Final Project : All Trump Tweets.

A live-updating JSON database containing all of President Donald J. Trump's tweets. Download : http://www.trumptwitterarchive.com/archive.

- Date : 2020-03-23
- Tweets : 46,760

Hypothesis to check for:

- Is there a relationship between engagement and source type?
- Does tweet's sentiment changes during time?
- Is there a relationship between Retweet count/ Favorite count and time created? what kind?
- Is there a relationship between word choice and engagement? what kind?
- Does word choice changes over time? How? what is the impact on engagement?
- Does number of tweets sent per day impact engagement?
- Is Trump systematically uses specific words to get more engagements?

Goal:

- Predict whether tweet was sent prior to presidency or as the president
- Predict whether tweet has positive sentiment or not

Limitations:

-- Daily tweet engagements can be impacted by external factors such as natural disasters, political reasons etc.

-- Sentiments can be adjusted better for more accuracy.

Overview of Tweet Data

	source	text	created_at	retweet_count	favorite_count	is_retweet	id_str
0	Twitter for iPhone	#FraudNewsCNN #FNN https://t.co/WYUnHjjUjg	2017-07-02 13:21:42+00:00	369530	605098	0.0	881503147168071680
1	Twitter for Android	TODAY WE MAKE AMERICA GREAT AGAIN!	2016-11-08 11:43:14+00:00	344806	573283	0.0	795954831718498304
2	Twitter Web Client	Why would Kim Jong-un insult me by calling me ...	2017-11-12 00:48:01+00:00	272776	616217	0.0	929511061954297856
3	Twitter for iPhone	A$AP Rocky released from prison and on his way...	2019-08-02 17:41:30+00:00	251530	879647	0.0	1157345692517634048
4	Twitter for Android	Such a beautiful and important evening! The fo...	2016-11-09 11:36:58+00:00	220796	633253	0.0	796315640307060736

```
In [83]: df.describe()
```

Out[83]:

	retweet_count	favorite_count	is_retweet	id_str
count	46760.000000	46760.000000	46702.000000	4.676000e+04
mean	7139.992002	23275.166339	0.070853	6.948758e+17
std	11472.783368	44521.069557	0.256583	3.387586e+17
min	0.000000	0.000000	0.000000	1.698309e+09
25%	30.000000	16.000000	0.000000	3.916754e+17
50%	822.000000	119.000000	0.000000	6.295806e+17
75%	12011.500000	29571.500000	0.000000	1.038787e+18
max	369530.000000	879647.000000	1.000000	1.241897e+18

