

PSOSM

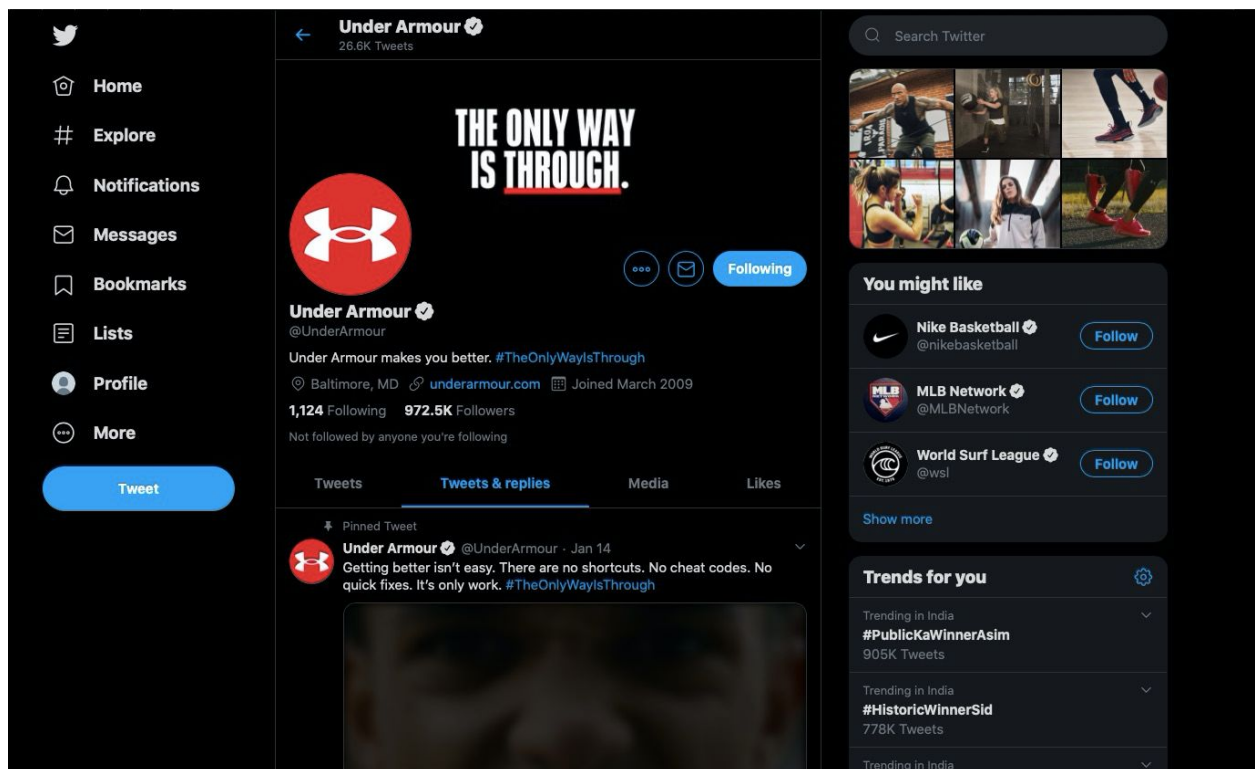
Assignment 2

- Gagandeep Singh (2016037)
- Suraj Prathik Kumar (2016101)

Question

We used python to complete all the tasks and did it in Jupyter Notebook. We used tweepy api to extract the information from twitter.

Twitter Handle : @UnderArmour



- a)** We used Cursor to collect the tweets.

Collected tweets from : 1st December 2018 to 25th January 2020.

