Evil Geniuses Data Analyst Intern Assessment

May 22, 2023

```
[1]: #Import relevant libraries
     import pandas as pd
     import matplotlib.pyplot as plt
     import numpy as np
     import calendar
```

```
[2]: #Read in .csv file from social_data.xlsx
     social_data=pd.read_csv('C:/Users/jakey/Downloads/social_data.xlsx - Data.csv')
     #Sort dataset by chronological date
     social_data=social_data.sort_values(by=['Published Date'])
     social_data=social_data.reset_index(drop=True)
     social_data
```

```
[2]:           Published Date    Account  Account Type        Campaign Name  \
     0      01-01-2023 14:59    General        FBPAGE                  N/A
     1      01-03-2023 10:34       CSGO       TWITTER                  N/A
     2      01-03-2023 10:34       CSGO       TWITTER                  N/A
     3      01-03-2023 10:34       CSGO       TWITTER                  N/A
     4      01-03-2023 10:35       CSGO       TWITTER                  N/A
     ...                 ...        ...           ...                  ...
     3474   03-31-2023 19:43       CSGO       TWITTER  Community Engagement
     3475   03-31-2023 19:49       CSGO       TWITTER                  N/A
     3476   03-31-2023 19:49       CSGO       TWITTER                  N/A
     3477   03-31-2023 19:49       CSGO       TWITTER                  N/A
     3478   03-31-2023 19:55       CSGO       TWITTER                  N/A

           Total Impressions  Total Engagements Media Type
     0                      0                  0      Photo
     1                      0                  0       Link
     2                   2116                 42       Text
     3                      0                  0       Link
     4                      0                  0       Link
     ...                  ...                ...        ...
     3474                9517               1215      Video
     3475                   0                  0       Text
     3476                   0                  0       Text
     3477                   0                  0       Text
     3478                   0                  0       Text
```

```
   [3479 rows x 7 columns]
```

```
[3]:  #Question 1
      #Typical engagment rate=total engagements/total impressions
      np.mean(social_data['Total Engagements']/social_data['Total Impressions'])
```

```
[3]:  0.40492621761200764
```

Commentary:

I calculate the typical engagement rate by dividing the "Total Engagements" column by the "Total Impressions" column. This gives me 3479 rows of information, one row per "Published Date" provided. In order to get the typical engagement rate expected, I take the mean of the column of quotients which gives me about a 40% typical engagement rate expected from the 3 months of data provided.

```
[4]:  social_data['Engagement Rate']=social_data['Total Engagements']/
        ↪social_data['Total Impressions']
      social_data['Engagement Rate']=social_data['Engagement Rate'].fillna(0)
      rows_greater_than_or_equal_15=len(social_data[social_data['Engagement Rate']>=.
        ↪15])
      rows_greater_than_or_equal_15/len(social_data)
```

```
[4]:  0.06496119574590399
```

Commentary:

To calculate the likelihood that we can achieve a 15% engagement rate, I create a new column in the "social_data" dataframe called "Engagement Rate" which is the quotient of "Total Engagements" divided by "Total Impressions," or the engagement rate for each row. I create a subset of the social_data dataframe where the "Engagement Rate" is greater than or equal to .15 called "rows_greater_than_or_equal_15" and find the length of it to be 226. This means there are 226 days where the engagement rate was greater than or equal to 15%. I create another subset called "rows_less_than_15" which is every other row that is less than .15. I divide 226 by the length of the "social_data" dataframe, which is 3479, to get the likelihood of an engagement rate of 15% or higher. This likelihood value is about 6.5%.

```
[5]:  #Question 2
      datetime=pd.to_datetime(social_data['Published Date'], format='%m-%d-%Y %H:%M')
      social_data['Date']=0
      social_data['Time']=0
      for i in range(len(social_data)):
          social_data.iat[i,8]=datetime[i].date()
          social_data.iat[i,9]=datetime[i].time()
      #Monday=0, Tuesday=1, Wednesday=2, Thursday=3, Friday=4, Saturday=5, Sunday=6
      social_data['Day of the Week']=0
      for i in range(len(social_data)):
          if social_data.iat[i,8].weekday()==0:
              social_data.iat[i,10]='Monday'
```

```
    elif social_data.iat[i,8].weekday()==1:
        social_data.iat[i,10]='Tuesday'
    elif social_data.iat[i,8].weekday()==2:
        social_data.iat[i,10]='Wednesday'
    elif social_data.iat[i,8].weekday()==3:
        social_data.iat[i,10]='Thursday'
    elif social_data.iat[i,8].weekday()==4:
        social_data.iat[i,10]='Friday'
    elif social_data.iat[i,8].weekday()==5:
        social_data.iat[i,10]='Saturday'
    elif social_data.iat[i,8].weekday()==6:
        social_data.iat[i,10]='Sunday'
social_data
```

