

```
In [1]: # import necessary Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import urllib.request, urllib.parse, urllib.error
import requests
from bs4 import BeautifulSoup
import ssl
import re
import json
```

```
In [2]: # Ignore SSL certificate errors
ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE
```

```
In [3]: import twitter
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

```
In [4]: api = twitter.Api(consumer_key='xxxxxxxxxxxx',
                        consumer_secret='xxxxxxxxxxxxxxxxxxxx',
                        access_token_key='xxxxxxxxxxxxxxxxxxxxL',
                        access_token_secret='xxxxxxxxxxxxxxxxxxxx')
```

```

In [5]: influencer = ['Justin Bieber', 'Barack Obama', 'Katy Perry', 'Rihanna', 'Cristi
ano Ronaldo', 'Taylor Swift',
                    'Lady Gaga', 'Ariana Grande', 'Ellen DeGeneres', 'Kim Kardashian']
records = []
for item in influencer:
    query = "q={item}&result_type=recent&since=2014-07-19&count=100".format(it
em = item)
    tweets = api.GetSearch(raw_query=query)
    analyzer = SentimentIntensityAnalyzer()
    for t in tweets:
        ps = analyzer.polarity_scores(t.text)
        if ps['compound'] != 0:
            score = ps['compound']
            record = [item, score]
            records.append(record)
sentiment = pd.DataFrame.from_records(records, columns = ["Influencer", "Scor
e" ])
print(sentiment)

```

```

      Influencer  Score
0   Justin Bieber  0.5106
1   Justin Bieber -0.4389
2   Justin Bieber  0.2023
3   Justin Bieber  0.5106
4   Justin Bieber -0.3164
..          ...    ...
507  Kim Kardashian -0.5574
508  Kim Kardashian  0.1531
509  Kim Kardashian -0.7964
510  Kim Kardashian  0.4995
511  Kim Kardashian  0.4995

```

[512 rows x 2 columns]

