Vikram Singh

Capstone Project 1: Milestone Report

In-Depth Analysis of Youtube Trending Videos

The purpose of this project is to analyze trending videos on Youtube with the purpose of providing knowledge to companies regarding where to focus advertising and promotions on Youtube. In the current era, Youtube has become more popular than television, and has become a massive platform for advertising, and promotions of every kind. By looking at characteristics of videos that go trending and then subsequently analyzing the metrics of those videos such as likes, dislikes, comments, title, time/date, and more, insight can be gained into the way videos are received by viewers, the content that trends frequently, and the types of videos that receive the most attention. This knowledge can potentially reveal unseen insights to videos as a whole on Youtube that wouldn't be apparent otherwise.

Upon finding this dataset, it was pretty clear that the data was pretty clean and didn't require much work. I simply had to test drive the data to gain an idea of what I will need to do. The data didn't need cleaning due to outliers and NAN's or organization due to the highly diverse nature of youtube's trending videos. I started by sorted the data into a few of the most popular content creators, and found relevant summary statistics of several of the variables, which is further detailed in my exploratory data analysis.

The only missing data that appears to be found in the data set in a column of interest is the "description" column which I didn't replace or alter in anyway, as the description of a video is a field that can intentionally be left blank when a video is uploaded. There are a surprising number of values that differ drastically from the mean and median, with several observations having a value of 0 for variables which have means/medians above 1000000.

In my exploratory data analysis, I began by searching for anything of interest I could find in the dataset. I noticed early on that there were a lot of trending videos that had very low quantities for views, likes, dislikes, and more. This was very surprising to me, as one would expect a video trending on youtube to be a popular video. Upon further analysis:

```
In [6]: print("There are", views[views == 0].count(), "trending videos with 0 views.")
        print("There are", likes[likes == 0].count(), "trending videos with 0 likes.")
        print("There are", dislikes[dislikes == 0].count(), "trending videos with 0 dislikes.")
        print("There are", comments[comments == 0].count(), "trending videos with 0 comments.")
        print("There are", desc.isnull().sum(), "trending videos with no description.")

        There are 0 trending videos with 0 views.
        There are 172 trending videos with 0 likes.
        There are 383 trending videos with 0 dislikes.
        There are 760 trending videos with 0 comments.
        There are 570 trending videos with no description.
```

There were several videos with values of 0 for key variables, indicating that some videos which go trending aren't met with any extra attention, and perhaps not all trending videos are given equal exposure. Although there are no videos which have 0 views, for a video to have no ratings or comments implies a very low number of views and lack of interest for that video, in general.This prompted me to check these variables and analyze the lowest and highest values, and in every case the range of values was extremely high.

```
In [9]: #Outlier/Range Analysis for Views
        print(x['views'].sort_values().head(10))
        print(x['views'].sort_values().tail(10))

        14335    549
        14563    554
        14782    559
        14531    658
        546      687
        777      704
        14750    713
        14984    745
        12716    748
        160      773
        Name: views, dtype: int64
        36710    179045286
        36913    184446490
        37123    190950401
        37333    196222618
        37531    200820941
        37730    205643016
        37935    210338856
        38146    217750076
        38345    220490543
        38547    225211923
        Name: views, dtype: int64
```

In [10]:
```
#Interesting spread here, there are SEVERAL videos with 0 likes that went trending, yet also many that hav
e 5 million +
print(x['likes'].sort_values().head(10))
print(x['likes'].sort_values().tail(10))
```

```
1490     0
14869    0
1868     0
23516    0
16303    0
22388    0
19093    0
16316    0
16324    0
3621     0
Name: likes, dtype: int64
36397    5053329
36611    5150831
36816    5232318
37031    5321402
37247    5386959
37453    5439015
37655    5486349
37861    5530568
38072    5595203
38273    5613827
Name: likes, dtype: int64
```

In [11]:
```
#Similar interesting trend to the likes
#Would be interesting if a video went trending that 0 people rated
print(x['dislikes'].sort_values().head(10))
print(x['dislikes'].sort_values().tail(10))
```

```
16762    0
10934    0
10931    0
1578     0
10917    0
16973    0
10907    0
1589     0
10902    0
10748    0
Name: dislikes, dtype: int64
10415    1278887
5236     1353647
10638    1415777
5452     1470383
10862    1517520
5699     1545015
5935     1602383
11096    1611043
6181     1643059
11323    1674420
Name: dislikes, dtype: int64
```

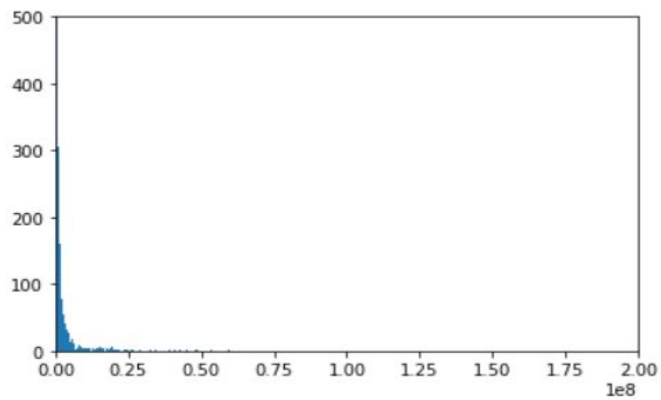
In [12]:
```
#Similar interesting trend of many videos with no comments.
print(x['comment_count'].sort_values().head(10))
print(x['comment_count'].sort_values().tail(10))
```

```
31474    0
20022    0
2362     0
5385     0
35241    0
5390     0
34228    0
2337     0
8313     0
2334     0
Name: comment_count, dtype: int64
10638    1194249
37453    1197130
37655    1204867
37861    1213172
38072    1225326
38273    1228655
10415    1238817
10862    1281094
11096    1321281
11323    1361580
Name: comment_count, dtype: int64
```

The analysis of the extreme values for these variables revealed many videos that received a lot of attention with millions of ratings, alongside many that received none. The presence of many entries with such low values, was certainly important to keep in mind moving forward, as I began to explore the most popular videos in the dataset. A look at the distribution of views reveals that the distribution is extremely skewed to the right, indicating most trending videos actually have a lower quantity of views. This is shown below.