Initial stats =>

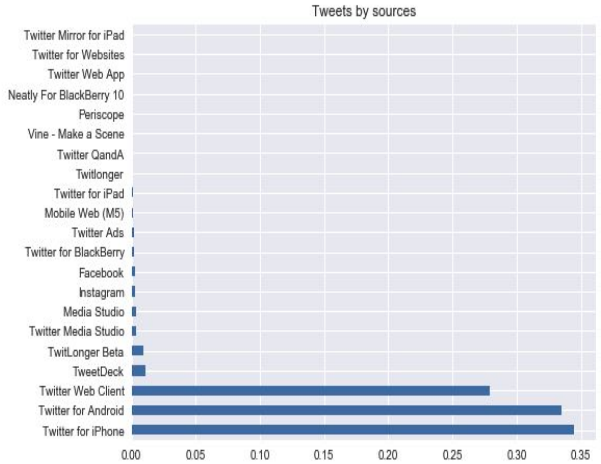
Removing Retweets from all tweets

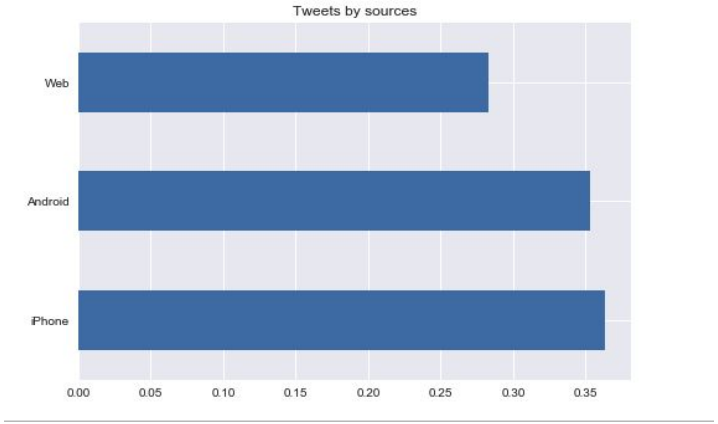
Goal is to analyze Trump's tweets

is_retweet	source	text	retweet_count	favorite_count	year_created	month_created	day_created	week_created	hour_created	weekday_created
0.0	43450	43450	43450	43450	43450	43450	43450	43450	43450	43450
1.0	3309	3309	3309	3309	3309	3309	3309	3309	3309	3309

Removing sources with less than 100 tweets

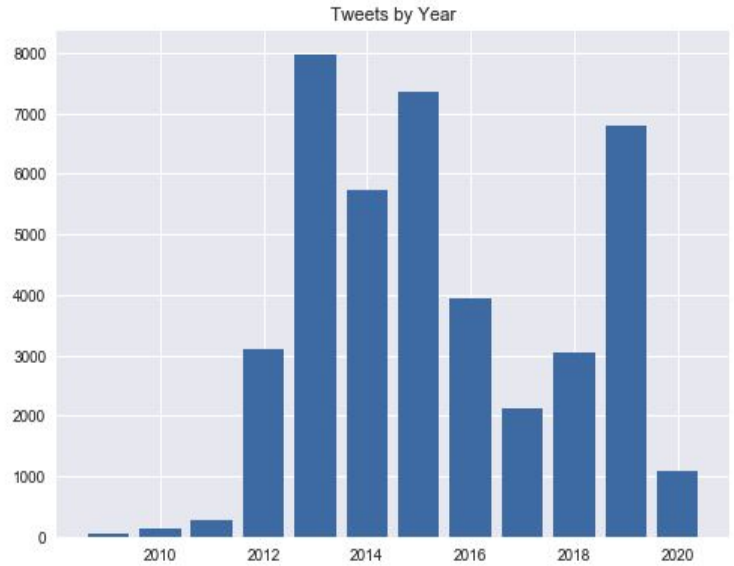
Top Sources: iphone - Android - Web

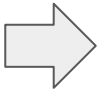


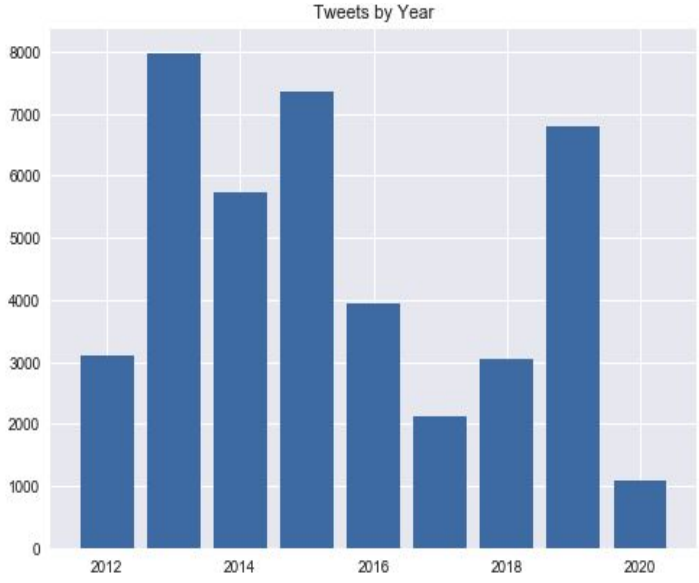


Filtering Years:

Filter out years with 1000 and less tweets

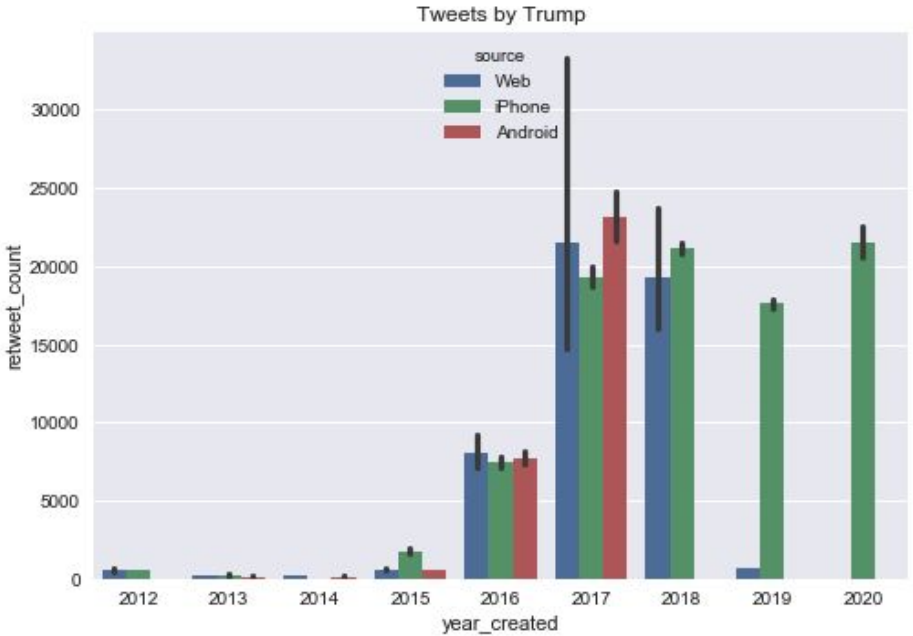






Yearly Tweets by source

It seems as president Trump uses iphone the most for tweeting

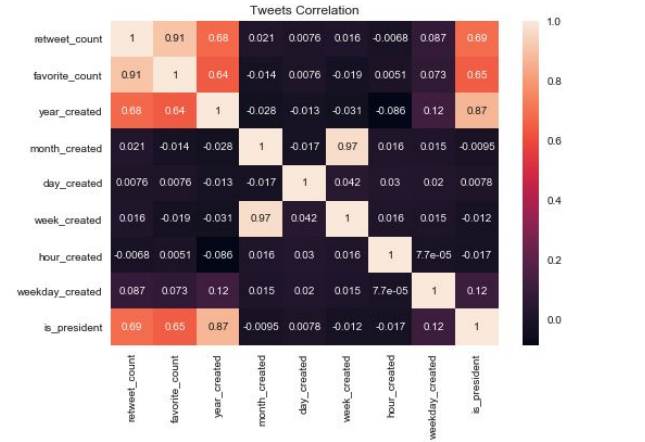


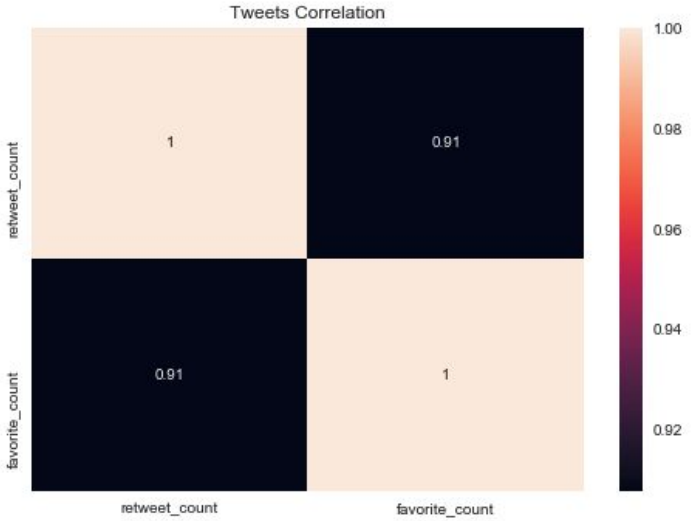
Multicollinearity Check