```
In [26]: sentiment.describe()
```

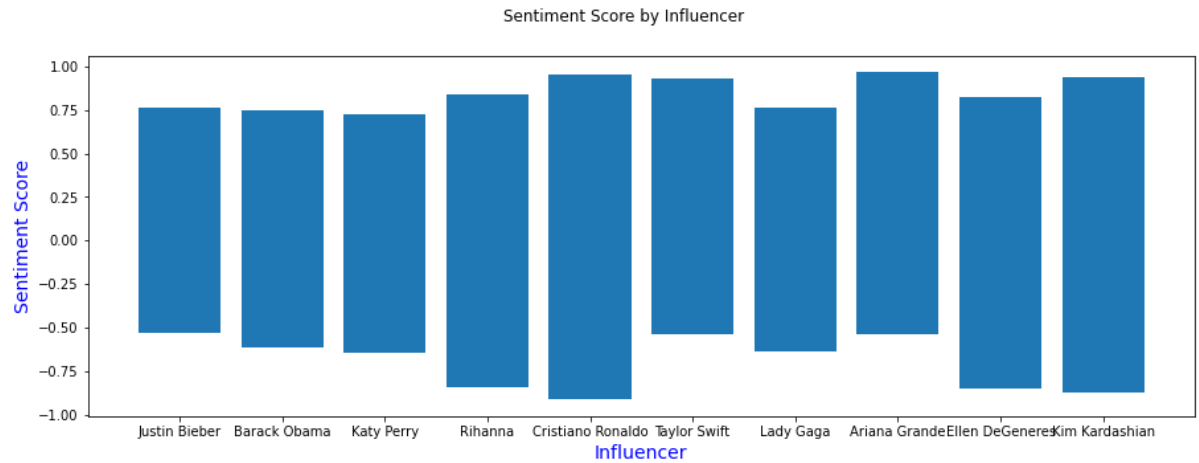
Out[26]:

	Score
count	512.000000
mean	0.243747
std	0.455317
min	-0.915300
25%	0.096325
50%	0.401900
75%	0.561025
max	0.965500

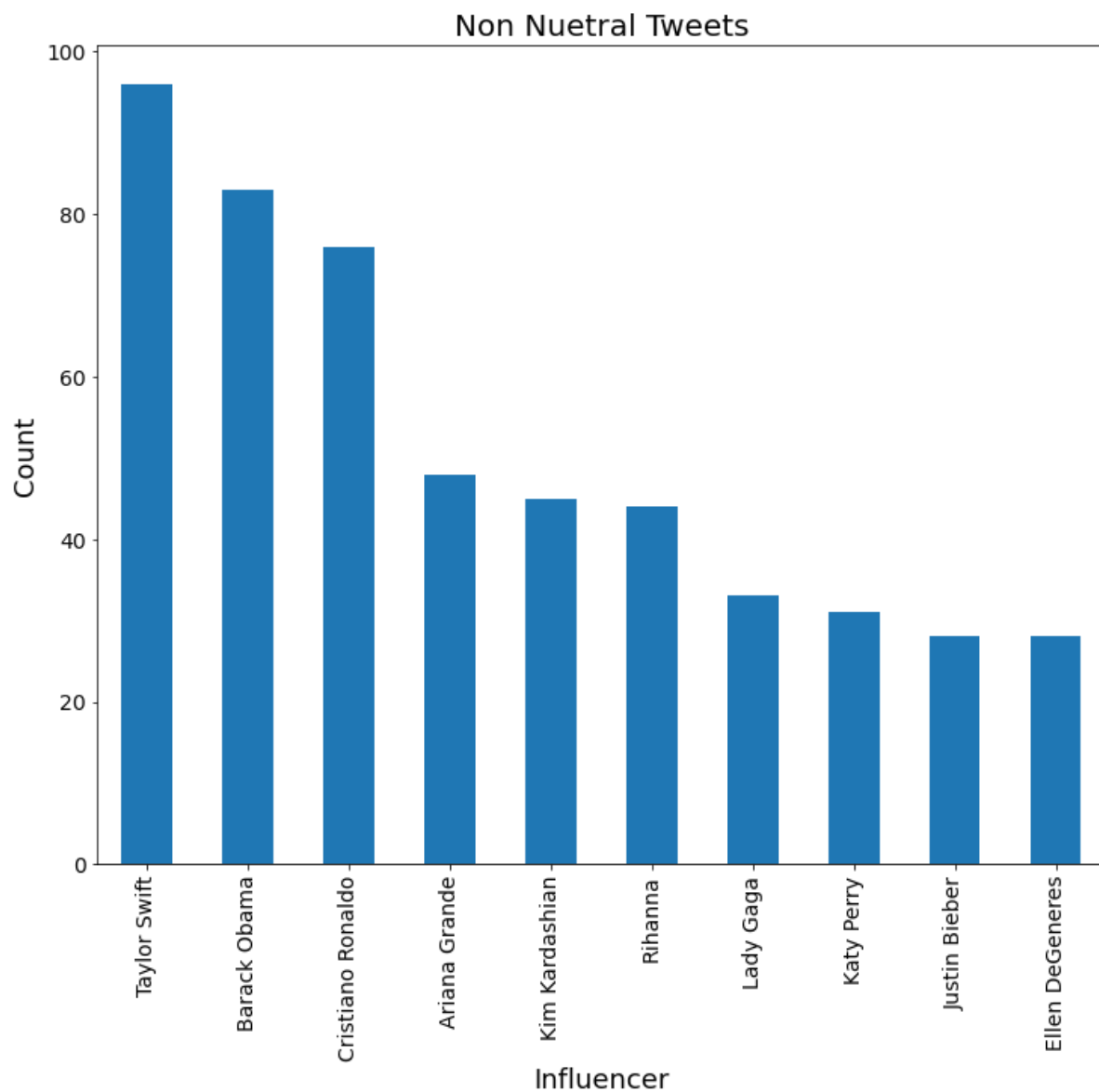
```
In [7]: df = sentiment
```

In [8]: *#plot polarity based on day of week*