Total Tweets between this period : 488

Stored the tweets in a string format and save it in a file

Filename : Tweets.txt

488

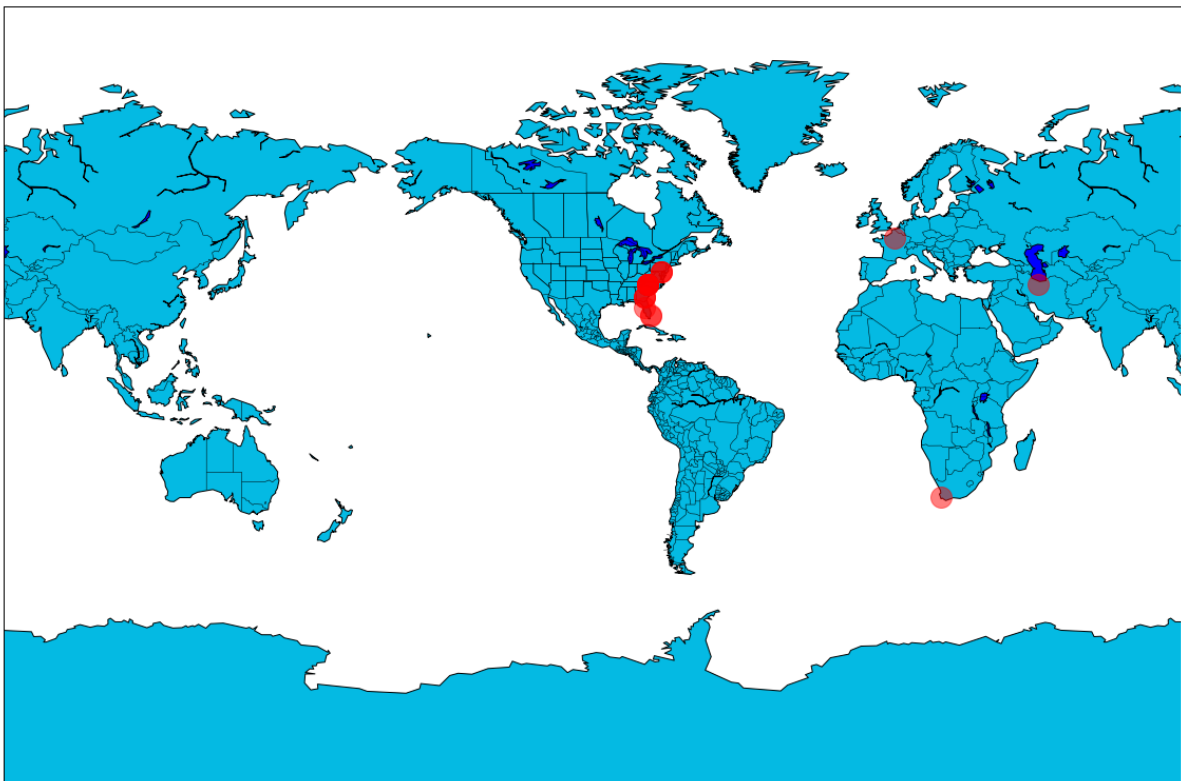
arr

[b'Get em\\xe2\\x80\\x99, @teddyriner! \\xf0\\x9f\\xa5\\x8b <https://t.co/W26A0ZKUWk>',
b'The #UAINfinityBra is our most innovative sports bra, ever. The infinity-shaped construction moves with your body
t\\xe2\\x80\\xa6 <https://t.co/FBQvKlJRc>',
b'Hear how the world\\xe2\\x80\\x99s most elite athletes train, compete and recover in our new podcast, The Only Way
Is Through. Li\\xe2\\x80\\xa6 <https://t.co/AkeJHD2rAU>',
b'.@StephenCurry30 celebrates Black History Month with a colorway inspired by David Adjaye\\xe2\\x80\\x99s iconic arc
hitectural des\\xe2\\x80\\xa6 <https://t.co/msu8wf37Vy>',
b'Getting better each day through the work and support of the team at the UA Human Performance Summit. \\xf0\\x9f\\x9
1\\x8a\\xe2\\x80\\xa6 <https://t.co/Ge6RITRYHI>',
b'To anybody who is struggling right now\\xe2\\x80\\xa6@JoelEmbiid has a story for you. #TheOnlyWayIsThrough \\xf0\\x9f
\\x91\\x89\\xe2\\x80\\xa6 <https://t.co/bL1aL2Azhj>',
b'RT @JoelEmbiid: #UAEmbiid1 @underarmour the only way is through <https://t.co/VKyJegsgMj>',
b'RT @lindseyvonn: Nothing in life is easy, especially my last season. Trying to workout and prepare for my final s
eason with a town LCL and 3\\xe2\\x80\\xa6',
b>Welcome to @Twitter, @PatrikFrisk! \\xf0\\x9f\\x91\\x8f #TheOnlyWayIsThrough <https://t.co/Ud278t0D0g>',
b'Getting better isn\\xe2\\x80\\x99t easy. There are no shortcuts. No cheat codes. No quick fixes. It\\xe2\\x80\\x99s on
ly work.\\xe2\\x80\\xa6 <https://t.co/hS0KrbqJSP>',
b'Enhanced grip & stability. Upward & downward flex. Do-it-all-durability. The UA Tribase Reign 2 features the
tools\\xe2\\x80\\xa6 <https://t.co/Nxm880yvrrl>']

- b)** We collected the first 1000 followers of the twitter handle (Under Armour) using `api.followers_ids`. We limited it to 1000 because of the wait rate limit. For each of these followers we swept through their timeline (`api.user_timeline`) and checked for posts which were geo tagged And stored them in a dictionary. These are the `user_id` of people who have tagged their posts.

```
{1563319724: [[-33.9045, 18.42]],
3428806071: [[48.85098, 2.47471], [35.781944, 51.375]],
36957942: [[1.29535636, 103.78069371],
[25.8871, -80.16532],
[26.28129302, -80.15075793]],
821375486: [[39.40887404, -76.58376594],
[39.40891813, -76.5838387],
[39.40881947, -76.58385521]],
842805157: [[31.7852078, -82.3531277],
[31.7852206, -82.3531476],
[31.7853776, -82.3531552]],
1445020056: [[28.5501746, -82.4043336]],
62992747: [[-6.518784, 106.810219],
[-6.518913, 106.810167],
[-6.518954, 106.810382],
[-6.518843, 106.810223],
[-6.518831, 106.810132]],
918898772129734656: [[35.7377, -81.3284],
[35.7377, -81.3284],
[35.7377, -81.3284],
[35.7377, -81.3284],
[35.7377, -81.3284]]}
```

We used basemap library in matplotlib to draw maps of the location of the users on the world map. Red is the location of the followers. Most of the followers were from North America.



- c) We used Tweepy.Cursor to crawl through all the tweets between the dates December 1, 2018 and January 25, 2020. Every tweet was checked for all the retweets and all the users who had retweeted were stored in a dictionary with keys as the **UserId** and value as the total number of retweets the user has made of the tweets of @UnderArmour.

```
In [34]: startDate=datetime.datetime(2018,12,1)
endDate=datetime.datetime(2020,1,25)
user={}
count=0
for tweet in tweepy.Cursor(api.user_timeline,id = "UnderArmour").items():
    if tweet.created_at>=startDate and tweet.created_at<=endDate:
        for j in api.retweeters(tweet.id):
            if(j not in user.keys()):
                user[j]=1
            else:
                user[j]+=1
        if tweet.created_at<startDate:
            break
        count+=1
        print(count)
```

Then the users were sorted in descending order based on the number of retweets they have made and stored in file.

```
In [38]: for i in high:
          print(i[0],":",i[1]," ")
```

```
1090292072865763328 : 110
1125901856843157504 : 67
1063354658910015488 : 53
2432521243 : 38
171763289 : 37
179991271 : 35
2981594310 : 34
158862823 : 32
2172359524 : 32
2283886454 : 32
```

5 inferences about these top 10 users

1. Followers/Followee:

- The number of followers and followee for each retweeter was collected and plotted on the graph.

```
In [15]: followers=[]
followee=[]
for userId in maxRetweeters.keys():
    user=api.get_user(userId)
    followers.append(user.followers_count)
    followee.append(user.friends_count)
print(followers)
print(followee)

[7186, 8, 341, 5116, 373, 12006, 2388, 139, 683, 506]
[5803, 337, 3630, 662, 1719, 13243, 4999, 72, 656, 4291]

In [74]: #code from matplotlib documentation(https://matplotlib.org/gallery/lines\_bars\_and\_markers/barchart.html#sphx-g)
screenNames=[api.get_user(userId).screen_name for userId in maxRetweeters.keys()]

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

x = np.arange(len(maxRetweeters)) # the label locations
width = 0.35 # the width of the bars

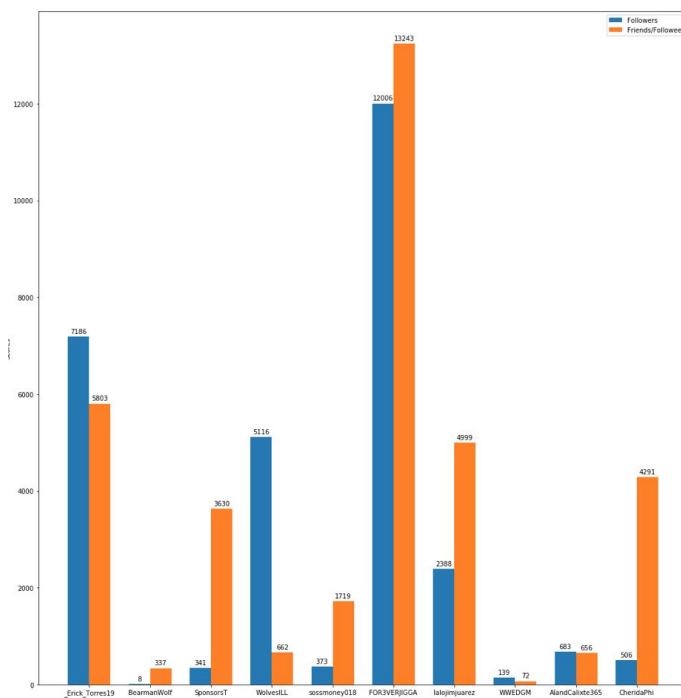
fig, ax = plt.subplots(figsize=(15,15))
rects1 = ax.bar(x - width/2, followers, width, label='Followers')
rects2 = ax.bar(x + width/2, followee, width, label='Friends/Followee')
ax.set_ylabel('Scores')

ax.set_xticks(x)
ax.set_xticklabels(screenNames)
ax.legend()

autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()
```



Analysis: Most of the users have almost equal number of followers and friends except a few. It could be inferred that these accounts could be fake accounts.

2. Frequency of Retweets

- a. All tweets which were retweets were collected for all 10 users. All these retweets were grouped according to month and for each user, the average number of retweets were calculated.

