Activity_Course 2 TikTok project lab

August 26, 2023

1 TikTok Project

Goals: 1. Acquaint you with the data

2. Compile summary information about the data
3. Begin the process of EDA and reveal insights contained in the data
4. Prepare you for more in-depth EDA, hypothesis testing, and statistical analysis

Part 1: Understand the situation * How can you best prepare to understand and organize the provided TikTok information?

Part 2: Understand the data

- Create a pandas dataframe for data learning and future exploratory data analysis (EDA) and statistical activities
- Compile summary information about the data to inform next steps

Part 3: Understand the variables

- Use insights from your examination of the summary data to guide deeper investigation into variables

1.0.1 Imports and data loading

[1]:
```
import pandas as pd

import numpy as np
```

[2]:
```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

1.0.2 Understand the data - Inspect the data

Question 1: When reviewing the first few rows of the dataframe, what do you observe about the data? What does each row represent?

Question 2: When reviewing the `data.info()` output, what do you notice about the different variables? Are there any null values? Are all of the variables numeric? Does anything else stand out?

Question 3: When reviewing the `data.describe()` output, what do you notice about the distributions of each variable? Are there any questionable values? Does it seem that there are outlier values?

[3]:
```python
data.head(10)
#Each row represents the claim number, its status, an assosiated video ID and␣
↪its length, the transcript text of the video, its verified and banned␣
↪status, and it s view, like, share, download, and comment counts.
```

[3]:
```
    # claim_status    video_id  video_duration_sec  \
0   1        claim  7017666017                  59
1   2        claim  4014381136                  32
2   3        claim  9859838091                  31
3   4        claim  1866847991                  25
4   5        claim  7105231098                  19
5   6        claim  8972200955                  35
6   7        claim  4958886992                  16
7   8        claim  2270982263                  41
8   9        claim  5235769692                  50
9  10        claim  4660861094                  45

                            video_transcription_text verified_status  \
0  someone shared with me that drone deliveries a…    not verified
1  someone shared with me that there are more mic…    not verified
2  someone shared with me that american industria…    not verified
3  someone shared with me that the metro of st. p…    not verified
4  someone shared with me that the number of busi…    not verified
5  someone shared with me that gross domestic pro…    not verified
6  someone shared with me that elvis presley has …    not verified
7  someone shared with me that the best selling s…    not verified
8  someone shared with me that about half of the …    not verified
9  someone shared with me that it would take a 50…        verified

  author_ban_status  video_view_count  video_like_count  video_share_count  \
0      under review          343296.0           19425.0              241.0
1            active          140877.0           77355.0            19034.0
2            active          902185.0           97690.0             2858.0
3            active          437506.0          239954.0            34812.0
4            active           56167.0           34987.0             4110.0
5      under review          336647.0          175546.0            62303.0
6            active          750345.0          486192.0           193911.0
7            active          547532.0            1072.0               50.0
8            active           24819.0           10160.0             1050.0
9            active          931587.0          171051.0            67739.0
```

```
   video_download_count  video_comment_count
0                   1.0                  0.0
1                1161.0                684.0
2                 833.0                329.0
3                1234.0                584.0
4                 547.0                152.0
5                4293.0               1857.0
6                8616.0               5446.0
7                  22.0                 11.0
8                  53.0                 27.0
9                4104.0               2540.0
```

[4]:
```python
data.info()
#There are 298 rows with missing claim status, and video view, like, share,
 ↪download, and comment data. This represents 1.5% of the data so it might be
 ↪best during the cleaning stage to remove those rows.
#View, like, share, download, and comment are Float values rather than Int
 ↪values even though those values are clearly integers.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19382 entries, 0 to 19381
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   #                         19382 non-null  int64
 1   claim_status              19084 non-null  object
 2   video_id                  19382 non-null  int64
 3   video_duration_sec        19382 non-null  int64
 4   video_transcription_text  19084 non-null  object
 5   verified_status           19382 non-null  object
 6   author_ban_status         19382 non-null  object
 7   video_view_count          19084 non-null  float64
 8   video_like_count          19084 non-null  float64
 9   video_share_count         19084 non-null  float64
 10  video_download_count      19084 non-null  float64
 11  video_comment_count       19084 non-null  float64
dtypes: float64(5), int64(3), object(4)
memory usage: 1.8+ MB
```

[5]:
```python
data.describe()
#Mean video duration is 32.4 seconds, the STDV is high at 16.2
#The shortest video is 5 seconds.
#The longest is 60 (this makes sense because TikTok limited videos to 60
 ↪seconds at this time, so it might be interesting to compare the distribution
 ↪of video lengths).
```

