Activity_Course 2 TikTok project lab

January 27, 2024

1 TikTok Project

Course 2 - Get Started with Python

Welcome to the TikTok Project!

You have just started as a data professional at TikTok.

The team is still in the early stages of the project. You have received notice that TikTok's leadership team has approved the project proposal. To gain clear insights to prepare for a claims classification model, TikTok's provided data must be examined to begin the process of exploratory data analysis (EDA).

A notebook was structured and prepared to help you in this project. Please complete the following questions.

2 Course 2 End-of-course project: Inspect and analyze data

In this activity, you will examine data provided and prepare it for analysis.

The purpose of this project is to investigate and understand the data provided. This activity will:

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

The goal is to construct a dataframe in Python, perform a cursory inspection of the provided dataset, and inform TikTok data team members of your findings. *This activity has three parts:*

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

To complete the activity, follow the instructions and answer the questions below. Then, you will us your responses to these questions and the questions included in the Course 2 PACE Strategy Document to create an executive summary.

Be sure to complete this activity before moving on to Course 3. You can assess your work by comparing the results to a completed exemplar after completing the end-of-course project.

3 Identify data types and compile summary information

Throughout these project notebooks, you'll see references to the problem-solving framework PACE. The following notebook components are labeled with the respective PACE stage: Plan, Analyze, Construct, and Execute.

4 PACE stages

- `[Plan](#scrollTo=psz51YkZVwtN&line=3&uniqifier=1)`
- `[Analyze](#scrollTo=mA7Mz_SnI8km&line=4&uniqifier=1)`
- `[Construct](#scrollTo=Lca9c8XON8lc&line=2&uniqifier=1)`
- `[Execute](#scrollTo=401PgchTPr4E&line=2&uniqifier=1)`

4.1 PACE: Plan

Consider the questions in your PACE Strategy Document and those below to craft your response:

4.1.1 Task 1. Understand the situation

- How can you best prepare to understand and organize the provided information?

Begin by exploring your dataset and consider reviewing the Data Dictionary.

Reading the data directory gives me a very good understanding of what the dataset should look like ideally. Also reading the description of the deliverables. And the emails sent to me regarding the project

4.2 PACE: Analyze

Consider the questions in your PACE Strategy Document to reflect on the Analyze stage.

4.2.1 Task 2a. Imports and data loading

Start by importing the packages that you will need to load and explore the dataset. Make sure to use the following import statements: * `import pandas as pd`

- `import numpy as np`

[3]:
```
import pandas as pd
import numpy as np
```

[4]:
```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

4.2.2 Task 2b. Understand the data - Inspect the data

View and inspect summary information about the dataframe by **coding the following:**

1. `data.head(10)`
2. `data.info()`
3. `data.describe()`

Consider the following questions:

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?

Then, load the dataset into a dataframe. Creating a dataframe will help you conduct data manipulation, exploratory data analysis (EDA), and statistical activities.

Note: As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

[36]:
```
data.head(10)
```

[36]:
```
   # 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  likes_per_view  comments_per_view  shares_per_view  \
0      under review        0.056584           0.000000         0.000702
1            active        0.549096           0.004855         0.135111
2            active        0.108282           0.000365         0.003168
3            active        0.548459           0.001335         0.079569
4            active        0.622910           0.002706         0.073175
5      under review        0.521454           0.005516         0.185069
6            active        0.647958           0.007258         0.258429
7            active        0.001958           0.000020         0.000091
8            active        0.409364           0.001088         0.042306
9            active        0.183612           0.002727         0.072714

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

[37]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19382 entries, 0 to 19381
Data columns (total 15 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   likes_per_view            19084 non-null  float64
 8   comments_per_view         19084 non-null  float64
 9   shares_per_view           19084 non-null  float64
 10  video_view_count          19084 non-null  float64
 11  video_like_count          19084 non-null  float64
 12  video_share_count         19084 non-null  float64
 13  video_download_count      19084 non-null  float64
 14  video_comment_count       19084 non-null  float64
dtypes: float64(8), int64(3), object(4)
memory usage: 2.2+ MB
```

[38]:
```
data.describe()
```

[38]:
```
                  #      video_id  video_duration_sec  likes_per_view  \
count  19382.000000  1.938200e+04        19382.000000    19084.000000
mean    9691.500000  5.627454e+09           32.421732        0.276093
std     5595.245794  2.536440e+09           16.229967        0.173006
min        1.000000  1.234959e+09            5.000000        0.000000
25%     4846.250000  3.430417e+09           18.000000        0.130240
50%     9691.500000  5.618664e+09           32.000000        0.264037
75%    14536.750000  7.843960e+09           47.000000        0.398482
max    19382.000000  9.999873e+09           60.000000        0.666648

       comments_per_view  shares_per_view  video_view_count  video_like_count  \
count       19084.000000     19084.000000      19084.000000      19084.000000
mean            0.000954         0.054860     254708.558688      84304.636030
std             0.001326         0.050597     322893.280814     133420.546814
min             0.000000         0.000000         20.000000          0.000000
```