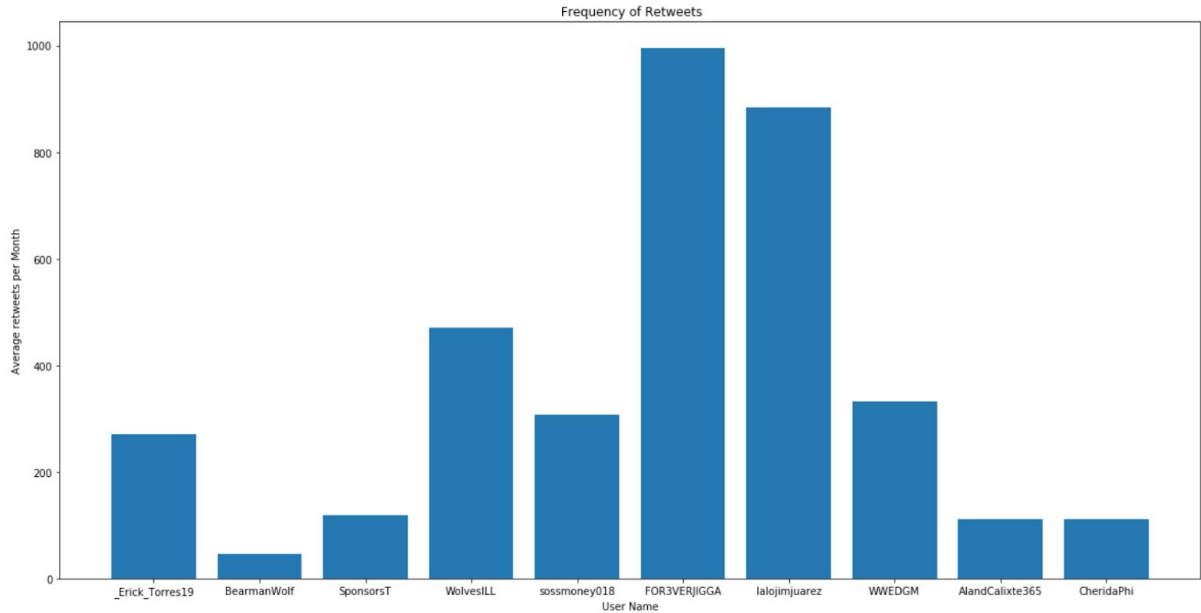
```
In [39]: allRetweets={}
for userId in maxRetweeters:
    user=api.get_user(userId)
    tweets=[]
    for tweet in tweepy.Cursor(api.user_timeline,id = userId).items(1000):
        if tweet.text.startswith("RT"):
            tweets.append(tweet.created_at)
    if(len(tweets)>0):
        allRetweets[userId]=tweets
    print(user.screen_name)

_Erick_Torres19
_BearmanWolf
_SponsorsT
_WolvesILL
_sossmoney018
_FOR3VERJIGGA
_laIojimjuarez
_WWEDGM
_AlandCalixte365
_CheridaPhi
```

```
In [42]: #average retweets per month
avgRTPerMonth={}
for userId in allRetweets.keys():
    tweets=allRetweets[userId]
    days=freqForEachUser(userId,tweets)
    # print(userId,days,len(tweets))
    totalRetweets=0
    for key in days.keys():
        totalRetweets+=days[key]
    avg=totalRetweets/len(days)
    avgRTPerMonth[api.get_user(userId).screen_name]=avg

avgRTPerMonth
```

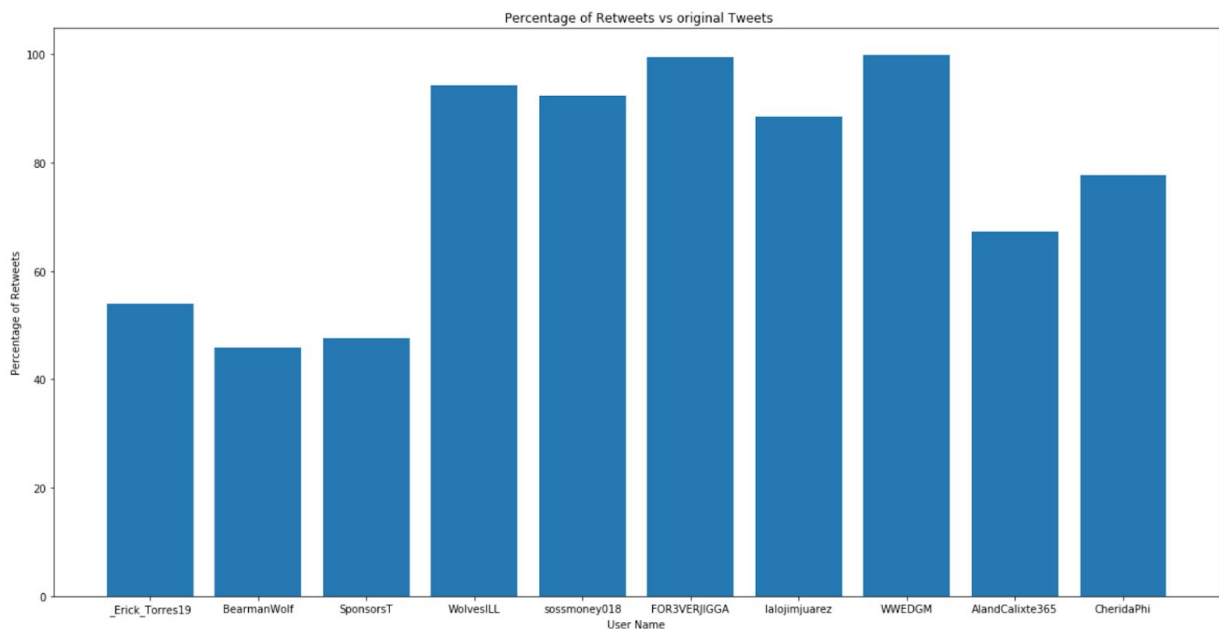
```
Out[42]: {'_Erick_Torres19': 270.0,
          'BearmanWolf': 45.8,
          'SponsorsT': 118.75,
          'WolvesILL': 471.0,
          'sossmoney018': 308.0,
          'FOR3VERJIGGA': 995.0,
          'laIojimjuarez': 884.0,
          'WWEDGM': 332.6666666666667,
          'AlandCalixte365': 112.16666666666667,
          'CheridaPhi': 110.85714285714286}
```



Analysis : Accounts which had less amount of followers are also less active. The accounts which are active tend to have 200-400 retweets per month. This shows that they have less content to share of their own.

