gjgazzcea

August 1, 2025

1 Project: Video Tiktok status prediction

1.0.1 By: Lam Gia Bao - HCMUS

[3]:

```
import numpy as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import classification_report, accuracy_score,␣
  ↪precision_score, \
recall_score, f1_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
```

This project uses a dataset called tiktok_dataset.csv. It contains synthetic data created for this project in partnership with TikTok. The data will be investigated, analyzed to prepare for building a simple Machine Learning model.

The dataset contains:

19,383 rows – Each row represents a different published TikTok video in which a claim/opinion has been made.

12 columns:

1. # (int): TikTok assigned number for video with claim/opinion.
2. **claim_status** (obj): Whether the published video has been identified as an "opinion" or a "claim." In this dataset, an "opinion" refers to an individual's or group's personal belief or thought. A "claim" refers to information that is either unsourced or from an unverified source.
3. **video_id** (int): Random identifying number assigned to video upon publication on TikTok.
4. **video_duration_sec** (int): How long the published video is measured in seconds.
5. **video_transcription_text** (obj): Transcribed text of the words spoken in the published video.
6. **verified_status** (obj): Indicates the status of the TikTok user who published the video in terms of their verification, either "verified" or "not verified."

7. **author_ban_status** (obj): Indicates the status of the TikTok user who published the video in terms of their permissions: “active,” “under review,” or “banned.”
8. **video_view_count** (float): The total number of times the published video has been viewed.
9. **video_like_count** (float): The total number of times the published video has been liked by other users.
10. **video_share_count** (float): The total number of times the published video has been shared by other users.
11. **video_download_count** (float): The total number of times the published video has been downloaded by other users.
12. **video_comment_count** (float): The total number of comments on the published video.

[4]:
```
from google.colab import drive
drive.mount("/content/drive")
PATH = "/content/drive/MyDrive/tiktok_dataset.csv"

data = pd.read_csv(PATH).drop(columns=['#','video_id'])
data.head(10)
```

```
Mounted at /content/drive
```

[4]:
```
  claim_status  video_duration_sec  \
0        claim                  59
1        claim                  32
2        claim                  31
3        claim                  25
4        claim                  19
5        claim                  35
6        claim                  16
7        claim                  41
8        claim                  50
9        claim                  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
```

[5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19382 entries, 0 to 19381
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   claim_status              19084 non-null  object
 1   video_duration_sec        19382 non-null  int64
 2   video_transcription_text  19084 non-null  object
 3   verified_status           19382 non-null  object
 4   author_ban_status         19382 non-null  object
 5   video_view_count          19084 non-null  float64
 6   video_like_count          19084 non-null  float64
 7   video_share_count         19084 non-null  float64
 8   video_download_count      19084 non-null  float64
 9   video_comment_count       19084 non-null  float64
dtypes: float64(5), int64(1), object(4)
memory usage: 1.5+ MB
```

[6]:

```
data.describe()
```

[6]:

```
       video_duration_sec  video_view_count  video_like_count  \
count        19382.000000      19084.000000      19084.000000
mean            32.421732     254708.558688      84304.636030
std             16.229967     322893.280814     133420.546814
min              5.000000         20.000000          0.000000
```

```
25%             18.000000       4942.500000       810.750000
50%             32.000000       9954.500000      3403.500000
75%             47.000000     504327.000000    125020.000000
max             60.000000     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
```

[7]:

```
data.isna().sum()
```

[7]:

```
claim_status                298
video_duration_sec            0
video_transcription_text    298
verified_status               0
author_ban_status             0
video_view_count            298
video_like_count            298
video_share_count           298
video_download_count        298
video_comment_count         298
dtype: int64
```

[8]:

```
data = data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 19084 entries, 0 to 19083
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   claim_status              19084 non-null  object
 1   video_duration_sec        19084 non-null  int64
 2   video_transcription_text  19084 non-null  object
 3   verified_status           19084 non-null  object
 4   author_ban_status         19084 non-null  object
 5   video_view_count          19084 non-null  float64
 6   video_like_count          19084 non-null  float64
 7   video_share_count         19084 non-null  float64
 8   video_download_count      19084 non-null  float64
 9   video_comment_count       19084 non-null  float64
```