Will remove favorite_count since it's highly correlated with retweet_count

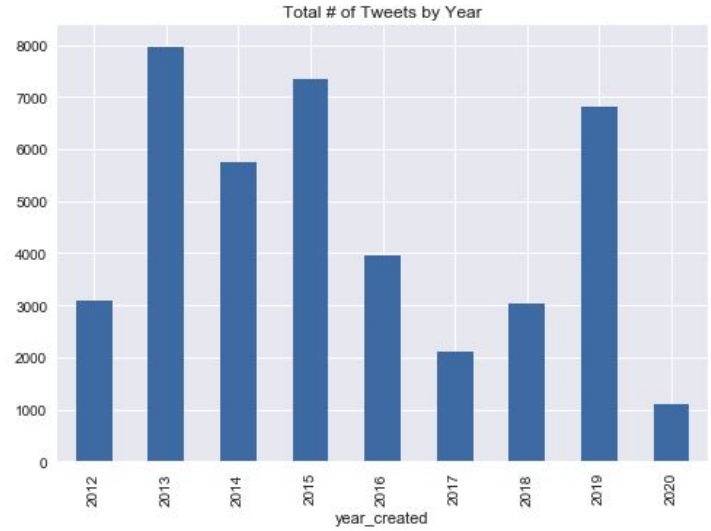
Multicollinearity Check

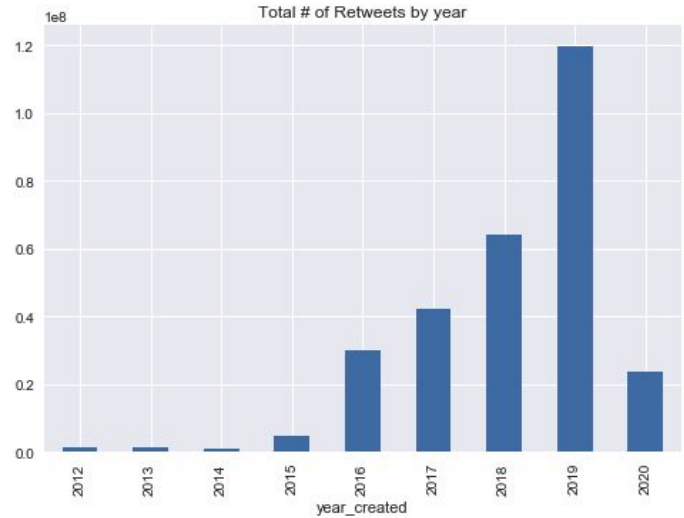
```
: tweets = df.copy()

: sns.heatmap( tweets.corr(),annot=True).set_title('Tweets Correlation'
```