```
25%              0.000098          0.014445      4942.500000       810.750000
50%              0.000455          0.039739      9954.500000      3403.500000
75%              0.001268          0.081864    504327.000000    125020.000000
max              0.010280          0.265956    999817.000000    657830.000000

       video_share_count  video_download_count  video_comment_count
count       19084.000000          19084.000000         19084.000000
mean        16735.248323           1049.429627           349.312146
std         32036.174350           2004.299894           799.638865
min             0.000000              0.000000             0.000000
25%           115.000000              7.000000             1.000000
50%           717.000000             46.000000             9.000000
75%         18222.000000           1156.250000           292.000000
max        256130.000000          14994.000000          9599.000000
```

data.head(10) The first few lines of data, makes light of few things. All the first five entries are claims. So that would be worth exploring wether theres some sort of bias, or there should be some sort of randomisation or other sorting and filtering. Its also worth noting that the first line has no video_comments, indicating that comments were disabled. There seems to be little correaltion between how many times a video is viewed and shared.

data.info() There first apparent observeration is the number of total rows and the non-null values. There seems to be an connection between 298 rows and null values. There is a mix of datatypes, both, ints, floats and objects. Objects for “verified_status” and “author_ban_status” could possibly be booleans instead since theyd take less space and memory? The number(#), could do with a more descpitive name, rather than a special character.

data.describe() “video_like_count”,“video_share_count”, “video_download_count”, “video_comment_count” seems likes columns worth investigating to establish wether the 0 value is a and outlier worth filtering out or they’re all relevant. Also the range of values for these fields are very wide, and would also indicate theres something in the date obscuring the view. They also have means that are very close to the 75% percentile, futher implying that the data in the current state is not giving the whole picture

“video_view_count” has an everage of 254708, but the less looking at the quantiles it suggest that a few videos are increasing the average. Is it worth using a median here to compare?

All the objects are missing due to not being possible to to numerical operatiosn on them. But the 3 of them could be boolean instead, which would make it alot easier to gain insight without compromising the data

4.2.3 Task 2c. Understand the data - Investigate the variables

In this phase, you will begin to investigate the variables more closely to better understand them.

You know from the project proposal that the ultimate objective is to use machine learning to classify videos as either claims or opinions. A good first step towards understanding the data might therefore be examining the `claim_status` variable. Begin by determining how many videos there are for each different claim status.

[33]:
```
print(data.groupby("claim_status")["claim_status"].count())
claim_num = (data.groupby("claim_status")["claim_status"].count()) / (data.
↪groupby("claim_status")["claim_status"].count().sum())

print(data.groupby("claim_status")["claim_status"].count().sum())


print((len(data)) == (data.groupby("claim_status")["claim_status"].count().
↪sum()))
print(claim_num * 100)
```

```
claim_status
claim      9608
opinion    9476
Name: claim_status, dtype: int64
19084
False
claim_status
claim      50.345839
opinion    49.654161
Name: claim_status, dtype: float64
```

There are rows missing their claim_status, as previously established. Apart from that, they're very equally split

Next, examine the engagement trends associated with each different claim status.

Start by using Boolean masking to filter the data according to claim status, then calculate the mean and median view counts for each claim status.

[35]:
```
mask_claim = data['claim_status'] == "claim"
mask_opinion = data[('claim_status')] == "opinion"



data[mask_opinion]
data[mask_claim]
```

[35]:
```
         # 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
…     …            …           …                   …
9603  9604        claim  3883493316                  49
9604  9605        claim  4765029942                   9
9605  9606        claim  3513102998                  27
9606  9607        claim  9461481859                  27
```