```
dtypes: float64(5), int64(1), object(4)
memory usage: 1.6+ MB
```

[9]:
```
data.duplicated().sum()
```

[9]:
```
np.int64(0)
```

[10]:
```
cat_cols = list(data.select_dtypes(include=["object"]).columns)
print("Categorical columns: ", cat_cols)
num_cols = list(data.select_dtypes(exclude=["object"]).columns)
print("Numerical columns: ", num_cols)
```

```
Categorical columns:  ['claim_status', 'video_transcription_text',
'verified_status', 'author_ban_status']
Numerical columns:  ['video_duration_sec', 'video_view_count',
'video_like_count', 'video_share_count', 'video_download_count',
'video_comment_count']
```

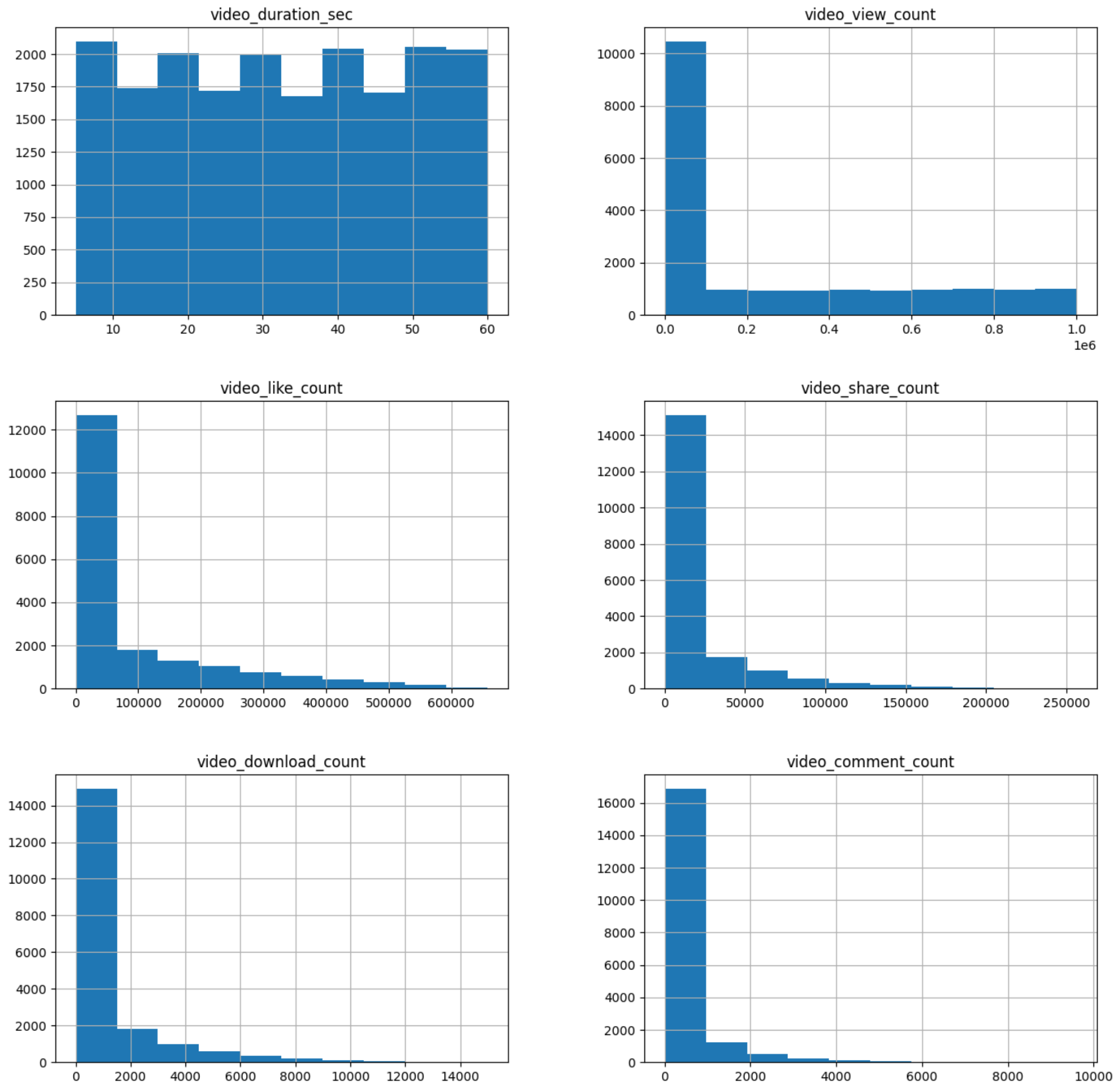
[11]:
```
for i in cat_cols:
  print(data[i].value_counts())
  print('------------------------------')
```