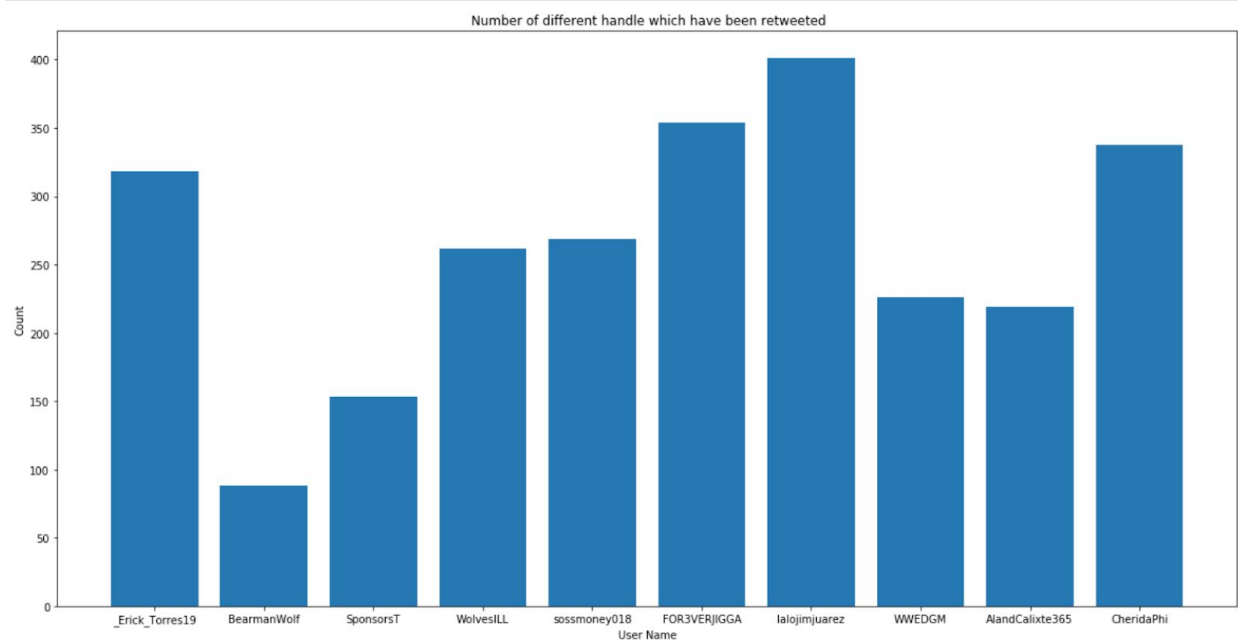
3. Percentage of Retweets vs Total Tweets

- A sample of 1000 tweets was collected for these users and checked for retweets.
- All of these accounts have more than 50% of their tweets as retweets.
- Few accounts even have as high as 998/1000 tweets as retweets

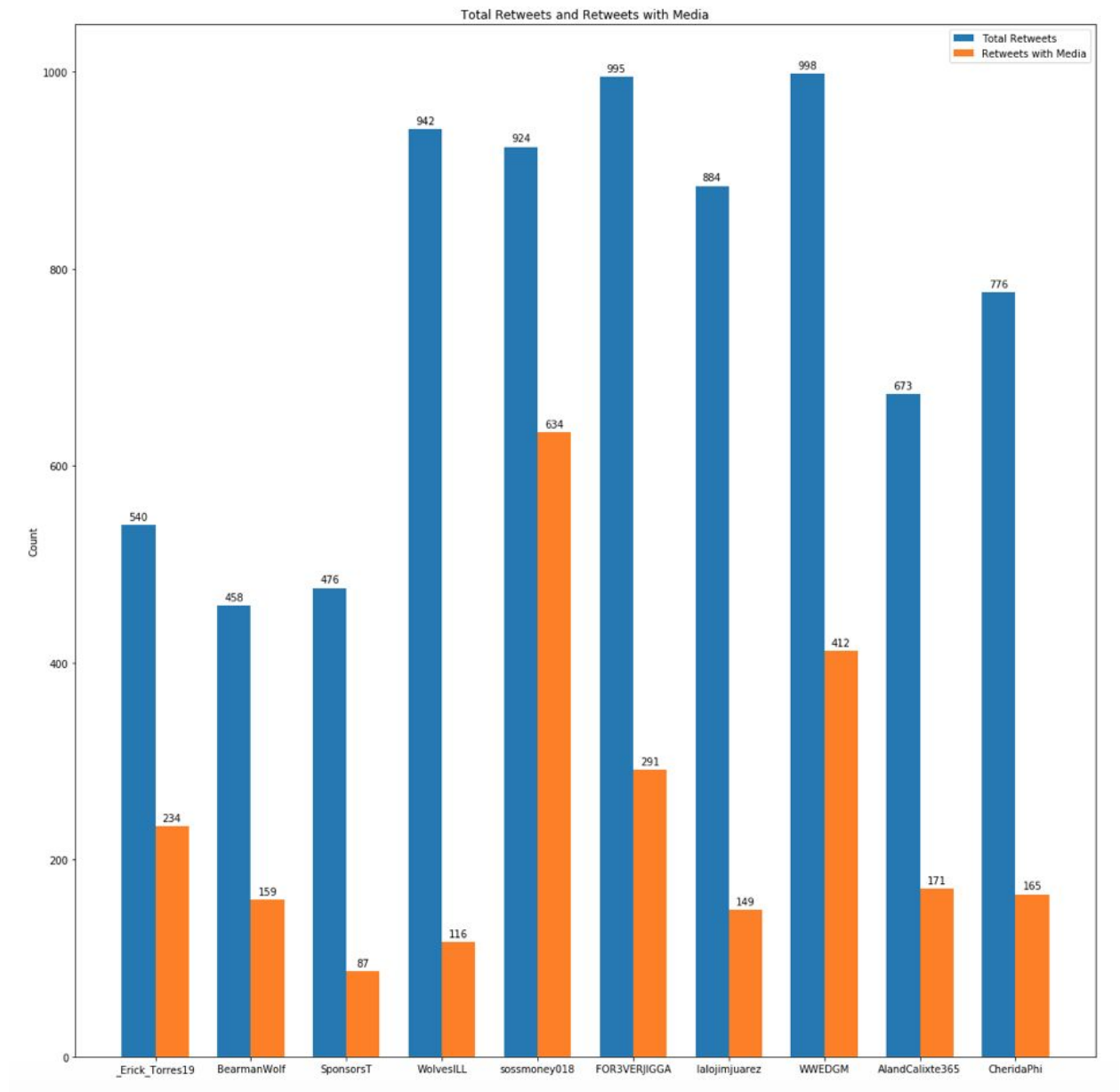


4. Number of different handles which the top retweeters have retweeted
- a. As seen above, most of the tweets on these accounts are retweets as they don't have any content of their own to show. These users tend to retweet tweets for twitter handles which are from different domain.

```
Out[60]: {'_Erick_Torres19': 318,  
          'BearmanWolf': 88,  
          'SponsorsT': 153,  
          'WolvesILL': 262,  
          'sossmoney018': 269,  
          'FOR3VERJIGGA': 354,  
          'lalojimjuarez': 401,  
          'WWEDGM': 226,  
          'AlandCalixte365': 219,  
          'CheridaPhi': 338}
```



5. Retweets that contain Media(images/video)
- a. More than 50% of the retweets by these users have media in them. This shows the reach of media is more than only text.



6. Geographical location: None of these 10 users had any geo-tag on their most recent 1000 tweets. This is in resonance with the fact that these accounts have major portions of their tweets as retweets.