```
#Mean views, likes, shares, downloads, and comments are all lower than their␣
↪STDVs, indicating that there are some extreme upper end outliers to be␣
↪investigated. Perhaps there is a way to divide the data into subsets based␣
↪on their view counts.
#This is reinforced by the fact that many of the max values are 100x larger␣
↪than their medians
```

[5]:

```
                  #      video_id  video_duration_sec  video_view_count  \
count  19382.000000  1.938200e+04        19382.000000      19084.000000
mean    9691.500000  5.627454e+09           32.421732     254708.558688
std     5595.245794  2.536440e+09           16.229967     322893.280814
min        1.000000  1.234959e+09            5.000000         20.000000
25%     4846.250000  3.430417e+09           18.000000       4942.500000
50%     9691.500000  5.618664e+09           32.000000       9954.500000
75%    14536.750000  7.843960e+09           47.000000     504327.000000
max    19382.000000  9.999873e+09           60.000000     999817.000000

       video_like_count  video_share_count  video_download_count  \
count      19084.000000       19084.000000          19084.000000
mean       84304.636030       16735.248323           1049.429627
std       133420.546814       32036.174350           2004.299894
min            0.000000           0.000000              0.000000
25%          810.750000         115.000000              7.000000
50%         3403.500000         717.000000             46.000000
75%       125020.000000       18222.000000           1156.250000
max       657830.000000      256130.000000          14994.000000

       video_comment_count
count         19084.000000
mean            349.312146
std             799.638865
min               0.000000
25%               1.000000
50%               9.000000
75%             292.000000
max            9599.000000
```

1.0.3 Understand the data - Investigate the variables

The ultimate objective is to use machine learning to classify videos as either claims or opinions. Examine the `claim_status` variable by determining how many videos there are for each different claim status:

[6]:

```
pd.Series(data['claim_status']).value_counts()
#The number of claims and opinions are similar and there are 298 null values as␣
↪previously noted.
```

[6]:
```
claim      9608
opinion    9476
Name: claim_status, dtype: int64
```

Using a Boolean masking to filter the data according to claim status, then calculate the mean and median view counts for each claim status:

[7]:
```python
# What is the average view count of videos with "claim" status?
claim_mask = data[data.claim_status == 'claim']
claim_av = np.average(claim_mask['video_view_count'])
claim_med = np.median(claim_mask['video_view_count'])
print('Claim views mean:',claim_av,'Claim views median:', claim_med, sep="\n")
#The average and median are very similar. This indicates that the views counts␣
↪of videos with claims are likely to be evenly distributed.
```

```
Claim views mean:
501029.4527477102
Claim views median:
501555.0
```

[8]:
```python
# What is the average view count of videos with "opinion" status?
op_mask = data[data.claim_status == 'opinion']
op_av = np.average(op_mask['video_view_count'])
op_med = np.median(op_mask['video_view_count'])
print('Opinion views mean:',op_av,'Opinion views median:', op_med, sep="\n")
#The average and median are very similar. This indicates that the views counts␣
↪of videos with claims are likely to be evenly distributed.
#Claim videos receive an average of 100x as many views as opinion videos.
```

```
Opinion views mean:
4956.43224989447
Opinion views median:
4953.0
```

Use `groupby()` to calculate how many videos there are for each combination of categories of claim status and author ban status.

[9]:
```python
# Get counts for each group combination of claim status and author ban status
data.groupby(['claim_status','author_ban_status']).size()
#Banned authors who make claims have a much higher percentage of their category␣
↪than banned others who present opinions.
#Authors who make claims are more likely to make claims that are innacurate and␣
↪claims that are innacurate are more likely to result in a ban.
```

[9]:
```
claim_status  author_ban_status
claim         active               6566
              banned               1439
              under review         1603
opinion       active               8817
```