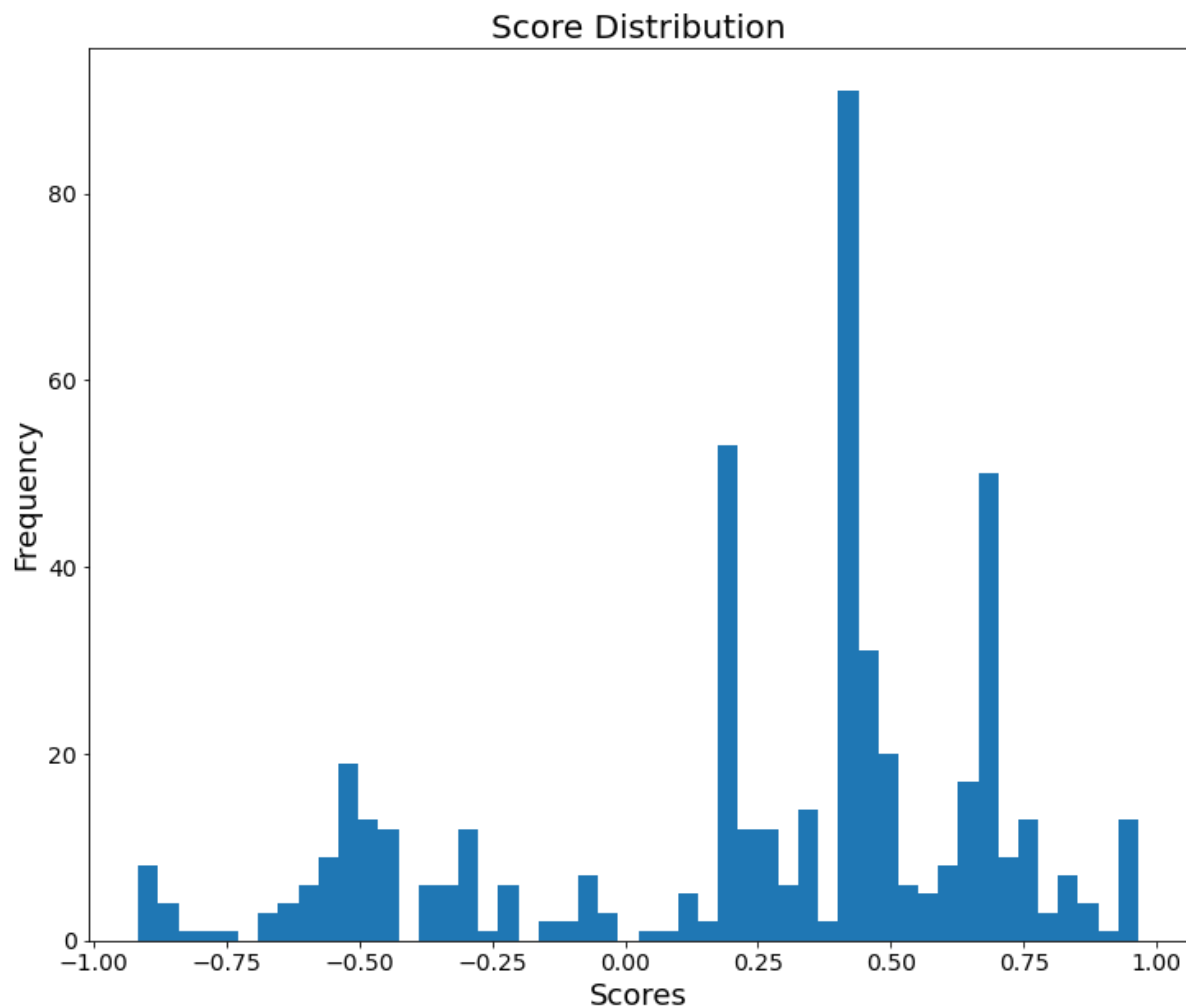
```
plt.figure(figsize=(15, 5))  
#bar plot  
plt.bar(sentiment.Influencer, sentiment.Score)  
plt.suptitle('Sentiment Score by Influencer')  
plt.xlabel('Influencer', fontsize=14, color='blue')  
plt.ylabel('Sentiment Score', fontsize=14, color='blue')  
plt.show()
```