[5]:

	Published Date	Account	Account Type	Campaign Name
0	01-01-2023 14:59	General	FBPAGE	N/A
1	01-03-2023 10:34	CSGO	TWITTER	N/A
2	01-03-2023 10:34	CSGO	TWITTER	N/A
3	01-03-2023 10:34	CSGO	TWITTER	N/A
4	01-03-2023 10:35	CSGO	TWITTER	N/A
...	...	...	...	...
3474	03-31-2023 19:43	CSGO	TWITTER	Community Engagement
3475	03-31-2023 19:49	CSGO	TWITTER	N/A
3476	03-31-2023 19:49	CSGO	TWITTER	N/A
3477	03-31-2023 19:49	CSGO	TWITTER	N/A
3478	03-31-2023 19:55	CSGO	TWITTER	N/A

	Total Impressions	Total Engagements	Media Type	Engagement Rate
0	0	0	Photo	0.000000
1	0	0	Link	0.000000
2	2116	42	Text	0.019849
3	0	0	Link	0.000000
4	0	0	Link	0.000000
...	...	...	...	...
3474	9517	1215	Video	0.127666
3475	0	0	Text	0.000000
3476	0	0	Text	0.000000
3477	0	0	Text	0.000000
3478	0	0	Text	0.000000

	Date	Time	Day of the Week
0	2023-01-01	14:59:00	Sunday
1	2023-01-03	10:34:00	Tuesday
2	2023-01-03	10:34:00	Tuesday
3	2023-01-03	10:34:00	Tuesday
4	2023-01-03	10:35:00	Tuesday
...	...	...	...

```
3474   2023-03-31   19:43:00              Friday
3475   2023-03-31   19:49:00              Friday
3476   2023-03-31   19:49:00              Friday
3477   2023-03-31   19:49:00              Friday
3478   2023-03-31   19:55:00              Friday

[3479 rows x 11 columns]
```

[6]:
```python
#Day of the week vs engagement rates
weekdays=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
mondaycount=0
tuesdaycount=0
wednesdaycount=0
thursdaycount=0
fridaycount=0
saturdaycount=0
sundaycount=0
for i in range(len(social_data)):
    if social_data.iat[i,10]=='Monday':
        mondaycount+=social_data.iat[i,7]
    elif social_data.iat[i,10]=='Tuesday':
        tuesdaycount+=social_data.iat[i,7]
    elif social_data.iat[i,10]=='Wednesday':
        wednesdaycount+=social_data.iat[i,7]
    elif social_data.iat[i,10]=='Thursday':
        thursdaycount+=social_data.iat[i,7]
    elif social_data.iat[i,10]=='Friday':
        fridaycount+=social_data.iat[i,7]
    elif social_data.iat[i,10]=='Saturday':
        saturdaycount+=social_data.iat[i,7]
    elif social_data.iat[i,10]=='Sunday':
        sundaycount+=social_data.iat[i,7]
weekdayrate=[mondaycount,tuesdaycount,wednesdaycount,thursdaycount,fridaycount,saturdaycount,s
plt.bar(weekdays,weekdayrate)
plt.xlabel('Weekdays')
plt.ylabel('Total Engagement Rate')
plt.title('Weekdays vs Total Engagement Rate')
plt.show()
```

