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.

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.

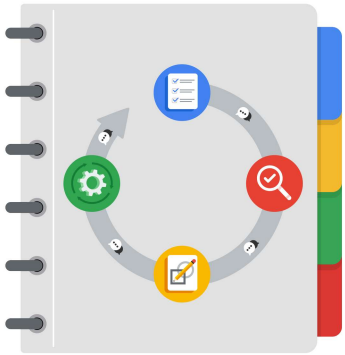
Be sure to complete this activity before moving on to Course 3. You can assess your work by comparing the results to a completed exemplar after completing the end-of-course project.

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.

PACE stages

- `[Plan]`
- `[Analyze]`
- `[Construct]`
- `[Execute]`



PACE: Plan

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

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.

Complete an initial investigation of the data by reviewing the data dictionary. This provides a high-level overview of the types of variables included in the dataset and their specific types. Then, conduct a more in-depth investigation of the dataset and generate descriptive statistics.

PACE: Analyze

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

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`

In [2]:
```
# Import packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

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.

In [3]:
```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

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?

- Each row represents a single TikTok video that's made it to production.

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?

- Seven of the columns appear to have missing values, though the amount seems insignificant compared to the total number of rows. Four of the variables are objects, while the remaining are numeric, either integers or floats.

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?

- The data appears to contain some outliers. Several variables exhibit significant right-skew, while others have more uniform distributions.

In [48]:
```
# Display and examine the first ten rows of the dataframe
data.head(10)
```

Out[48]:

	#	claim_status	video_id	video_duration_sec	video_transcription_text	verified_statu
0	1	claim	7017666017	59	someone shared with me that drone deliveries a...	not verifie
1	2	claim	4014381136	32	someone shared with me that there are more mic...	not verifie
2	3	claim	9859838091	31	someone shared with me that american industria...	not verifie
3	4	claim	1866847991	25	someone shared with me that the metro of st. p...	not verifie
4	5	claim	7105231098	19	someone shared with me that the number of busi...	not verifie
5	6	claim	8972200955	35	someone shared with me that gross domestic pro...	not verifie
6	7	claim	4958886992	16	someone shared with me that elvis presley has ...	not verifie
7	8	claim	2270982263	41	someone shared with me that the best selling s...	not verifie
8	9	claim	5235769692	50	someone shared with me that about half of the ...	not verifie
9	10	claim	4660861094	45	someone shared with me that it would take a 50...	verifie

In [5]:
```
# Get summary info
data.info()
```

```
<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
```

In [6]:
```
# Get summary statistics
data.describe()
```

Out[6]:

	#	video_id	video_duration_sec	video_view_count	video_like_count
count	19382.000000	1.938200e+04	19382.000000	19084.000000	19084.000000
mean	9691.500000	5.627454e+09	32.421732	254708.558688	84304.636030
std	5595.245794	2.536440e+09	16.229967	322893.280814	133420.546814
min	1.000000	1.234959e+09	5.000000	20.000000	0.000000
25%	4846.250000	3.430417e+09	18.000000	4942.500000	810.750000
50%	9691.500000	5.618664e+09	32.000000	9954.500000	3403.500000
75%	14536.750000	7.843960e+09	47.000000	504327.000000	125020.000000
max	19382.000000	9.999873e+09	60.000000	999817.000000	657830.000000

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.

In [7]:
```
# What are the different values for claim status and how many of each are in the da
print(data['claim_status'].value_counts())
print()
print(data['claim_status'].value_counts(normalize=True))
```

```
claim_status
claim      9608
opinion    9476
Name: count, dtype: int64

claim_status
claim      0.503458
opinion    0.496542
Name: proportion, dtype: float64
```

Question: What do you notice about the values shown?

- The 'claim_status' variable shows a balanced distribution between its categories. This being the case, any class balancing will likely not need to occur.

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.

In [8]:
```
# What is the average view count of videos with "claim" status?
average_claim_view_count = data[data['claim_status'] == 'claim']['video_view_count'
median_claim_view_count = data[data['claim_status'] == 'claim']['video_view_count']

print("Average Claim View Count:", average_claim_view_count)
print("Median Claim View Count:", median_claim_view_count)
```

```
Average Claim View Count: 501029.4527477102
Median Claim View Count: 501555.0
```

In [9]:
```
# What is the average view count of videos with "opinion" status?
average_opinion_view_count = data[data['claim_status'] == 'opinion']['video_view_co
median_opinion_view_count = data[data['claim_status'] == 'opinion']['video_view_cou

print("Average Opinion View Count:", average_opinion_view_count)
print("Median Opinion View Count:", median_opinion_view_count)
```

```
Average Opinion View Count: 4956.43224989447
Median Opinion View Count: 4953.0
```

Question: What do you notice about the mean and median within each claim category?

- The mean and median are relatively close, suggesting a symmetric distribution, such as a uniform or normal distribution.

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.

In [49]:
```
# Get counts for each group combination of claim status and author ban status
grouped_data = data.groupby(['claim_status', 'author_ban_status']).size().unstack()

print(grouped_data)
```

```
author_ban_status  active  banned  under review
claim_status
claim                6566    1439          1603
opinion              8817     196           463
```

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

- There are significantly more authors of videos with claims than opinions. This could indicate the presence of highly controversial claims.