```
In [9]: ax = sentiment.Influencer.value_counts().plot(kind='bar', fontsize=14, figsize=(12,10))
ax.set_title('Non Nuetral Tweets', fontsize=20)
ax.set_xlabel('Influencer', fontsize=18)
ax.set_ylabel('Count', fontsize=18);
plt.show()
```



```
In [10]: ax=sentiment.Score.plot(kind='hist', bins=50, fontsize=14, figsize=(12,10))
ax.set_title('Score Distribution', fontsize=20)
ax.set_ylabel('Frequency', fontsize=18)
ax.set_xlabel('Scores', fontsize=18);
plt.show()
```



```

In [11]: import plotly.graph_objects as go
from plotly.offline import iplot

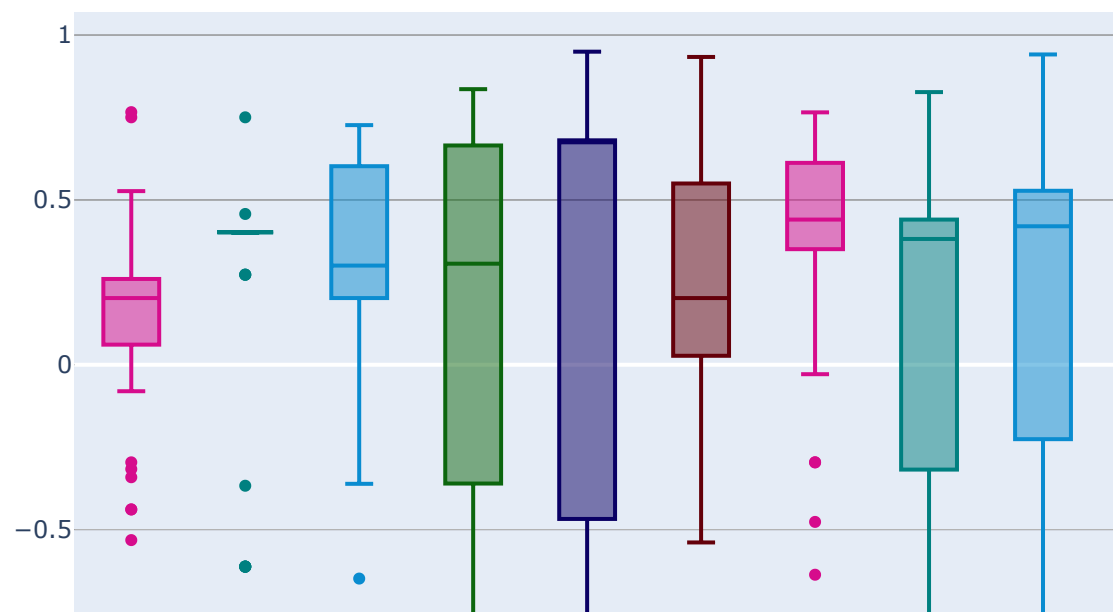
y0 = df.loc[df['Influencer'] == 'Justin Bieber']['Score']
y1 = df.loc[df['Influencer'] == 'Barack Obama']['Score']
y2 = df.loc[df['Influencer'] == 'Katy Perry']['Score']
y3 = df.loc[df['Influencer'] == 'Rihanna']['Score']
y4 = df.loc[df['Influencer'] == 'Cristiano Ronaldo']['Score']
y5 = df.loc[df['Influencer'] == 'Taylor Swift']['Score']
y6 = df.loc[df['Influencer'] == 'Lady Gaga']['Score']
y7 = df.loc[df['Influencer'] == 'Ellen DeGeneres']['Score']
y8 = df.loc[df['Influencer'] == 'Kim Kardashian']['Score']
y9 = df.loc[df['Influencer'] == 'Ariana Grande']['Score']