```
claim_status
claim      9608
opinion    9476
Name: count, dtype: int64
------------------------------
video_transcription_text
a colleague learned  from the media a claim that sputnik was the first
artificial satellite in space                        2
a friend read  in the media that badminton is the fastest ball and net sport in
the world                                    2
a colleague learned  from the media a claim that the earliest playable recording
of a person singing was recorded in 1860    2
a colleague read  in the media that earth days are getting longer over time due
to orbital expansion                         2
someone learned  from the media a claim that the japanese word ‘karaoke’ comes
from a phrase meaning ’empty orchestra’       2
                                            ..
a colleague learned  on a website a claim that there are more stars in the
universe than grains of sand on earth              1
a colleague learned  on a website a claim that 5 dwarf planets are recognized in
the solar system                            1
a colleague learned  on a website a claim that saturn is less dense than water
1
a colleague learned  on a website a claim that the moon was once part of the
earth                                            1
a colleague learned  on a website a claim that our sense of smell and taste
```

```
decreases by 20%-50% during airplane flights     1
Name: count, Length: 19012, dtype: int64
------------------------------
verified_status
not verified    17884
verified         1200
Name: count, dtype: int64
------------------------------
author_ban_status
active          15383
under review     2066
banned           1635
Name: count, dtype: int64
------------------------------
```

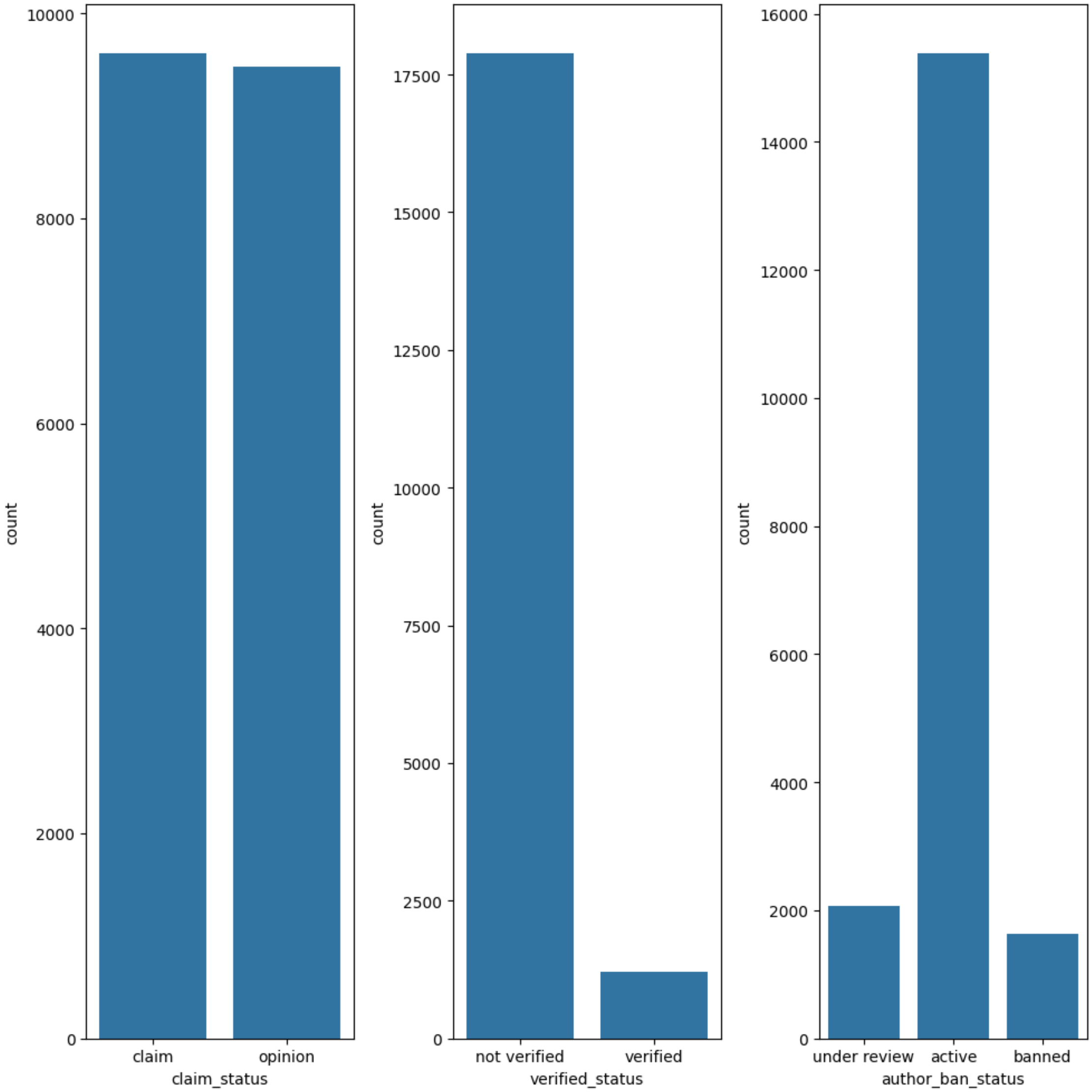
```
[12]: data[num_cols].hist(figsize=(15,15))
```