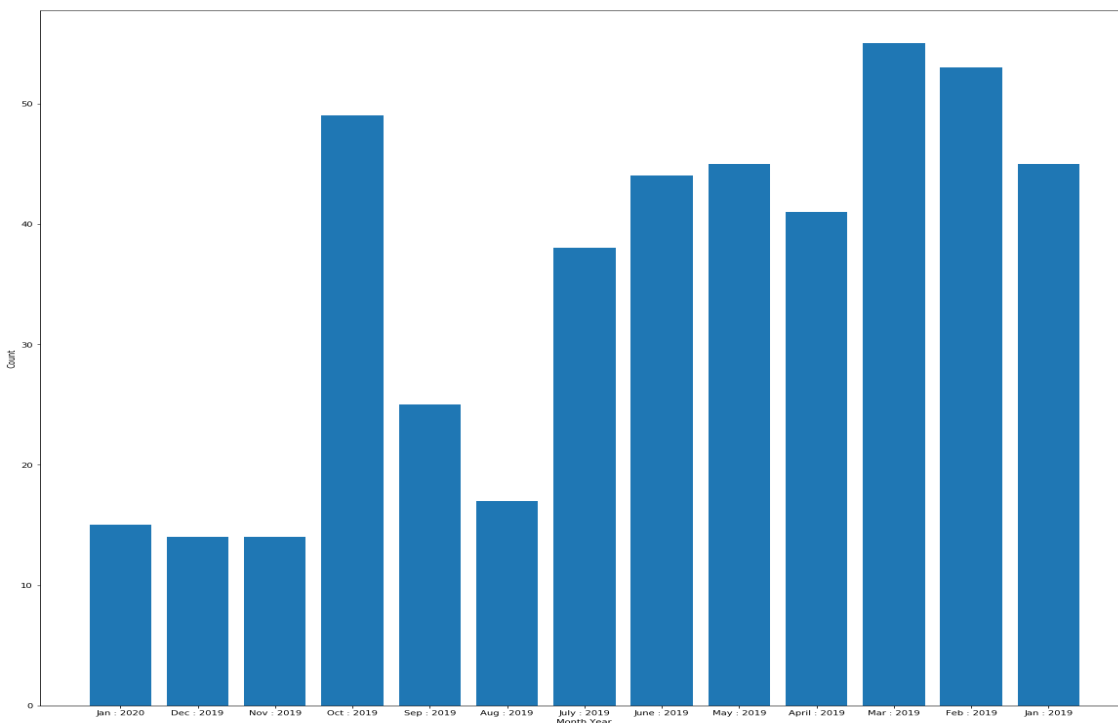
d) We collected tweets from Jan 2019 to Jan 2020 (Continuing). The following is the monthly tweet count of Under Armour.

```
{ 'Jan : 2020': 15,  
  'Dec : 2019': 14,  
  'Nov : 2019': 14,  
  'Oct : 2019': 49,  
  'Sep : 2019': 25,  
  'Aug : 2019': 17,  
  'July : 2019': 38,  
  'June : 2019': 44,  
  'May : 2019': 45,  
  'April : 2019': 41,  
  'Mar : 2019': 55,  
  'Feb : 2019': 53,  
  'Jan : 2019': 45}
```

We notice that during the start of the year the tweet counts increase during the first half of the year and then reduce in the later half.

The reasons can be that Under Armour fresh products are launched at the beginning of the year so maybe for more publicity.

It can also be that people tend to go to the gym at the beginning of the year and they want to sell maximum products during that time. Moreover, after the holiday season around the world when most of the products are on sale, the new products with revised prices are Under Armour pushes their new products through digital marketing.



e) Score - $((1.5 * \text{Number of Retweets} + 0.5 * \text{No. of Likes}) / \text{No. of Followers}) * 100$

We prioritise retweeters for calculating the score. The reason for this is that people may follow pages and but how much do they actually interact was the main focus to create the scoring system. To get this we could see how many retweets are happening, how many likes the handle is getting on their posts, how many people reply to their posts etc.

We collected the number of retweets that were done in a year from Jan'19 to Jan'20. We got 225382 retweets of the handle's post.

The second metric was like. To see how many people see the post and actually react to it. We got 64663 likes of the handle's post

Replies was also one important metric for us but the twitter api didn't support it (premium account was needed).

Finally we gave more weight to retweets as that showed how many people read the post and felt like retweeting so that others can see and gave less weightage to likes since the person can like the post just for the sake of it without seriously thinking about the post. And dividing this with the total number of followers to get a proper idea to see how many people are being influenced out of the total number of people.

To find the percentage we multiply it by 100.

We got a score of 38.08% out of 100

```
numberRetweets = 0
numberLikes = 0
numberFollowers = api.get_user(screen_name = "UnderArmour").followers_count
print(numberFollowers)
```

972509

```
for tweet in arr:
    numberLikes += tweet.favorite_count
    numberRetweets += tweet.retweet_count
print(numberLikes)
print(numberRetweets)
```

64663
225382

```
score = ((1.5 * numberRetweets + 0.5 * numberLikes) / numberFollowers) * 100
print (score)
```

38.087513843059554

- f) We use Tweepy.Cursor to collect top 2000 tweets on @UnderArmour's twitter handle.

```
In [8]: count=0
tweets=[]
for tweet in tweepy.Cursor(api.user_timeline,id = "UnderArmour").items():
    tweets.append(tweet)
    print(tweet.text)
    count+=1
    if count>2000:
        break
```

Through the work, through the sweat, through the voice in your head. Push through in @TheRock's latest #ProjectRock
... <https://t.co/x813evnUa7>
From 3 🌟 to All-Star. The latest #Curry7 celebrates @StephenCurry30's meteoric rise from unknown to superstar. The
... <https://t.co/JDQaad8jnJ>
.@HailieDeegan proves that the right training and mindset yield results. 🏆 Congrats to Hailie on finishing 2nd at
A... <https://t.co/ZMZBECc8mk>
"It's the disappointment, failure and rejection along the way that will make all the successes that much sweeter. Y
... <https://t.co/qpaw8iWM1P>
Introducing the #UAHOVRMachina. We've crafted the ultimate running shoe that syncs with UA MapMyRun to provide you...
<https://t.co/Jw3j2IjISt>
As a new parent, @natashahastings was told that qualifying for the 2020 Olympics wasn't possible...but she has a ha
... <https://t.co/3Daa0ImnxD>
A year ago @IamDeeFord lined up offsides during a crucial drive in the AFC Championship game. Ford was then traded...
<https://t.co/gHZjRAkdjk>
Here to equip you with the tools you need to reach your 2020 goals. Introducing the #UAHOVRMachina, featuring real-
... <https://t.co/yuEEHJVI1P>
The #UAINfinityBra ∞ is designed to move with you, not against you. Shop now. <https://t.co/nZyUtr5SW> <https://t.co/SJlTGuj33d>
Prayers to all those who lost their lives in Calabasas today. Including one of the greatest who ever competed. Resp