In [27]:
```
plt.hist(views, bins = 10000)
plt.axis([0, 200000000, 0, 500])
```

Out[27]: [0, 200000000, 0, 500]



To gauge who the content creators I should keep an eye on while performing my analysis, I wanted to see which channels were featured the most in the dataset. Most of the below channels are considered mainstream, and it's no surprise that they appear the most frequently.

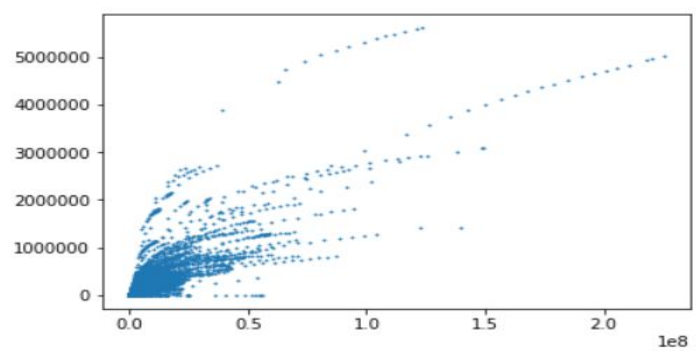
```
In [7]: #Which channels have the most trending videos on Youtube?
        x['channel_title'].value_counts().head(20)

Out[7]: ESPN                                      203
        The Tonight Show Starring Jimmy Fallon    197
        Netflix                                   193
        TheEllenShow                              193
        Vox                                       193
        The Late Show with Stephen Colbert        187
        Jimmy Kimmel Live                         186
        Late Night with Seth Meyers               183
        Screen Junkies                            182
        NBA                                       181
        CNN                                       180
        Saturday Night Live                       175
        WIRED                                     171
        BuzzFeedVideo                             169
        INSIDER                                   167
        The Late Late Show with James Corden      163
        TED-Ed                                    162
        Tom Scott                                 159
        WWE                                       157
        CollegeHumor                              156
        Name: channel_title, dtype: int64
```

The next step was to determine the level of correlation between variables, which proved to reveal many insights that are somewhat intuitive. Given that the amount of views a video received is the most important metric when discussing the attention a video received along with its reach, I began by checking the correlation of views with other variables. The correlation between views and likes was the strongest, with a strong R of .849.

```
In [18]: #There appears to be a generally linear relationship here, as one would expect.
         plt.scatter(views, likes, s = 1)
         np.corrcoef(views, likes)

Out[18]: array([[1.        , 0.84917652],
                [0.84917652, 1.        ]])
```

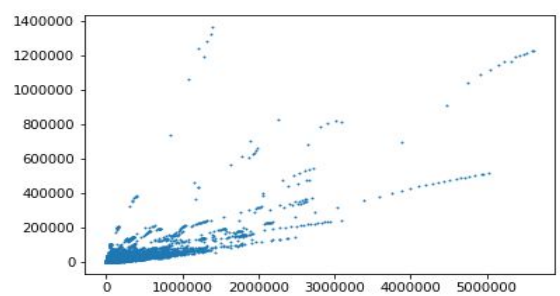


There was a much smaller correlation between views and dislikes, with R = 0.4722. This is expected as videos that are received negatively will generally not receive as much attention. The correlation of views and comments comes out to be R = 0.617, which can be considered expected as more opinions and individual messages would be provided on videos which gain more attention. The correlation between likes and dislikes was rather low, standing at R = 0.447, indicating videos usually remain highly liked or highly disliked, and usually not a combination of the two.

An interesting correlation was present between likes and comments, which is furthered by the correlation between dislikes and comments. Both of these combinations showed rather high correlations, indicating that videos which are received very positively and negatively prompt the public to express sentiment. However, between the two the correlation is higher between likes and comments, indicating people are more likely to express sentiment on videos that are well received. This following scatter plots indicate this trend:

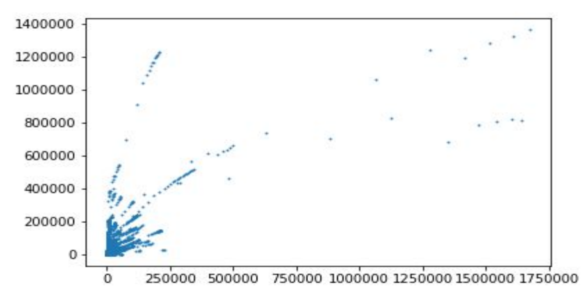
In [22]:
```
#Seems pretty linear for the most part, which makes sense as you'd expect people to
#comment on videos they like.
plt.scatter(likes, comments, s = 1)
np.corrcoef(likes, comments)
```

Out[22]:
```
array([[1.        , 0.80305686],
       [0.80305686, 1.        ]])
```



In [23]:
```
#This is interesting.
plt.scatter(dislikes, comments, s = 1)
np.corrcoef(dislikes, comments)
```

Out[23]:
```
array([[1.        , 0.70018362],
       [0.70018362, 1.        ]])
```



Analysis by category, in conjunction with above key variables, along with a predictive model will be the next steps.