trace0 = go.Box(
    y=y0,
    name = 'Bieber',
    marker = dict(
        color = 'rgb(214, 12, 140)',
    )
)
trace1 = go.Box(
    y=y1,
    name = 'Obama',
    marker = dict(
        color = 'rgb(0, 128, 128)',
    )
)
trace2 = go.Box(
    y=y2,
    name = 'Perry',
    marker = dict(
        color = 'rgb(10, 140, 208)',
    )
)
trace3 = go.Box(
    y=y3,
    name = 'Rihanna',
    marker = dict(
        color = 'rgb(12, 102, 14)',
    )
)
trace4 = go.Box(
    y=y4,
    name = 'Ronaldo',
    marker = dict(
        color = 'rgb(10, 0, 100)',
    )
)
trace5 = go.Box(
    y=y5,
    name = 'Swift',
    marker = dict(
        color = 'rgb(100, 0, 10)',
    )
)

```

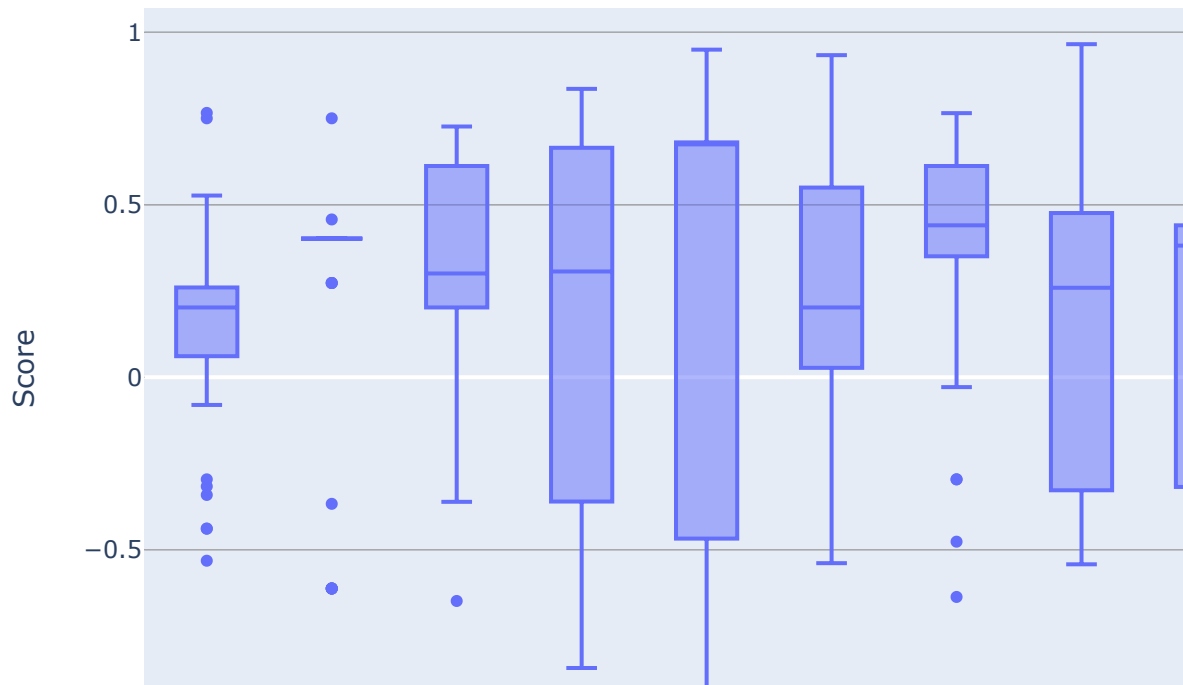
```
trace6 = go.Box(  
    y=y6,  
    name = 'Gaga',  
    marker = dict(  
        color = 'rgb(214, 12, 140)',  
    )  
)  
trace7 = go.Box(  
    y=y7,  
    name = 'DeGeneres',  
    marker = dict(  
        color = 'rgb(0, 128, 128)',  
    )  
)  
trace8 = go.Box(  
    y=y8,  
    name = 'Kardashian',  
    marker = dict(  
        color = 'rgb(10, 140, 208)',  
    )  
)  
trace9 = go.Box(  
    y=y9,  
    name = 'Grande',  
    marker = dict(  
        color = 'rgb(12, 102, 14)',  
    )  
)  
  
data = [trace0, trace1, trace2, trace3, trace4, trace5, trace6, trace7, trace8  
, trace9]  
layout = go.Layout(  
    title = "Sentiment Polarity Boxplot of Influencer"  
)  
  
fig = go.Figure(data=data,layout=layout)  
iplot(fig, filename = "Sentiment Polarity Boxplot of Influencer Name")
```

Sentiment Polarity Boxplot of Influencer