Weekdays vs Total Engagement Rate
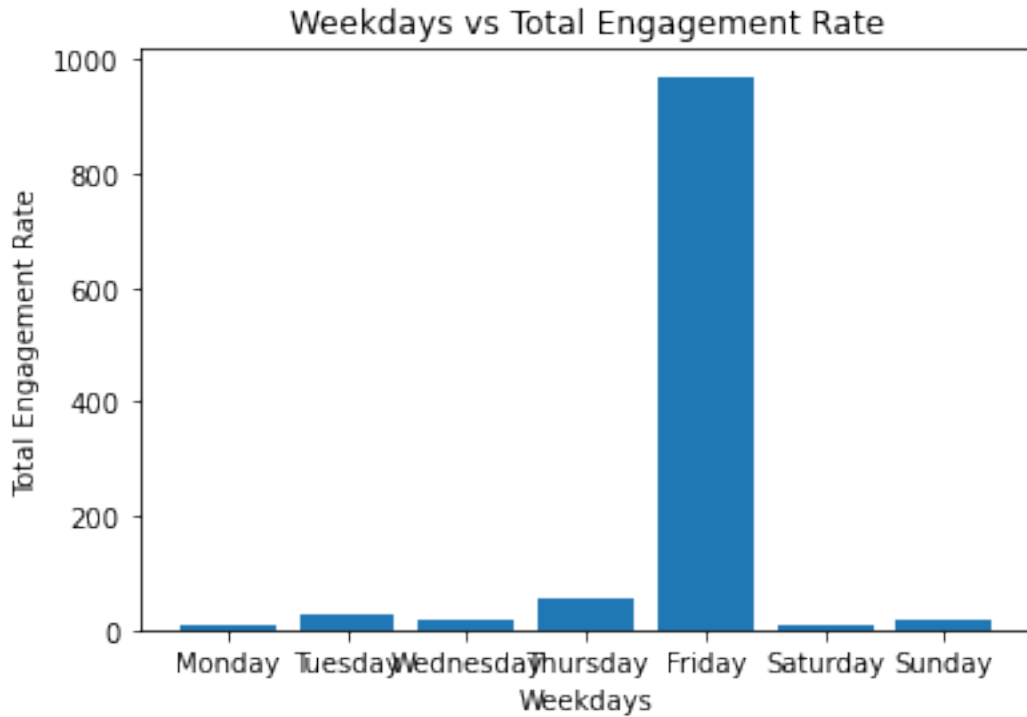
Commentary:

The histogram above shows that day of the week does affect engagement rates because Friday's engagement rate is drastically higher than the other days of the week. This means Evil Geniuses should try to post on their social accounts on Friday to get the most engagement rates.

```
[7]: #Time vs engagement rates
     count0=0
     count1=0
     count2=0
     count3=0
     count4=0
     count5=0
     count6=0
     count7=0
     count8=0
     count9=0
     count10=0
     count11=0
     count12=0
     count13=0
     count14=0
     count15=0
     count16=0
     count17=0
```

```python
count18=0
count19=0
count20=0
count21=0
count22=0
count23=0
for i in range(len(social_data)):
    if social_data.iat[i,9].hour==0:
        count0+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==1:
        count1+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==2:
        count2+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==3:
        count3+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==4:
        count4+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==5:
        count5+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==6:
        count6+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==7:
        count7+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==8:
        count8+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==9:
        count9+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==10:
        count10+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==11:
        count11+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==12:
        count12+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==13:
        count13+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==14:
        count14+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==15:
        count15+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==16:
        count16+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==17:
        count17+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==18:
        count18+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==19:
        count19+=social_data.iat[i,7]
```

```
    elif social_data.iat[i,9].hour==20:
        count20+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==21:
        count21+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==22:
        count22+=social_data.iat[i,7]
    elif social_data.iat[i,9].hour==23:
        count23+=social_data.iat[i,7]
hourcount=[count0,count1,count2,count3,count4,count5,count6,count7,count8,count9,count10,count
hours=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]
plt.bar(hours,hourcount)
plt.xlabel('Hours')
plt.ylabel('Total Engagement Rate')
plt.title('Hours vs Total Engagement Rate')
plt.show()
```
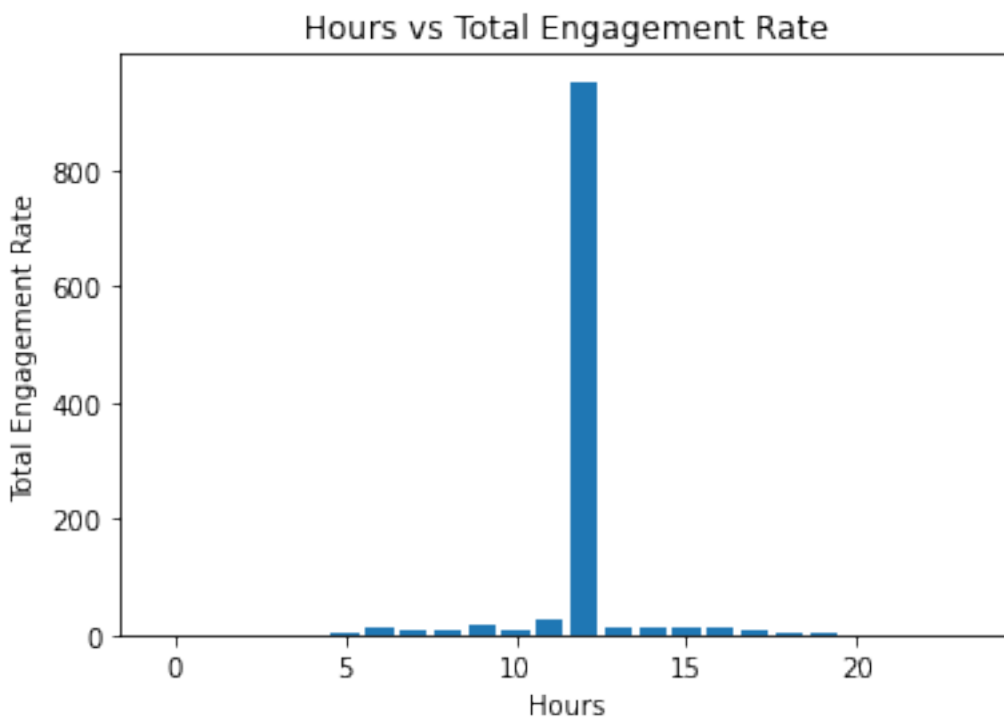

Commentary:

The above histogram shows time does affect engagement rates as well. The 12 o'clock hour has significantly higher engagement rates than all the other hours of the day. With this data in mind, the best time to post social media activity would be Friday around 12 pm.

```
[8]: #Question 3
     #Game titles
     #Use .value_counts() to see the different game titles first
```

```
social_data['Account'].value_counts()
```

[8]:
```
General            2271
DOTA2               803
CSGO                270
Valorant             60
Content Creators     53
General              22
Name: Account, dtype: int64
```

Commentary:

I notice there are repeating "General" Account categories with 2 separate counts. After analyzing, I realize there are different inputs for "General" which are "General" and "General" with a space at the end. The function counts these separately so I will account for this.

[15]:
```python
generalcount=0
dota2count=0
csgocount=0
valorantcount=0
contentcreatorscount=0
for i in range(len(social_data)):
    if social_data.iat[i,1]=='General':
        generalcount+=social_data.iat[i,7]
    elif social_data.iat[i,1]=='General ':
        generalcount+=social_data.iat[i,7]
    elif social_data.iat[i,1]=='DOTA2':
        dota2count+=social_data.iat[i,7]
    elif social_data.iat[i,1]=='CSGO':
        csgocount+=social_data.iat[i,7]
    elif social_data.iat[i,1]=='Valorant':
        valorantcount+=social_data.iat[i,7]
    elif social_data.iat[i,1]=='Content Creators':
        contentcreatorscount+=social_data.iat[i,7]
accounts=['General','DOTA2','CSGO','Valorant','Content Creators']
accountcount=[generalcount,dota2count,csgocount,valorantcount,contentcreatorscount]
plt.bar(accounts,accountcount)
plt.xlabel('Accounts')
plt.ylabel('Total Engagement Rate')
plt.title('Accounts vs Total Engagement Rate')
plt.show()
```