```
                banned                 196
                under review           463
dtype: int64
```

Calculate the median video share count of each author ban status.

[10]:
```python
ban_mask = data[data.author_ban_status == 'banned']
active_mask = data[data.author_ban_status == 'active']
ur_mask = data[data.author_ban_status == 'under review']

ban_med = np.nanmedian(ban_mask['video_share_count'])
active_med = np.nanmedian(active_mask['video_share_count'])
ur_med = np.nanmedian(ur_mask['video_share_count'])

print('Banned median:',ban_med,'Active median:',active_med,'Under review median:
↪',ur_med, sep="\n")
```

```
Banned median:
14468.0
Active median:
437.0
Under review median:
9444.0
```

[11]:
```python
# What's the median video share count of each author ban status?
#Banned: 14468
#Active: 437
#Under review: 9444
#The median share count of banned authors is massive in comparison to active␣
↪authors.
```

Use `groupby()` to group the data by `author_ban_status`, then use `agg()` to get the count, mean, and median of each of the following columns: * `video_view_count` * `video_like_count` * `video_share_count`

[12]:
```python
ban_views = data.groupby(['author_ban_status']).
↪agg(TotalViews=('video_view_count','sum'),MeanViews=('video_view_count','mean'),MedViews=('

ban_likes = data.groupby(['author_ban_status']).
↪agg(TotalLikes=('video_like_count','sum'),MeanLikes=('video_like_count','mean'),MedLikes=('

ban_shares = data.groupby(['author_ban_status']).
↪agg(TotalShares=('video_share_count','sum'),MeanShares=('video_share_count','mean'),MedShar

print(ban_views, ban_likes, ban_shares, sep='\n\n')
#Active authors have about 50% more views and likes than banned authors and 5x␣
↪more shares than banned authors.
```

```
#Banned authors have significantly higher mean and median views, likes, and␣
↪shares than active authors though.
#Specifically, banned authors have massive mean and median shares compared to␣
↪active authors.
#Perhaps those who watch videos from banned authors are more likely to share␣
↪videos.
```

```
                     TotalViews      MeanViews  MedViews
author_ban_status
active              3.321606e+09  215927.039524    8616.0
banned              7.289573e+08  445845.439144  448201.0
under review        8.102952e+08  392204.836399  365245.5

                     TotalLikes      MeanLikes   MedLikes
author_ban_status
active              1.092755e+09   71036.533836    2222.0
banned              2.501832e+08  153017.236697  105573.0
under review        2.659315e+08  128718.050339   71204.5

                   TotalShares    MeanShares  MedShares
author_ban_status
active              217076684.0  14111.466164      437.0
banned               49048271.0  29998.942508    14468.0
under review         53250524.0  25774.696999     9444.0
```

Create three new columns to help better understand engagement rates: * `likes_per_view`: represents the number of likes divided by the number of views for each video * `comments_per_view`: represents the number of comments divided by the number of views for each video * `shares_per_view`: represents the number of shares divided by the number of views for each video

[13]:
```
# Create a likes_per_view column
data['LPV'] = data['video_like_count']/data['video_view_count']

# Create a comments_per_view column
data['CPV'] = data['video_comment_count']/data['video_view_count']

# Create a shares_per_view column
data['SPV'] = data['video_share_count']/data['video_view_count']
```

Use `groupby()` to compile the information in each of the three newly created columns for each combination of categories of claim status and author ban status, then use `agg()` to calculate the count, the mean, and the median of each group.

[14]:
```
ban_LPV = data.groupby(['author_ban_status','claim_status']).
↪agg(TotalLPV=('LPV','sum'),MeanLPV=('LPV','mean'),MedLPV=('LPV','median'))
#LikesPerView
ban_CPV = data.groupby(['author_ban_status','claim_status']).
↪agg(TotalCPV=('CPV','sum'),MeanCPV=('CPV','mean'),MedCPV=('CPV','median'))
```

```python
#CommentsPerView
ban_SPV = data.groupby(['author_ban_status','claim_status']).
 ↪agg(TotalSPV=('SPV','sum'),MeanSPV=('SPV','mean'),MedSPV=('SPV','median'))
#SharesPerView
print(ban_LPV, ban_CPV, ban_SPV, sep='\n\n')
#Likes, commends, and share are more common for claims than opions, but over␣
 ↪10x as likely for banned users.
#It appears that authors who are banned got a much more significant portion of␣
 ↪their engagement from claims than from opinions.
```