All the textual analysis was done on these 2000 tweets

1. Collecting top 10 HashTags use:
 - a. HashTags were collected from these tweets by using the following code:

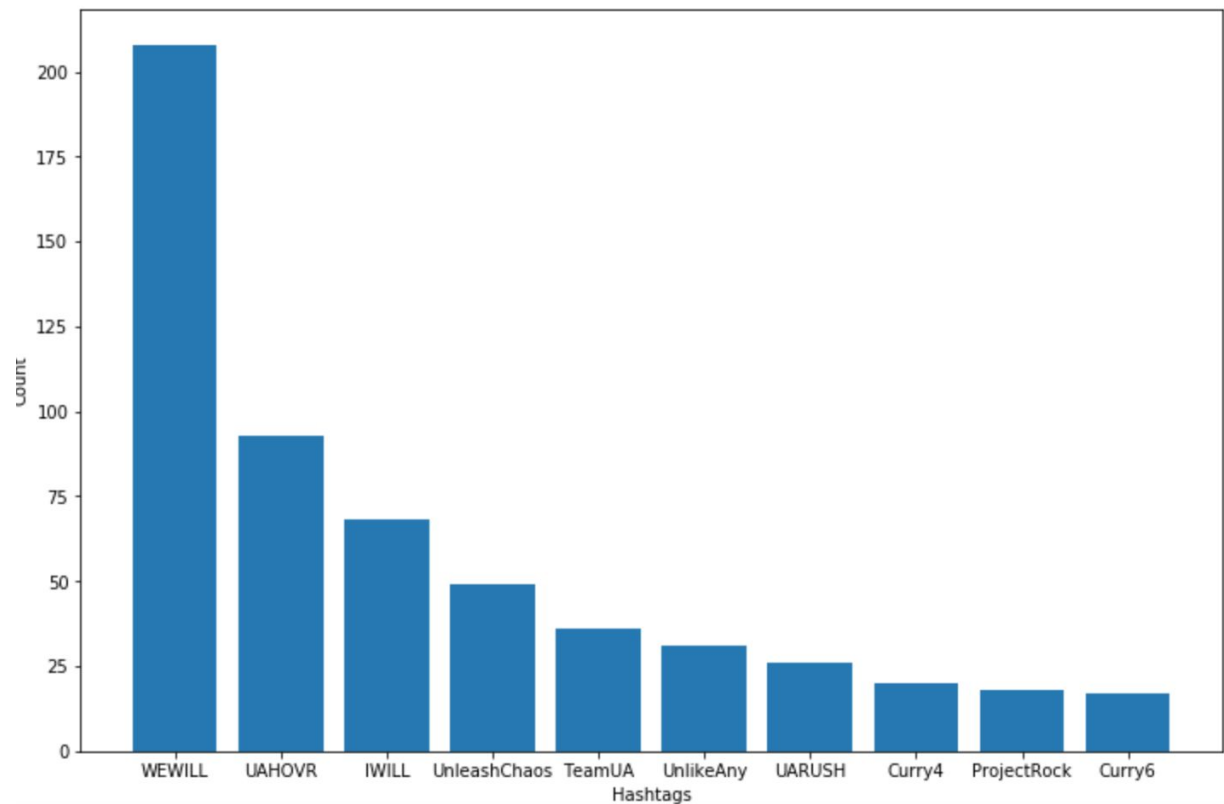
```
In [18]: hashtags={}
for tweet in tweets:
    for hashtag in tweet.entities['hashtags']:
        if hashtag['text'] in hashtags.keys():
            hashtags[hashtag['text']] += 1
        else:
            hashtags[hashtag['text']] = 1
hashtags

# if(len(tweet.entities['hashtags'])>2):
#     print(tweet.entities['hashtags'])
#     break
```

```
'FathersDay': 1,
'Pride': 5,
'UAForg96': 15,
'GlobalRunningDay': 1,
'NBAFinals': 2,
'UnitedWeWin': 1,
'WEWILL': 208,
'NBALIVEMobile': 2,
'RuinTheGame': 10,
'UAFloral': 4,
'UnderArmour': 1,
'UnleashChaos': 49,
'HAVOR': 1,
'TheSpeedProject': 1,
'WarEagle': 7,
'OpeningDay': 1,
'ATN1': 1
```

- b. Top ten hashtags were filtered out of the dictionary. The hashtags are shown below:

```
Out[23]: {'WEWILL': 208,  
          'UAHOVR': 93,  
          'IWILL': 68,  
          'UnleashChaos': 49,  
          'TeamUA': 36,  
          'UnlikeAny': 31,  
          'UARUSH': 26,  
          'Curry4': 20,  
          'ProjectRock': 18,  
          'Curry6': 17}
```

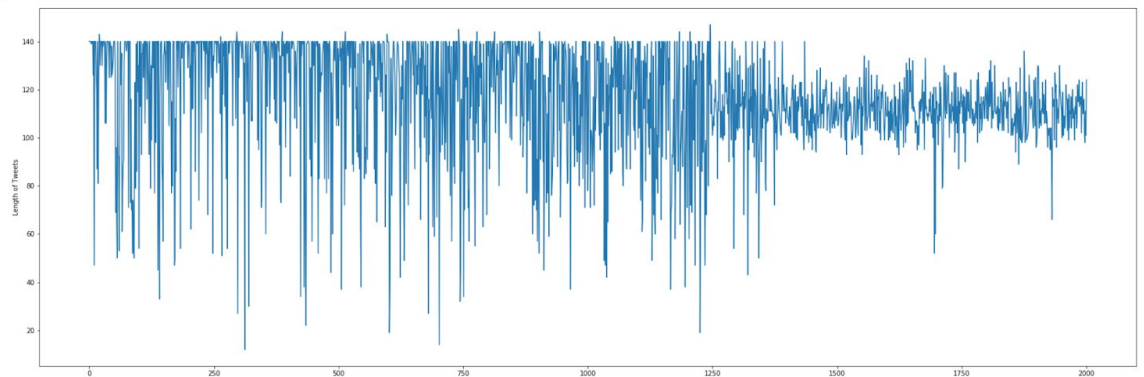


Analysis: Most common hashtags which are used are the latest technologies and product line of Under Armour which are being promoted on their Twitter handle. They also promote their company slogan, i.e., "WeWill, IWill". Other hashtags involve promotional stuff for their most famous sponsored athletes like The Rock and Stephen Curry. Under Armour is most famous for their shoes and thus hashtags like "UAHOVR", "CURRY4", "CURRY6", "UARUSH" are frequently used as these are their latest show technologies.

2. Length of tweets

- a. All the 2000 tweets were crawled to store their length using “len(tweet.text)”. Most of the tweets had a length of 140.

```
In [110]: plt.figure(figsize=(30,10))
plt.plot(lenOfTweets)
# plt.xlabel('Hashtags')
plt.ylabel('Length of Tweets')
plt.show()
```



```
In [60]: mean=np.mean(lenOfTweets)
median=np.median(lenOfTweets)
mode=stats.mode(lenOfTweets)
print('Mean:',mean)
print('Median:',median)
print('Mode:',mode[0][0], 'which appears', mode[1][0], 'times')
```

```
Mean: 116.06296851574213
Median: 118.0
Mode: 140 which appears 369 times
```

Analysis: Most of the tweets by @UnderArmour are focused more on advertising their products, especially athletic wear as it is a sports brand. Thus, almost every tweet by @UnderArmour has promotional images and videos which include their sponsored athletes. Hence, the major focus of their tweets are these promotional videos and images, thus the amount of text in these tweets is very limited and has an average length of around **116** letters per tweet. Most of the tweets have a length of **140**.