```
[12]: array([[<Axes: title={'center': 'video_duration_sec'}>,
               <Axes: title={'center': 'video_view_count'}>],
              [<Axes: title={'center': 'video_like_count'}>,
               <Axes: title={'center': 'video_share_count'}>],
              [<Axes: title={'center': 'video_download_count'}>,
               <Axes: title={'center': 'video_comment_count'}>]], dtype=object)
```



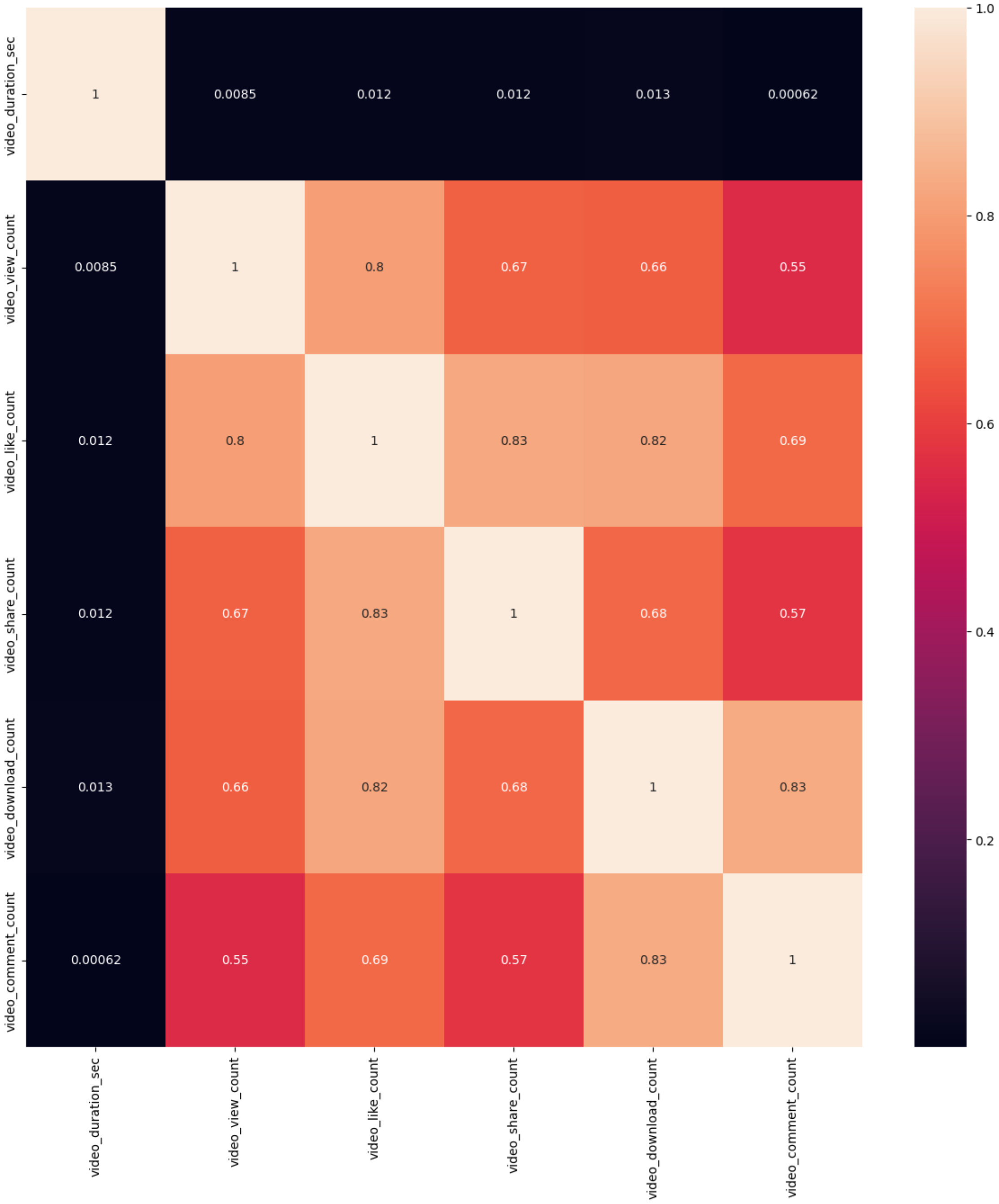
[13]:

```
plt.figure(figsize=(10,10))
col2 = ['claim_status', 'verified_status', 'author_ban_status']
for i, col in enumerate(col2):
  ax = plt.subplot(1,3,i+1)
  sns.countplot(data=data, x=col, ax=ax)
  plt.tight_layout()
plt.show()
```



```
[14]: plt.figure(figsize=(15,15))
      sns.heatmap(data[num_cols].corr(), annot=True)
```

```
[14]: <Axes: >
```



1.1 Statistical Analysis

We will use some statistical methods to analysis the difference between verified and non-verified videos depends on the number of views, likes, shares, downloads and comments.

[15]:
```
data_verified = data[data['verified_status'] == 'verified']
data_not_verified = data[data['verified_status'] == 'not verified']
```

```
video_count =␣
 ↪['video_view_count','video_like_count','video_share_count','video_download_count','video_co
```

[16]:

```
data.groupby('verified_status')[video_count].mean()
```

[16]:

```
                 video_view_count  video_like_count  video_share_count  \
verified_status
not verified        265663.785339      87925.772422       17415.888000
verified             91439.164167      30337.633333        6591.448333

                 video_download_count  video_comment_count
verified_status
not verified              1095.814080           363.700514
verified                   358.146667           134.877500
```

We consider the hypothesis and null-hypothesis:

H_0: There is NO difference about video count between each verified or non-verified videos.

H_1: There are differences about video count between each verified or non-verified videos.

To find the true answer, we'll use the **t-test** method with significance level is $\alpha = 5\% = 0.05$.

[17]:

```
alpha = 0.05
for i in video_count:
  test = stats.ttest_ind(data_verified[i], data_not_verified[i])
  print(test)
  if test.pvalue < alpha:
    print('Reject the null hypothesis')
  else:
    print('Fail to reject the null hypothesis')
```