Accounts vs Total Engagement Rate
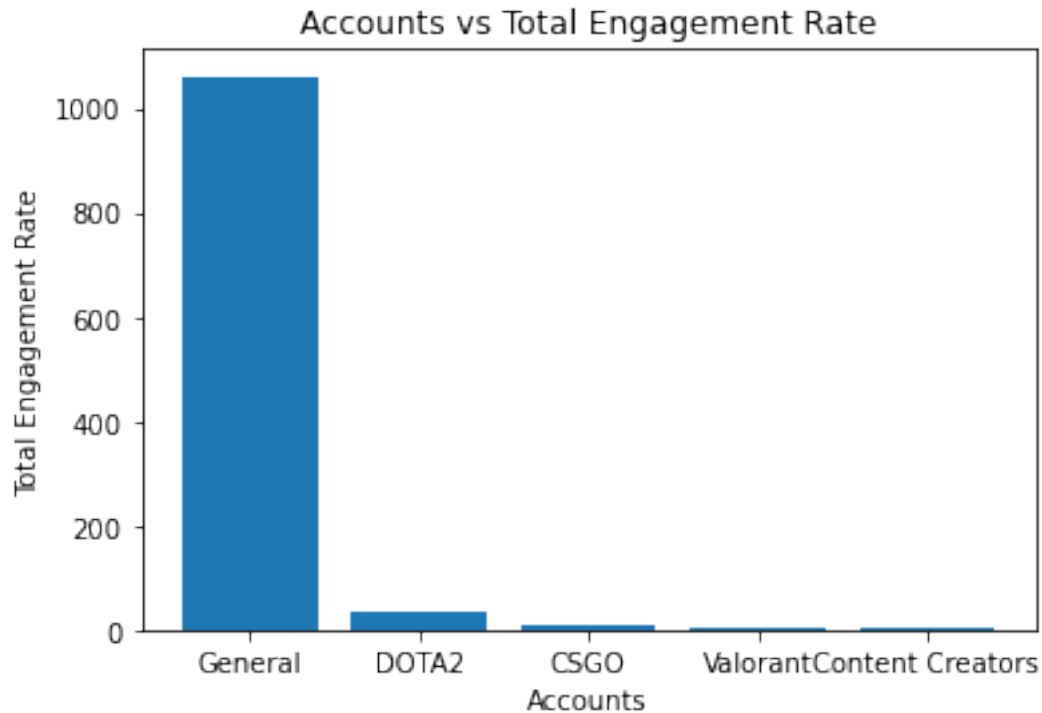
Commentary:

The "General" game title is doing the best in terms of social performance. If Evil Geniuses is solely focused on profits in the short term, they should prioritize posts related to the "General" category. However, if Evil Geniuses wants to grow as an organization in the long term, they should seek to grow engagement in the other accounts like DOTA2, CSGO, Valorant, and Content Creators to diversify engagement and profit. For example, Evil Geniuses could host a giveaway in Valorant from Evil Geniuses' pro players will which bring more attention to Evil Geniuses' Valorant account and perhaps viewers will stay if the giveaways are consistent or they come to like something about the players or team.

```python
#Question 4
#Media
social_data['Media Type'].value_counts()
```

```
[17]: Photo       1490
      Video        967
      Text         910
      Link          94
      Carousel       9
      Mixed          5
      Album          4
      Name: Media Type, dtype: int64
```

```
[24]: photocount=0
      videocount=0
      textcount=0
      linkcount=0
      carouselcount=0
      mixedcount=0
      albumcount=0
      for i in range(len(social_data)):
          if social_data.iat[i,6]=='Photo':
              photocount+=social_data.iat[i,7]
          elif social_data.iat[i,6]=='Video':
              videocount+=social_data.iat[i,7]
          elif social_data.iat[i,6]=='Text':
              textcount+=social_data.iat[i,7]
          elif social_data.iat[i,6]=='Link':
              linkcount+=social_data.iat[i,7]
          elif social_data.iat[i,6]=='Carousel':
              carouselcount+=social_data.iat[i,7]
          elif social_data.iat[i,6]=='Mixed':
              mixedcount+=social_data.iat[i,7]
          elif social_data.iat[i,6]=='Album':
              albumcount+=social_data.iat[i,7]
      mediatype=['Photo','Video','Text','Link','Carousel','Mixed','Album']
      mediatypecount=[photocount,videocount,textcount,linkcount,carouselcount,mixedcount,albumcount]
      plt.bar(mediatype,mediatypecount)
      plt.xlabel('Media Type')
      plt.ylabel('Total Engagement Rate')
      plt.title('Media Type vs Total Engagement Rate')
      plt.show()
```