3. Creating WordCloud

- Using the **Regular Expression library** to remove the punctuation marks in the text of each tweets.

```
In [62]: import re
#regular expression library
```

```
In [65]: puncRemoved=[]
#removing punctuations
for tweet in tweets:
    puncRemoved.append(re.sub('[,\.!?', '', tweet.text))
```

```
"From 3 🌟 to All-Star The latest #Curry7 celebrates @StephenCurry30's meteoric rise from unknown to superstar The
... https://tco/JDQaad8jnJ",
'@HailieDeegan proves that the right training and mindset yield results 🏆 Congrats to Hailie on finishing 2nd at
A... https://tco/ZMZBECc8mk',
'It's the disappointment failure and rejection along the way that will make all the successes that much sweeter Y
... https://tco/qpaw8iWM1P',
Introducing the #UAHOVRMachina We've crafted the ultimate running shoe that syncs with UA MapMyRun to provide you
... https://tco/Jw3j2IjISt',
'As a new parent @natashahastings was told that qualifying for the 2020 Olympics wasn't possiblebut she has a ha...
https://tco/3Doa0Imnx0",
'A year ago @IamDeeFord lined up offsides during a crucial drive in the AFC Championship game Ford was then traded
... https://tco/gHZjRAkdjk',
'Here to equip you with the tools you need to reach your 2020 goals Introducing the #UAHOVRMachina featuring real-
... https://tco/yuEEHJVI1P',
'The #UAINfinityBra ∞ is designed to move with you not against you Shop now https://tco/nZyUtr5SW https://tco/SJ
LTGuj33d',
'Prayers to all those who lost their lives in Calabasas today Including one of the greatest who ever competed Resp
... https://tco/h9GBUHoybD',
'Get em' @teddyriner 🙌 https://tco/W26A0ZKUWK',
'The #UAINfinityBra is our most innovative sports bra ever The infinity-shaped construction moves with your body t
```

- Using **NLTK** library to remove all the stopwords(words which do not contribute to sentiment of the text, like 'a', 'the' etc). Using the **RE** library to remove links in the text as the every text contains a link to that twitter post.

Every post was converted into lower case.

```
In [100]: import nltk
from nltk.corpus import stopwords
```

```
In [101]: LowerTextTweets=[]
for tweet in puncRemoved:
    tweet=re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[*()\\.]|(?%[0-9a-fA-F][0-9a-fA-F]))+', '',tweet)
    words=tweet.split()
    withoutStop=[i for i in words if i.lower() not in stopwords.words('english')]
    tweet=' '.join(withoutStop)
    LowerTextTweets.append(tweet.lower())

LowerTextTweets
```

```
Out[101]: ['work sweat voice head push @therock's latest #projectrock...',
'3 🌟 all-star latest #curry7 celebrates @stephencurry30's meteoric rise unknown superstar the...',
'@hailiedeegan proves right training mindset yield results 🏆 congrats hailie finishing 2nd a...',
'it's disappointment failure rejection along way make successes much sweeter y...',
'introducing #uahovrmachina we've crafted ultimate running shoe syncs ua mapmyrun provide you...',
'new parent @natashahastings told qualifying 2020 olympics possiblebut ha...',
'year ago @iamdeeford lined offsides crucial drive afc championship game ford traded...',
'equip tools need reach 2020 goals introducing #uahovrmachina featuring real...',
'#uainfinitybra ∞ designed move shop',
'prayers lost lives calabasas today including one greatest ever competed resp...',
'get em' @teddyriner 🙌',
'#uainfinitybra innovative sports bra ever infinity-shaped construction moves body t...',
'hear world's elite athletes train compete recover new podcast way li...',
'stephencurry30 celebrates black history month colorway inspired david adjaye's iconic architectural des...',
'getting better day work support team ua human performance summit 🙌...',
```


- c. The text of all the tweets were joined together and **WordCloud** library was used to create a word cloud.

```
In [104]: wordcloud = WordCloud(background_color="white", max_words=2000)
```

```
In [107]: wordcloud.generate(allTweets)
wordcloud.to_image()
```

4. Top twenty word count

- a. Python's machine learning library **SKLearn** was used to count the frequency of each word that appeared in the top 2000 tweets. CountVectorizer was imported and preprocessed tweets were trained.

```
In [111]: from sklearn.feature_extraction.text import CountVectorizer
```

```
In [113]: bagOfWords = CountVectorizer().fit(lowerTextTweets)
```

```
In [118]: countOfWords=bagOfWords.transform(lowerTextTweets)
```

```
In [121]: import numpy as np
```

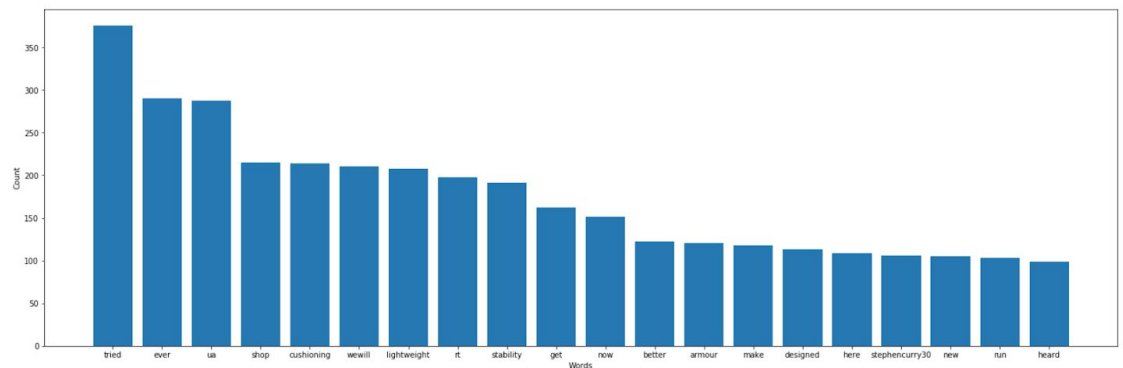
```
In [122]: totalCounts=np.zeros(len(bagOfWords.get_feature_names()))
```


b. A dictionary was created showing the top 20 words used in the tweets of @UnderArmour.

```
In [139]: mostFreqWords={}
          for word in topTenWords:
              mostFreqWords[word[0]]+=word[1]
          mostFreqWords
```

```
Out[139]: {'tried': 376.0,
           'ever': 290.0,
           'ua': 288.0,
           'shop': 215.0,
           'cushioning': 214.0,
           'wewill': 210.0,
           'lightweight': 208.0,
           'rt': 198.0,
           'stability': 191.0,
           'get': 162.0,
           'now': 151.0,
           'better': 122.0,
           'armour': 121.0,
           'make': 118.0,
           'designed': 113.0,
           'here': 109.0,
           'stephencurry30': 106.0,
           'new': 105.0,
           'run': 103.0,
           'heard': 99.0}
```

```
In [142]: plt.figure(figsize=(25,8))
          plt.bar(mostFreqWords.keys(), mostFreqWords.values())
          plt.xlabel('Words')
          plt.ylabel('Count')
          plt.show()
```



Analysis: Just like the hashtags, words used in tweets of @UnderArmour are motivational words for sportsmen, which resonates with the company's domain and their potential target audience. Moreover, words like "lightweight", "stability" etc are used to express the unique selling proposition of the products.

Use of words like 'steph curry', 'shop', 'better' etc motivates their audience to visit their site and shop their products.