```
9607  9608        claim  1622115206                16

                               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
…                                                  …             …
9603  a colleague discovered on the radio a claim th…    not verified
9604  a colleague discovered on the radio a claim th…        verified
9605  a colleague discovered on the radio a claim th…    not verified
9606  a colleague discovered on the radio a claim th…    not verified
9607  a colleague discovered on the radio a claim th…    not verified

     author_ban_status  likes_per_view  comments_per_view  shares_per_view  \
0         under review        0.056584           0.000000         0.000702
1               active        0.549096           0.004855         0.135111
2               active        0.108282           0.000365         0.003168
3               active        0.548459           0.001335         0.079569
4               active        0.622910           0.002706         0.073175
…                  …             …                  …              …
9603            active        0.625010           0.004574         0.073999
9604            active        0.658297           0.004446         0.145060
9605      under review        0.236556           0.000897         0.050011
9606            active        0.278612           0.001393         0.058910
9607            banned        0.195480           0.001627         0.058388

      video_view_count  video_like_count  video_share_count  \
0             343296.0           19425.0              241.0
1             140877.0           77355.0            19034.0
2             902185.0           97690.0             2858.0
3             437506.0          239954.0            34812.0
4              56167.0           34987.0             4110.0
…                  …                …                 …
9603          737177.0          460743.0            54550.0
9604          546987.0          360080.0            79346.0
9605          885521.0          209475.0            44286.0
9606          356747.0           99394.0            21016.0
9607          114288.0           22341.0             6673.0

      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
```

```
…                    …                   …
9603            8119.0              3372.0
9604            4537.0              2432.0
9605            1210.0               794.0
9606            1163.0               497.0
9607             284.0               186.0

[9608 rows x 15 columns]
```

[25]:
```python
# What is the average view count of videos with "opinion" status?

(data[mask_opinion]["video_view_count"]).mean()

data[mask_opinion].describe()
#data[mask_claim].describe()
```

[25]:
```
                  #      video_id  video_duration_sec  likes_per_view  \
count   9476.000000  9.476000e+03         9476.000000             0.0
mean   14346.500000  5.622382e+09           32.359856             NaN
std     2735.629909  2.530209e+09           16.281705             NaN
min     9609.000000  1.234959e+09            5.000000             NaN
25%    11977.750000  3.448802e+09           18.000000             NaN
50%    14346.500000  5.611857e+09           32.000000             NaN
75%    16715.250000  7.853243e+09           47.000000             NaN
max    19084.000000  9.999835e+09           60.000000             NaN

       video_view_count  video_like_count  video_share_count  \
count       9476.000000       9476.000000        9476.000000
mean        4956.432250       1092.729844         217.145631
std         2885.907219        964.099816         252.269583
min           20.000000          0.000000           0.000000
25%         2467.000000        289.000000          34.000000
50%         4953.000000        823.000000         121.000000
75%         7447.250000       1664.000000         314.000000
max         9998.000000       4375.000000        1674.000000

       video_download_count  video_comment_count
count           9476.000000          9476.000000
mean              13.677290             2.697446
std               16.200652             4.089288
min                0.000000             0.000000
25%                2.000000             0.000000
50%                7.000000             1.000000
75%               19.000000             3.000000
max              101.000000            32.000000
```

Question: What do you notice about the mean and media within each claim category? The average views are much higher for opinions than claims 501029 vs 4956

Now, examine trends associated with the ban status of the author.

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

[34]:
```
data.groupby(['claim_status','author_ban_status'])['video_id'].count()
```

[34]:
```
claim_status  author_ban_status
claim         active               6566
              banned               1439
              under review         1603
opinion       active               8817
              banned                196
              under review          463
Name: video_id, dtype: int64
```

Question: What do you notice about the number of claims videos with banned authors? Why might this relationship occur?

The claims category have much higher numbers in the category of banned and under review. They also have lower number of active users.

Continue investigating engagement levels, now focusing on `author_ban_status`.

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

[ ]:
```

```

[41]:
```
# What's the median video share count of each author ban status?
data.groupby(['author_ban_status'])['video_share_count'].median()
```

[41]:
```
author_ban_status
active              437.0
banned            14468.0
under review       9444.0
Name: video_share_count, dtype: float64
```

Question: What do you notice about the share count of banned authors, compared to that of active authors? Explore this in more depth.

The median video_share_count is much higher for banned users than active users.

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`

Remember, the argument for the `agg()` function is a dictionary whose keys are columns. The values for each column are a list of the calculations you want to perform.

[40]:
```
data.
↪groupby(['author_ban_status'])["video_view_count","video_like_count","video_share_count"].
↪agg(['count', 'mean', 'median'])
```

[40]:
```
                  video_view_count                          video_like_count  \
                             count           mean    median            count
author_ban_status
active                       15383  215927.039524    8616.0            15383
banned                        1635  445845.439144  448201.0             1635
under review                  2066  392204.836399  365245.5             2066