```
TtestResult(statistic=np.float64(-18.250939509545823),
pvalue=np.float64(8.632160883925904e-74), df=np.float64(19082.0))
Reject the null hypothesis
TtestResult(statistic=np.float64(-14.554067146196873),
pvalue=np.float64(9.91794948296101e-48), df=np.float64(19082.0))
Reject the null hypothesis
TtestResult(statistic=np.float64(-11.3686123831593),
pvalue=np.float64(7.477270020175633e-30), df=np.float64(19082.0))
Reject the null hypothesis
TtestResult(statistic=np.float64(-12.391247887257451),
pvalue=np.float64(3.978848655008078e-35), df=np.float64(19082.0))
Reject the null hypothesis
TtestResult(statistic=np.float64(-9.619069379156674),
pvalue=np.float64(7.44664113555214e-22), df=np.float64(19082.0))
Reject the null hypothesis
```

We also consider the hypothesis and null-hypothesis:

H_0: There is NO difference about video count between each author ban status.

H_1: There are differences about video count between each author ban status.

To find the true answer, we'll use the **t-test** method with significance level is $\alpha = 5\% = 0.05$.

[18]:

```
data_active = data[data['author_ban_status'] == 'active']
data_under_review = data[data['author_ban_status'] == 'under review']
data_banned = data[data['author_ban_status'] == 'banned']
```

[19]:

```
data.groupby('author_ban_status')[video_count].mean()
```

[19]:

```
                   video_view_count  video_like_count  video_share_count  \
author_ban_status
active                215927.039524      71036.533836       14111.466164
banned                445845.439144     153017.236697       29998.942508
under review          392204.836399     128718.050339       25774.696999

                   video_download_count  video_comment_count
author_ban_status
active                       882.276344           295.134499
banned                      1886.296024           614.956575
under review                1631.734753           542.480639
```

[20]:

```
alpha = 0.05
for i in video_count:
  chi2, p_value, dof, expected = stats.chi2_contingency(pd.
 ↪crosstab(data['author_ban_status'], data[i]))
  print(p_value)
  if p_value < alpha:
    print('Reject the null hypothesis')
  else:
    print('Fail to reject the null hypothesis')
```

```
1.2082532937509246e-55
Reject the null hypothesis
6.961666781257757e-233
Reject the null hypothesis
0.0
Reject the null hypothesis
0.0
Reject the null hypothesis
5.256193539998813e-233
Reject the null hypothesis
```

1.2 Feature Engineering

[21]:

```
status = data['claim_status'].replace({'opinion': 0, 'claim': 1}).astype(int)
data2 = data.drop(columns=['claim_status'])
data = pd.get_dummies(data2, columns=['verified_status', 'author_ban_status'],␣
  ↪dtype=int)
data.insert(0, 'claim_status', status)
data.head()
```

```
/tmp/ipython-input-107422386.py:1: FutureWarning: Downcasting behavior in
`replace` is deprecated and will be removed in a future version. To retain the
old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to
the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  status = data['claim_status'].replace({'opinion': 0, 'claim': 1}).astype(int)
```

[21]:

```
   claim_status  video_duration_sec  \
0             1                  59
1             1                  32
2             1                  31
3             1                  25
4             1                  19

                            video_transcription_text  video_view_count  \
0  someone shared with me that drone deliveries a…          343296.0
1  someone shared with me that there are more mic…          140877.0
2  someone shared with me that american industria…          902185.0
3  someone shared with me that the metro of st. p…          437506.0
4  someone shared with me that the number of busi…           56167.0

   video_like_count  video_share_count  video_download_count  \
0           19425.0              241.0                   1.0
1           77355.0            19034.0                1161.0
2           97690.0             2858.0                 833.0
3          239954.0            34812.0                1234.0
4           34987.0             4110.0                 547.0

   video_comment_count  verified_status_not verified  \
0                  0.0                             1
1                684.0                             1
2                329.0                             1
3                584.0                             1
4                152.0                             1

   verified_status_verified  author_ban_status_active  \
0                         0                         0
1                         0                         1
2                         0                         1
```