Media Type vs Total Engagement Rate
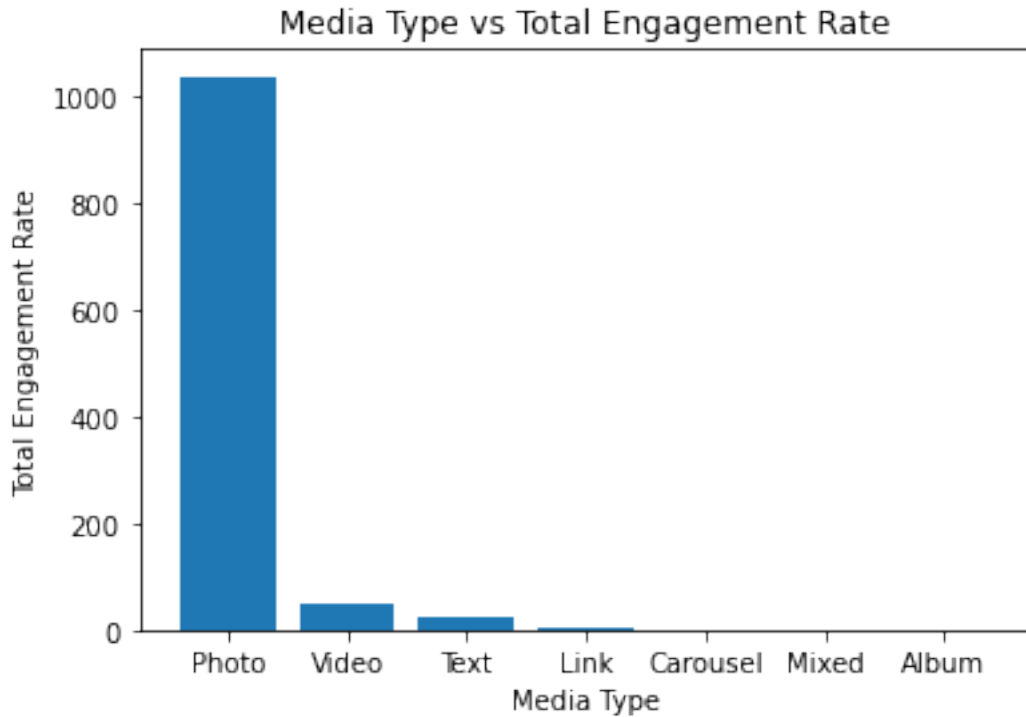
Commentary:
The "Photo" media type performs the best.

```
[26]: #Question 5
      #Campaign
      social_data['Campaign Name'].value_counts()
```

```
[26]: N/A                      1485
      Community Engagement     1411
      Evil Exhibited            420
      Evergreen                 163
      Name: Campaign Name, dtype: int64
```

```
[34]: nacount=0
      communityengagementcount=0
      evilexhibitedcount=0
      evergreencount=0
      for i in range(len(social_data)):
          if social_data.iat[i,3]=='N/A ':
              nacount+=social_data.iat[i,7]
          elif social_data.iat[i,3]=='Community Engagement ':
              communityengagementcount+=social_data.iat[i,7]
          elif social_data.iat[i,3]=='Evil Exhibited ':
              evilexhibitedcount+=social_data.iat[i,7]
```

```
    elif social_data.iat[i,3]=='Evergreen ':
        evergreencount+=social_data.iat[i,7]
campaigntype=['N/A','Community Engagement','Evil Exhibited','Evergreen']
campaigntypecount=[nacount,communityengagementcount,evilexhibitedcount,evergreencount]
plt.bar(campaigntype,campaigntypecount)
plt.xlabel('Campaign Type')
plt.ylabel('Total Engagement Rate')
plt.title('Campaign Type vs Total Engagement Rate')
plt.show()
```
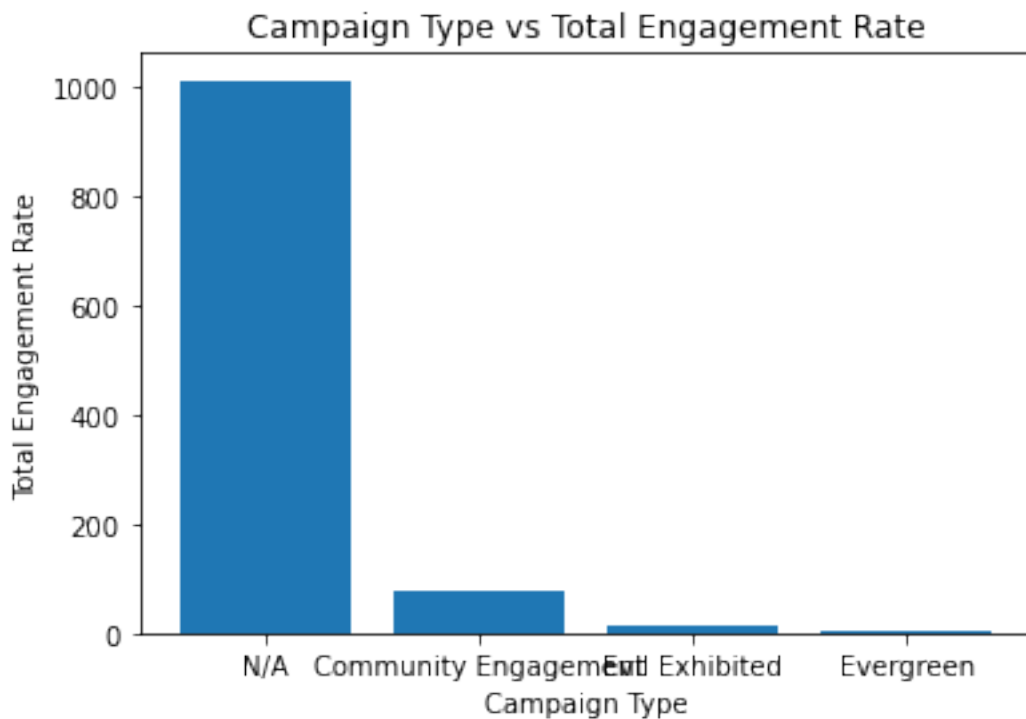