                                          video_share_count                \
                           mean    median             count          mean
author_ban_status
active             71036.533836    2222.0             15383  14111.466164
banned            153017.236697  105573.0              1635  29998.942508
under review      128718.050339   71204.5              2066  25774.696999


                    median
author_ban_status
active               437.0
banned             14468.0
under review        9444.0
```

Question: What do you notice about the number of views, likes, and shares for banned authors compared to active authors? Banned users are more popular in view_count, like_count and video_share. Almost by double compared to active ones. Even the under review status, is more popular than the active ones. but by average and median.

Now, 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

[6]:
```
# Create a likes_per_view column
data.insert(7,"likes_per_view",(data['video_like_count'] /␣
↪data['video_view_count']))

# Create a comments_per_view column
data.insert(8,"comments_per_view",(data['video_comment_count'] /␣
↪data['video_view_count']))

# Create a shares_per_view column
data.insert(9,"shares_per_view",(data['video_share_count'] /␣
↪data['video_view_count']))
```

```
␣
↪---------------------------------------------------------------------------

    ValueError                                Traceback (most recent call␣
↪last)

    <ipython-input-6-5c8583dd377e> in <module>
      1 # Create a likes_per_view column
----> 2 data.insert(7,"likes_per_view",(data['video_like_count'] /␣
↪data['video_view_count']))
      3
      4 # Create a comments_per_view column
      5 data.insert(8,"comments_per_view",(data['video_comment_count'] /␣
↪data['video_view_count']))


    /opt/conda/lib/python3.7/site-packages/pandas/core/frame.py in␣
↪insert(self, loc, column, value, allow_duplicates)
   4412         if not allow_duplicates and column in self.columns:
   4413             # Should this be a different kind of error??
-> 4414             raise ValueError(f"cannot insert {column}, already␣
↪exists")
   4415         if not isinstance(loc, int):
   4416             raise TypeError("loc must be int")


    ValueError: cannot insert likes_per_view, already exists
```

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.

[25]:
```
data.
↪groupby(['claim_status','author_ban_status'])['comments_per_view',"likes_per_view","shares_
↪agg(['median','mean','count'])
```

[25]:
```
                              comments_per_view                   \
                                         median      mean count
claim_status author_ban_status
claim        active                    0.000776  0.001393  6566
             banned                    0.000746  0.001377  1439
             under review              0.000789  0.001367  1603
opinion      active                    0.000252  0.000517  8817
             banned                    0.000193  0.000434   196
             under review              0.000293  0.000536   463
```

claim_status	author_ban_status	likes_per_view median	likes_per_view mean	likes_per_view count	shares_per_view median	shares_per_view mean	shares_per_view count
claim	active	0.326538	0.329542	6566	0.049279	0.065456	6566
	banned	0.358909	0.345071	1439	0.051606	0.067893	1439
	under review	0.320867	0.327997	1603	0.049967	0.065733	1603
opinion	active	0.218330	0.219744	8817	0.032405	0.043729	8817
	banned	0.198483	0.206868	196	0.030728	0.040531	196
	under review	0.228051	0.226394	463	0.035027	0.044472	463

Question:

How does the data for claim videos and opinion videos compare or differ? Consider views, comments, likes, and shares. Claim videos generate more response from the audience. Both in terms sharing, likes and comments.

4.3 PACE: Construct

Note: The Construct stage does not apply to this workflow. The PACE framework can be adapted to fit the specific requirements of any project.

4.4 PACE: Execute

Consider the questions in your PACE Strategy Document and those below to craft your response.

4.4.1 Given your efforts, what can you summarize for Rosie Mae Bradshaw and the TikTok data team?

Note for Learners: Your answer should address TikTok's request for a summary that covers the following points:

- What percentage of the data is comprised of claims and what percentage is comprised of opinions?
- What factors correlate with a video's claim status?
- What factors correlate with a video's engagement level?

The percentage of claim 50.34 % and opinion 49.65%

What what seems to be a correlation in the video's engagement is the notoriety. The Claims that come from users that have been banned or are under reviewd, seem to be causing a lot of user generated traffic, shares, likes and comments.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.