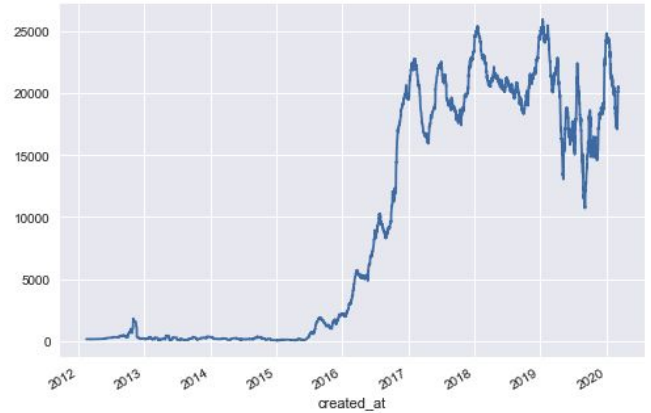


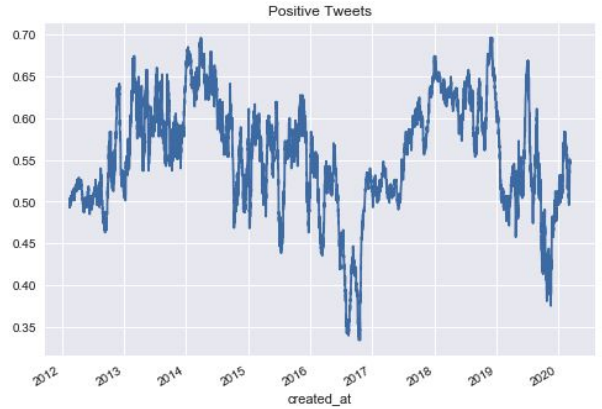
Yearly Tweets / Retweets





Yearly Tweets / Sentiment Trend





In 2016 we can see the trend is changing drastically.

2016 is the year Trump enters Presidential race and wins.

Using 2016 mark we can divide tweets to :

1- As President

2 - Prior To presidency

Also we will use Sentimental Analysis of tweets to categorize them into Positive vs Negative tweets

Target variables Distributions

is_positive	count	unique	top	freq
0	18326	18200	PRESIDENTIAL HARASSMENT!	10
1	22845	22709	MAKE AMERICA GREAT AGAIN!	33

is_president	is_positive	count	mean	std	min	25%	50%	75%	max
0	0	10297.0	414.828785	2308.866465	0.0	15.0	55.0	339.0	141644.0
	1	13864.0	319.557487	1254.307084	0.0	10.0	29.0	248.0	50145.0
1	0	8029.0	16522.170631	13558.652936	0.0	7850.0	14243.0	22045.0	369530.0
	1	8981.0	16401.219129	12090.433709	0.0	8769.0	15024.0	21436.0	220796.0

Model 1 - Results:

We will use or NLP model to predict , if a tweet is posted as president or not president.

```
Prediction results:
Null  Accuracy Score: 0.4073642281161955
Model Accuracy Score: 0.8814728456232391
------------
              precision    recall  f1-score   support

           0       0.88      0.92      0.90      6100
           1       0.88      0.82      0.85      4193

    accuracy                           0.88     10293
   macro avg       0.88      0.87      0.88     10293
weighted avg       0.88      0.88      0.88     10293
```

* Our model can predict with 88% accuracy wether or not a tweet was posted as president or prior to that versus the null accuracy of 41%.

Model 2 - Results:

We will use or NLP model to predict , if a tweet is posted is positive or negative

```
Null  Accuracy Score: 0.5690274944136792
Model Accuracy Score: 0.7506072087826678
------------
              precision    recall  f1-score   support

           0       0.82      0.54      0.65      4436
           1       0.72      0.91      0.81      5857

    accuracy                           0.75     10293
   macro avg       0.77      0.73      0.73     10293
weighted avg       0.76      0.75      0.74     10293
```

- Our model can predict with 75% accuracy wether or not a tweet is positive or negative versus the null accuracy of 56%.

Python Packages **used** in this Analysis

```
import pandas as pd
import numpy as np
import seaborn as sns
import scipy as sp
import matplotlib.pyplot as plt

from datetime import date

from statsmodels.tsa.seasonal import seasonal_decompose
from pandas.plotting import autocorrelation_plot
import nltk

import re
from nltk.tokenize import word_tokenize
from string import punctuation
from nltk.corpus import stopwords
from textblob import TextBlob, Word
from nltk.stem.snowball import SnowballStemmer

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB          # Naive Bayes
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report

from sklearn import metrics


from sklearn.linear_model import LogisticRegression , LinearRegression


%matplotlib inline
plt.style.use('seaborn')
```