Commentary:
The 'N/A' campaign type is the best performing campaign.

```
[36]:   #Question 6
        #Account
        social_data['Account Type'].value_counts()
```

```
[36]:   TWITTER            1951
        INSTAGRAM           588
        FBPAGE              585
        YOUTUBE             220
        TIKTOK_BUSINESS     113
        LINKEDIN_COMPANY     22
        Name: Account Type, dtype: int64
```

```
[38]: twittercount=0
      instagramcount=0
      facebookcount=0
      youtubecount=0
      tiktokcount=0
      linkedincount=0
      for i in range(len(social_data)):
          if social_data.iat[i,2]=='TWITTER':
              twittercount+=social_data.iat[i,7]
          elif social_data.iat[i,2]=='INSTAGRAM':
              instagramcount+=social_data.iat[i,7]
          elif social_data.iat[i,2]=='FBPAGE':
              facebookcount+=social_data.iat[i,7]
          elif social_data.iat[i,2]=='YOUTUBE':
              youtubecount+=social_data.iat[i,7]
          elif social_data.iat[i,2]=='TIKTOK_BUSINESS':
              tiktokcount+=social_data.iat[i,7]
          elif social_data.iat[i,2]=='LINKEDIN_COMPANY':
              linkedincount+=social_data.iat[i,7]
      accounttype=['TWITTER','INSTAGRAM','FBPAGE','YOUTUBE','TIKTOK_BUSINESS','LINKEDIN_COMPANY']
      accounttypecount=[twittercount,instagramcount,facebookcount,youtubecount,tiktokcount,linkedinc
      plt.bar(accounttype,accounttypecount)
      plt.xlabel('Account Type')
      plt.ylabel('Total Engagement Rate')
      plt.title('Account Type vs Total Engagement Rate')
      plt.show()
```
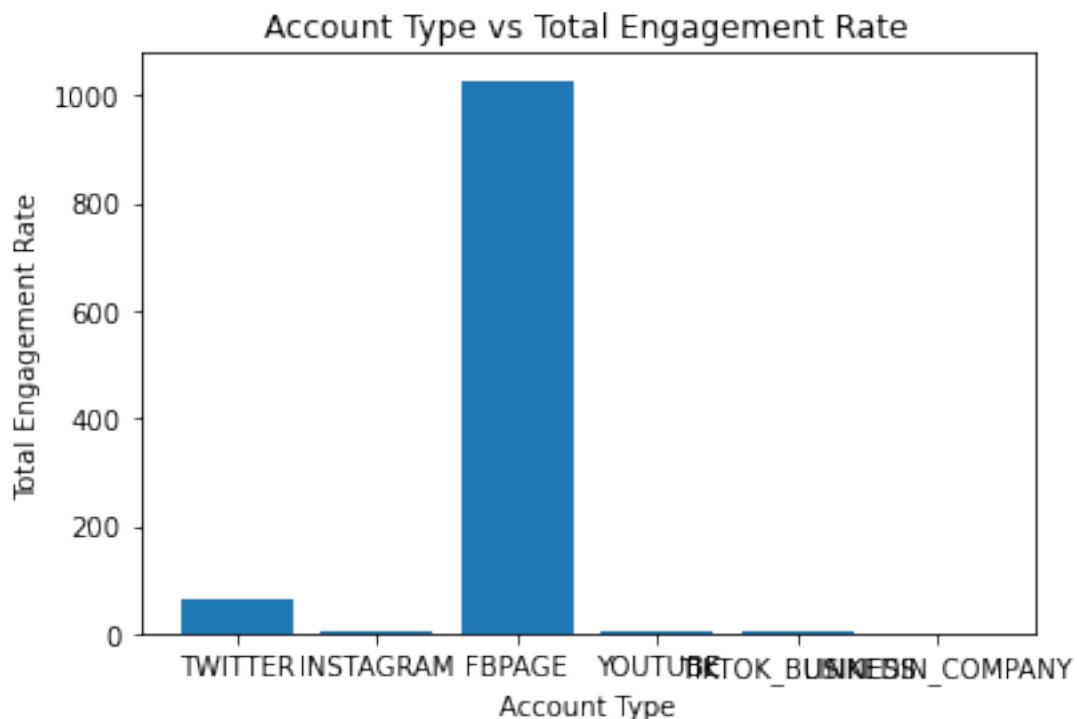
Commentary:

The Facebook page account type has the highest engagement rate.

Commentary:

Based on my discoveries, I would advise the following posting strategy: weigh the more popular variable types more but don't neglect the unpopular variable types. Continue to be consistent with the most highly engaged variables to maximize profit and keep it at high engagement rate. Specifically, continue to post photos of the "General" game type under the "N/A" campaign on Facebook on Fridays at 12 pm. At the same time, if Evil Geniuses wants to grow its organization name under every game type, it needs to diversify its social media channels discoveries. So while the most highly engaged variables should be posted more on, the other social channels, games, and campaigns also need to have some activity. This is not only to grow in all aspects but what if something bad suddenly happens to Facebook or the "General" game category? Then Evil Geniuses won'y have other sources to rely on. By maintaining and seeking to grow the less popular categories, Evil Geniuses can hedge against this possible risk. But, as we see from the data, solely posting won't grow these variables. Evil Geniuses

Question 7

Commentary:

As we see from the data, solely posting won't grow the other variables because they are not getting much engagement to begin with. Thus the social media team needs to have activity outside just posting to bring engagment to the other accounts. For example, post about a Valorant giveaway on Instagram or Twitter because these are all less popular variables. Just by word of mouth and reposting, more and more people will see and become engaged with these accounts. In addition, because the "General" game is doing so well but social media accounts, besides Facebook, are not, perhaps the social media team could post about the "General" game category to these social accounts to attract users who may not use Facebook but use other social media.