```
In [12]: import plotly.express as px

fig = px.box(df, x="Influencer", y="Score")
fig.update_traces(quartilemethod="exclusive") # or "inclusive", or "linear" by
default
fig.show()
```



```
In [20]: df2 = df.groupby('Influencer', as_index=False)['Score'].mean()
df2 = df2.sort_values('Score')
```

```
In [21]: print(df2)
```

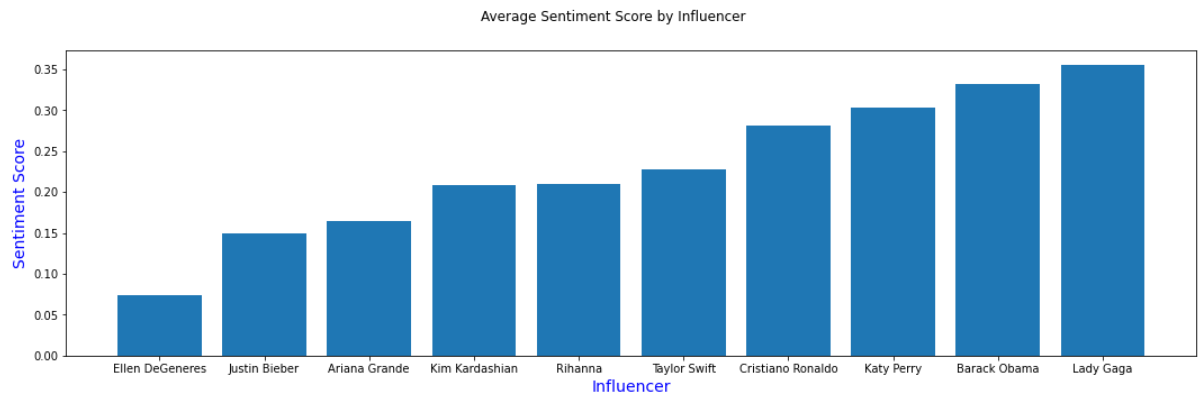
	Influencer	Score
3	Ellen DeGeneres	0.074039
4	Justin Bieber	0.148996
0	Ariana Grande	0.164975
6	Kim Kardashian	0.209204
8	Rihanna	0.210475
9	Taylor Swift	0.228406
2	Cristiano Ronaldo	0.281003
5	Katy Perry	0.303435
1	Barack Obama	0.331752
7	Lady Gaga	0.355588

```
In [22]: df3 = df.groupby('Influencer', as_index=False)['Score'].median()  
df3 = df3.sort_values('Score')
```

```
In [23]: print(df3)
```

	Influencer	Score
4	Justin Bieber	0.20230
9	Taylor Swift	0.20230
0	Ariana Grande	0.25935
5	Katy Perry	0.30080
8	Rihanna	0.30660
3	Ellen DeGeneres	0.38155
1	Barack Obama	0.40190
6	Kim Kardashian	0.42010
7	Lady Gaga	0.44040
2	Cristiano Ronaldo	0.68080

```
In [24]: plt.figure(figsize=(18, 5))  
#bar plot  
plt.bar(df2.Influencer, df2.Score)  
plt.suptitle('Average Sentiment Score by Influencer')  
plt.xlabel('Influencer', fontsize=14, color='blue')  
plt.ylabel('Sentiment Score', fontsize=14, color='blue')  
plt.show()
```



```
In [25]: plt.figure(figsize=(18, 5))  
#bar plot  
plt.bar(df3.Influencer, df3.Score)  
plt.suptitle('Median Sentiment Score by Influencer')  
plt.xlabel('Influencer', fontsize=14, color='blue')  
plt.ylabel('Sentiment Score', fontsize=14, color='blue')  
plt.show()
```