```
3                          0                          1
4                          0                          1

   author_ban_status_banned  author_ban_status_under review
0                         0                               1
1                         0                               0
2                         0                               0
3                         0                               0
4                         0                               0
```

[22]:
```
data['claim_status'].value_counts()
```

[22]:
```
claim_status
1    9608
0    9476
Name: count, dtype: int64
```

[23]:
```
from sklearn.feature_extraction.text import CountVectorizer
count_vec = CountVectorizer(ngram_range=(2, 3),
                            max_features=15,
                            stop_words='english')
count_vec
```

[23]:
```
CountVectorizer(max_features=15, ngram_range=(2, 3), stop_words='english')
```

[24]:
```
count_data = count_vec.fit_transform(data['video_transcription_text']).toarray()
count_data
count_df = pd.DataFrame(data=count_data, columns=count_vec.
 ↪get_feature_names_out())
df = pd.concat([data.drop(columns=['video_transcription_text']).
 ↪reset_index(drop=True), count_df], axis=1)
df.head()
```

[24]:
```
   claim_status  video_duration_sec  video_view_count  video_like_count  \
0             1                  59          343296.0           19425.0
1             1                  32          140877.0           77355.0
2             1                  31          902185.0           97690.0
3             1                  25          437506.0          239954.0
4             1                  19           56167.0           34987.0

   video_share_count  video_download_count  video_comment_count  \
0              241.0                   1.0                  0.0
1            19034.0                1161.0                684.0
2             2858.0                 833.0                329.0
3            34812.0                1234.0                584.0
4             4110.0                 547.0                152.0
```

```
   verified_status_not verified  verified_status_verified  \
0                             1                         0
1                             1                         0
2                             1                         0
3                             1                         0
4                             1                         0

   author_ban_status_active  …  friend read  internet forum  learned media  \
0                         0  …            0               0              0
1                         1  …            0               0              0
2                         1  …            0               0              0
3                         1  …            0               0              0
4                         1  …            0               0              0

   learned news  media claim  news claim  point view  read media  \
0             0            0           0           0           0
1             0            0           0           0           0
2             0            0           0           0           0
3             0            0           0           0           0
4             0            0           0           0           0

   social media  willing wager
0             0              0
1             0              0
2             0              0
3             0              0
4             0              0

[5 rows x 27 columns]
```

1.3 Build a Random Forest model

[25]:
```python
X = df.drop(columns=['claim_status'])
y = df['claim_status']
trainX, testX, trainY, testY = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
print('Train X shape:', trainX.shape)
print('Train Y shape:', trainY.shape)
print('Test X shape:', testX.shape)
print('Test Y shape:', testY.shape)
```

```
Train X shape: (15267, 26)
Train Y shape: (15267,)
Test X shape: (3817, 26)
Test Y shape: (3817,)
```

[27]:

```
def build_model(X, y):
    rf = RandomForestClassifier(random_state=0)
    cv_params = {'max_depth': [5, 7, None],
                 'max_features': [0.3, 0.6],
                 'max_samples': [0.7],
                 'min_samples_leaf': [1,2],
                 'min_samples_split': [2,3],
                 'n_estimators': [75,100,200],
                 }
    rf_cv = GridSearchCV(rf, cv_params, scoring=['accuracy', 'precision',␣
 ↪'recall', 'f1'], cv=5, refit='recall')
    return rf_cv.fit(X, y)

def calculate_performance(y_true, y_pred):
    print("Precision: ", precision_score(y_true, y_pred))
    print("Recall: ", recall_score(y_true, y_pred))
    print("F1: ", f1_score(y_true, y_pred))
    print("Confusion matrix: \n", confusion_matrix(y_true, y_pred))
    print("Classification report: \n", classification_report(y_true, y_pred))
    main_score = f1_score(y_true, y_pred)
    return main_score

model = build_model(trainX, trainY)
print(model.best_params_)
print(model.best_score_)
pred = model.predict(testX)
calculate_performance(testY, pred)
```

```
{'max_depth': None, 'max_features': 0.3, 'max_samples': 0.7, 'min_samples_leaf':
1, 'min_samples_split': 3, 'n_estimators': 200}
0.9955729166666668
Precision:  1.0
Recall:  0.995850622406639
F1:  0.997920997920998
Confusion matrix:
 [[1889    0]
 [   8 1920]]
Classification report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      1889
           1       1.00      1.00      1.00      1928

    accuracy                           1.00      3817
   macro avg       1.00      1.00      1.00      3817
weighted avg       1.00      1.00      1.00      3817
```

```
[27]: 0.997920997920998
```