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

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

In [50]:
```
# What's the median video share count of each author ban status?
median_share_count = data.groupby('author_ban_status')['video_share_count'].median(

print(median_share_count)
```

```
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.

- There are significantly more videos shared by banned authors than by active authors. Active authors who follow the guidelines are far less likely to upload videos containing highly contentious material, reducing their risk of being banned. The large number of videos from banned authors could be due to misinformation campaigns or groups deliberately trying to spread misinformation.

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.

In [12]:
```
# Group by 'author_ban_status' and aggregate 'video_view_count', 'video_like_count'
view_count_stats = data.groupby('author_ban_status')['video_view_count'].agg(['mean
like_count_stats = data.groupby('author_ban_status')['video_like_count'].agg(['mean
share_count_stats = data.groupby('author_ban_status')['video_share_count'].agg(['me

# Print the results
print("Video View Count:\n", view_count_stats)
print("\nVideo Like Count:\n", like_count_stats)
print("\nVideo Share Count:\n", share_count_stats)
```

```
Video View Count:
                            mean    median  count
author_ban_status
active              215927.039524    8616.0  15383
banned              445845.439144  448201.0   1635
under review        392204.836399  365245.5   2066

Video Like Count:
                            mean    median  count
author_ban_status
active               71036.533836    2222.0  15383
banned              153017.236697  105573.0   1635
under review        128718.050339   71204.5   2066

Video Share Count:
                           mean   median  count
author_ban_status
active              14111.466164    437.0  15383
banned              29998.942508  14468.0   1635
under review        25774.696999   9444.0   2066
```

Question: What do you notice about the number of views, likes, and shares for banned authors compared to active authors?

- The data indicates that videos from banned authors receive significantly higher numbers of views, likes, and shares compared to those from active authors. This trend is concerning, as it suggests that potentially harmful or misleading content from banned users is reaching a wider audience.

Next, create three new columns to better understand engagement rates:

- `likes_per_view` : the number of likes divided by the number of views for each video
- `comments_per_view` : the number of comments divided by the number of views for each video
- `shares_per_view` : the number of shares divided by the number of views for each video

In [51]:
```
# Create a likes_per_view column
data['likes_per_view'] = data['video_like_count'] / data['video_view_count']

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

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

# Display the first few rows of the new columns
print(data[['likes_per_view', 'comments_per_view', 'shares_per_view']].head(10))
```

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

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.

In [47]:
```
# Group by 'claim_status' and 'author_ban_status' and aggregate the mean and median
likes_per_view_stats = data.groupby(['claim_status', 'author_ban_status'])['likes_p
comments_per_view_stats = data.groupby(['claim_status', 'author_ban_status'])['comm
shares_per_view_stats = data.groupby(['claim_status', 'author_ban_status'])['shares

# Print the results
print("Likes Per View:\n", likes_per_view_stats)
print("\nComments Per View:\n", comments_per_view_stats)
print("\nShares Per View:\n", shares_per_view_stats)
```

```
Likes Per View:
                                     mean    median  count       min       max
claim_status author_ban_status
claim        active              0.329542  0.326538   6566  0.000000  0.666648
             banned              0.345071  0.358909   1439  0.000259  0.666116
             under review        0.327997  0.320867   1603  0.000815  0.665468
opinion      active              0.219744  0.218330   8817  0.000000  0.444409
             banned              0.206868  0.198483    196  0.001802  0.444303
             under review        0.226394  0.228051    463  0.000000  0.443316

Comments Per View:
                                     mean    median  count  min       max
claim_status author_ban_status
claim        active              0.001393  0.000776   6566  0.0  0.010280
             banned              0.001377  0.000746   1439  0.0  0.009162
             under review        0.001367  0.000789   1603  0.0  0.010049
opinion      active              0.000517  0.000252   8817  0.0  0.004673
             banned              0.000434  0.000193    196  0.0  0.003032
             under review        0.000536  0.000293    463  0.0  0.003552

Shares Per View:
                                     mean    median  count       min       max
claim_status author_ban_status
claim        active              0.065456  0.049279   6566  0.000000  0.265956
             banned              0.067893  0.051606   1439  0.000023  0.255618
             under review        0.065733  0.049967   1603  0.000015  0.257914
opinion      active              0.043729  0.032405   8817  0.000000  0.175632
             banned              0.040531  0.030728    196  0.000000  0.156641
             under review        0.044472  0.035027    463  0.000000  0.164691
```

Question:

How does the data for claim videos and opinion videos compare or differ? Consider views, comments, likes, and shares.

- Overall, there are more videos from banned authors categorized as claims compared to opinions. Additionally, opinion videos are less likely to result in bans than claim videos. Claim videos, on average, have higher engagement metrics (views, comments, likes, and shares) and are likely more harmful as they reach a wider audience.

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.

PACE: Execute

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

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?

- Approximately 50.35% of the data consists of claims, while 49.65% consists of opinions.

What factors correlate with a video's claim status?

- It appears a video's claim status (opinion or claim) is correlated with engagement; videos that have been categorized as claims appear to reach a wider audience, and are also more likely to be authored by banned authors, indicating the nature of the content contained in claim videos is likely violating terms of service on average, promoting misinformation and/or containing harmful content.

What factors correlate with a video's engagement level?

- Likes, comments, and shares are directly correlated with a video's engagement level. The engineered variables 'likes_per_view', 'shares_per_view', and 'comments_per_view' serve

as engagement metrics, indicating that as these metrics increase, a video's engagement also increases.

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.