```
                                 TotalLPV   MeanLPV    MedLPV
author_ban_status claim_status
active            claim         2163.772970  0.329542  0.326538
                  opinion       1937.478628  0.219744  0.218330
banned            claim          496.556528  0.345071  0.358909
                  opinion         40.546207  0.206868  0.198483
under review      claim          525.778643  0.327997  0.320867
                  opinion        104.820594  0.226394  0.228051

                              TotalCPV   MeanCPV    MedCPV
author_ban_status claim_status
active            claim         9.144001  0.001393  0.000776
                  opinion       4.559119  0.000517  0.000252
banned            claim         1.981775  0.001377  0.000746
                  opinion       0.085135  0.000434  0.000193
under review      claim         2.191442  0.001367  0.000789
                  opinion       0.247962  0.000536  0.000293

                                TotalSPV   MeanSPV    MedSPV
author_ban_status claim_status
active            claim         429.782704  0.065456  0.049279
                  opinion       385.554840  0.043729  0.032405
banned            claim          97.698624  0.067893  0.051606
                  opinion         7.944022  0.040531  0.030728
under review      claim         105.370751  0.065733  0.049967
                  opinion        20.590731  0.044472  0.035027
```

Executive summary for this TikTok project

Here is an initial summary of the findings in the TikTok data set which could impact the ability to create a model for predicting whether a video contains a claim or an opinion.

There are 298 rows with missing claim status, and video view, like, share, download, and comment data. This represents 1.5% of the data so it might be best during the cleaning stage to remove those rows or impute 0s. Number, video ID, and video duration are integers even though video duration could be a float value, but it seems that video duration was only measured to the nearest second. It is not indicated what type of rounding was used. Claim, transcript, verified, and banned are all object values. Claim has 'claim' and 'opinion', verified has 'verified' and 'not verified', and banned has 'banned', 'active', and 'under review.' View, like, share, download, and comment are

Float values rather than Int values even though those values are clearly integers.

The mean views, likes, and shares of active authors are all many times larger than their medians, indicating a large amount of skew in the data for active authors. The mean likes and shares of banned authors are also 2-3x as large as their medians, indicating some skew in the data for banned authors. When broken down into a per view basis, likes are more likely to be normally distributed, comments are much more likely to be skewed, and shares are somewhat likely to be skewed. Skew measurements for each of these might provide more insight about the types of tests or analysis tools could be used to make accurate predictions given their varying levels of skew.

Video durations have a minimum of 5 seconds and a maximum of 60 as per TikToks previous video length restrictions. This might impact reliability of models built off this data since videos are now allowed to be longer. Video view counts range the most from 20 to almost 1 million with a heavy skew to the right Like, comment, download, and share values all range from 0 to 650k, 250k, 15k, and 10k respectively and all of these are heavily skewed to the right.

Specifically with respect to the variables in question: claims and opinions: The average and median views of videos that are claims are approximately 501k and 502k respectively. The average and median views of videos that are opinions are approximately 4956 and 4953 respectively.

I would recommend further exploration before beginning EDA. A lot of insight could be gained about the way to explore the data more specifically by gathering more descriptive statistics, especially about skew and kurtosis since those seem to be prevalent throughout the data set. Every interval category has a STDV larger than its mean, so it is likely worth the time to trim outliers from either end of this dataset so that it's more likely for analysis like hypothesis testing to be effective. I don't expect any of these categories to become normally distributed, but they almost definitely have extreme outliers that would warp any model that could be created based on the data as it is.

Additional data about Banned authors who posted Opinions could be a relevant part of a model as that is the main intersection with a small amount of values (less than 200).

Another piece of data that would be useful if it's accessible is the watch time for each video. Data about how long a user focused on each type of video might be an indicator the users impression of whether or not the video contains a claim and therefore could be a usable predictor in the model.

Data about how many followers each author had when the video was posted could also help in case the concept of "viralness" comes into the model.

[ ]:
```

```