
'business', 'strong', 'military', 'total', 'vote', 'democrat', 'tax', 'border',
'suppo', 'china', 'deal', 'run', 'need']
```

```
In [19]: topic_values = lda_model.transform(data_matrix)
tweets_df['Topic'] = topic_values.argmax(axis=1)
tweets_df.head()
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

```
Out[19]:
```

	text	Topic
0	Republicans and Democrats have both created ou...	0
1	I was thrilled to be back in the Great city of...	6
2	RT @CBS_Herridge: READ: Letter to surveillance...	6
3	The Unsolicited Mail In Ballot Scam is a major...	5
4	RT @MZHemingway: Very friendly telling of even...	9

```
In [20]: # outputting the topic label and writing into a new csv file
df['Topic'] = topic_values.argmax(axis=1)
df.to_csv('../output/tweets_with_topic_label.csv', index=None)
```

```
/Users/aprilyang/opt/anaconda3/lib/python3.8/site-packages/ipykernel/ipkernel.p
y:283: DeprecationWarning: `should_run_async` will not call `transform_cell` aut
omatically in the future. Please pass the result to `transformed_cell` argument
and any exception that happen during thetransform in `preprocessing_exc_tuple` i
n IPython 7.17 and above.
```

```
and should_run_async(code)
```

```
In [ ]:
```