- g)** We collected data for 3200 tweets. We used extended mode in Cursor to get detailed information (tweet_mode="extended").
- We segregated the tweets into 4. Only text contents, Only image contents, Only video contents, All different kinds of media content.
 - We put 4 posts on our twitter handle to check and test the json data that was returned in the tweet.
 - Only Text had no media tag under entities or in extended entities. Also removing Retweets and Reply tweets.
 - Only Image or Video had media tag under entities or in extended entities. But for full_text or text it had a link to that tweet if the post had no text (not None or Null).
 - To differentiate between the type of media(Image or Video) we used type under extended_entities as it gave a precise type of image or video else it only gave type as photo in normal entities.
 - Combination tweets had text not only the link to the tweet in their text or full_text and also had a media tag in their entities or extended_entities.

While running the code on 3200 tweets and collecting the first 20 of each type of the 4 tweets mentioned. The tweet with a combination of media and text was filled in 24 tweets/3200. Since it is a sports brand it tends to put more posts with text and images or videos to describe their product better.

```
Tweet 23
Only Text 2
Only Vid 0
Only Image 0
All Other 19
Tweet 24
Only Text 2
Only Vid 0
Only Image 0
All Other 20
Tweet 25
Only Text 2
Only Vid 0
Only Image 0
All Other 20
Tweet 26
Only Text 2
Only Vid 0
Only Image 0
All Other 20
```

The tweets which had only text were filled at 205th tweet/3200. These posts were mostly congratulatory posts congratulating other players etc.. While Only Video and Only Image were still 0.

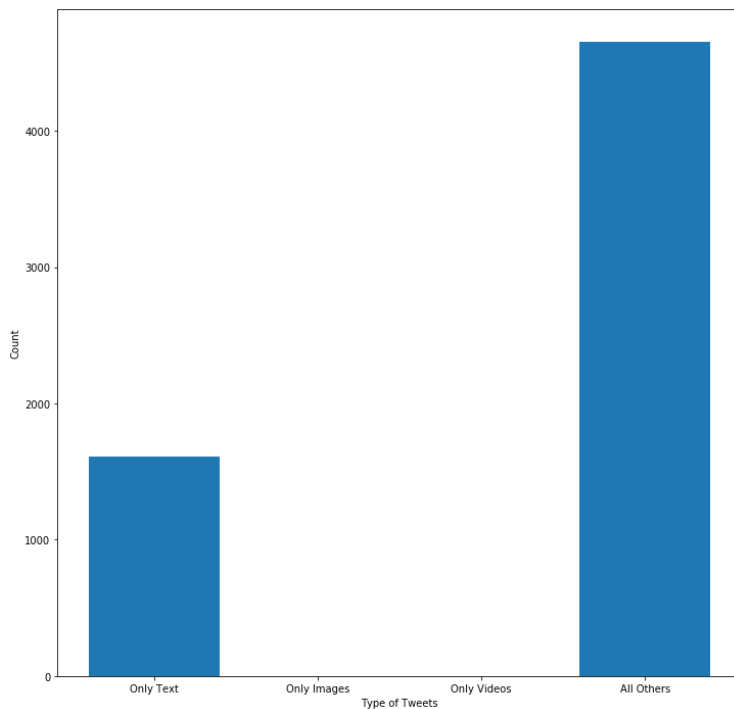
```
Tweet 204
Only Text 19
Only Vid 0
Only Image 0
All Other 20
Tweet 205
Only Text 20
Only Vid 0
Only Image 0
All Other 20
Tweet 206
Only Text 20
Only Vid 0
Only Image 0
All Other 20
Tweet 207
Only Text 20
Only Vid 0
Only Image 0
All Other 20
```

Tweet with Only Image and Only Image were 0 even at 3200th tweet. So it was concluded that only Image and only Video post did not exist at this twitter handle. The reason can be that a sports brand just doesn't put a post of Image or Video without a caption. Each and every post had some or the text of explanation of the post or a motivational line, a story etc..

```
Tweet 3197
Only Text 20
Only Vid 0
Only Image 0
All Other 20
Tweet 3198
Only Text 20
Only Vid 0
Only Image 0
All Other 20
Tweet 3199
Only Text 20
Only Vid 0
Only Image 0
All Other 20
Tweet 3200
Only Text 20
Only Vid 0
Only Image 0
All Other 20
```

So, For comparison purposes we will only take posts with Only Text and Combination of text and media.

Number of Likes

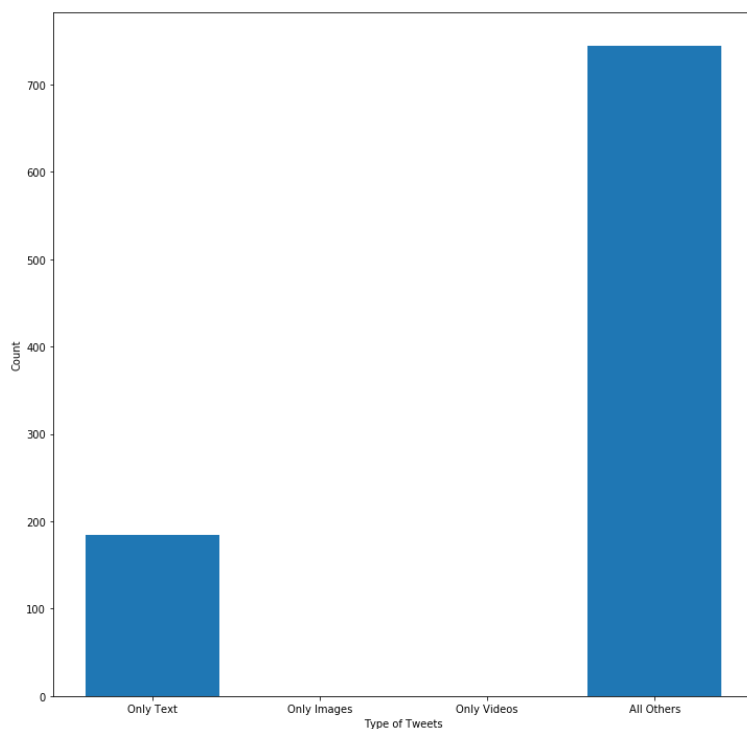


Total Number of likes on first 20 only text posts were : 1610

Total Number of likes on first 20 combination of text and media were : 4656

This shows that people don't like that posts with only text but prefer media such as photos or videos with text. It attracts more people and people tend to prefer combination of text and media posts.

Number of Retweets



Total Number of retweets on first 20 only text posts were : 185

Total Number of retweets on first 20 combination of text and media were : 745

This also clearly supports our argument that people prefer posts with media having text over just text posts. There was a drastic difference in the number of retweets for the above mentioned numbers.

References :

1. https://matplotlib.org/gallery/lines_bars_and_markers/barchart.html#sphx-glr-gallery-lines-bars-and-markers-barchart-py
2. <https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0>
3. [Matplotlib library](#)
4. <https://www.tandfonline.com/doi/pdf/10.1080/23311975.2018.1564168?needAccess=true&>
5. <https://basemaptutorial.readthedocs.io/en/latest/>
6. <http://docs.tweepy.org/en/v3.5.0/api.html>
7. <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object>