COGS 108 - Final Project

Piazza Post @858: This notebook contains interactive plots which might not be displayed under some circumstances. If you can't see any plot, please run all the cells. Thanks!

Overview

We chose to take data gathered from YouTube's United States Trending page and do sentiment analysis on the transcripts gathered from the videos in order to observe what, if any, differences there were between videos from different categories. First, we cleaned the data set to remove repeat instances of the same video showing up on the trending page and videos without transcripts, among other things. We then ran a sentiment analysis algorithm on the transcripts of all of the cleaned data set both line by line per video (line-wise) and for the transcript as a whole (video-wise), which provided us with polarity (positive or negative sentiment) and subjectivity for each video. From there, we grouped the videos based on their categories in order to quantify the differences in sentiment (polarity and subjectivity) between videos from different categories, analyze the results, and compare the results to our hypotheses.

Group Members IDs & Names

- A12326075 Ata Tafazoli Yazdi
- A13809639 Gaotong Wu
- A13565157 Sam Baik
- A15120468 Thomas Lauer
- A15529641 Vas Sengupta
- A13935712 Zihao Zhou

Research Question

What, if any, are the differences in sentiment (measured by polarity and subjectivity) between one category and all the other categories of trending YouTube videos in the United States based on the video transcripts?

Background and Prior Work

Throughout the last decade, we have seen the decline of traditional sources of media in the form of television and radio and the rise of new media in the form of various social media platforms. Video-sharing websites like YouTube have and are growing every day; more and more people are tuning into YouTube channels over television channels for their entertainment, news, and education. What is fascinating about this trend is the increased public access to the growing data on the media content people consume. Thus, we became curious about what we could learn about the content on YouTube people were consuming.

To aid us in our research, we looked at a previous data science project about YouTube and a book on sentiment analysis. In this project (https://towardsdatascience.com/youtube-views-predictor-9ec573090acb), the contributors, Allen Wang, Aravind Srinivasan, Kevin Yee, and Ryan O'Farrell, used Python to try to predict the view count of Fitness videos on YouTube by analyzing the video's title and thumbnail. What they discovered was that "the best predictor of how well [a] channel will do is the number of views [the channel's] previous videos have had." While this project was focused on the factors of video virality, we were more interested in the content itself of trending YouTube videos. What were the characteristics of popular YouTube video content? What made a piece of video content different from another? Other than subject matter, how else do video content differ across YouTube video categories? These questions led us to think about how we could analyze video content on YouTube using data science.

To answer these questions, we needed a method to gain insight from what people were saying in their videos. Though some of us were previously introduced to the concepts and methods of sentiment analysis, we conducted some research to understand the applications of sentiment analysis better. From Bing Liu's book (https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf), we discovered that sentiment analysis had been used in a variety of fields, such as using Twitter sentiment to predict election results and movie box-office revenues. Given the success of these applications in different contexts, we decided to use sentiment analysis on the transcriptions of trending YouTube videos in various categories to help answer our questions. More precisely, we were interested in whether a video category was more or less polar and subjective than all the other categories. Both assessments of sentiment that range from a score of [-1, 1], polarity is a measure of how positive or negative a statement is, and subjectivity is a measure of how objective or opinionated a statement is. According to Liu, "opinions are central to almost all human activities because they are key influencers of our behaviors. Whenever we need to make a decision, we want to know others' opinions". Liu's statement emphasizes the importance of understanding sentiment and highlights why it is important to consider the sentiment of the YouTube video content we all consume. We should expect categories like news and education to be more objective than other categories, or else we may need to question how much we should trust the information we gain from such videos. Our research and these expectations and concerns, therefore, led us to our hypothesis about YouTube video categories.

References (include links):

1) YouTube Views Predictor Data Science Project: https://towardsdatascience.com/youtube-views-predictor-9ec573090acb (https://towardsdatascience.com/youtube-views-predictor-9ec573090acb)

2) Sentiment Analysis Book: https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf (https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf)

Hypothesis

We hypothesize that comedy videos contain more positive sentences than other categories, that news videos contain more objective sentences than other categories and that polarity and subjectivity of sentences in music videos vary more than in other categories. We believe these predictions to be true because we assume that jokes that make people feel more positive are positively worded, news journalists should aim to be as objective as possible, there is a wide range of emotions expressed in the lyrics of songs, and lyrics contain statements that are characterized as both personal feelings or opinion (e.g., "I think we should get back together" and "Think we kissed but I forgot") and as factual (e.g., "Pictures of last night ended up online" and "Glitter all over the room").

Setup

In [1]:
```
import pickle
import csv
import pandas as pd
import plotly.figure_factory as ff
import numpy as np
from textblob import TextBlob

import plotly.offline as py
import plotly.graph_objs as go
import plotly.express as px
import cufflinks
#import emoji
from scipy.stats import mannwhitneyu, ttest_ind, bartlett
cufflinks.go_offline(connected=True)
```

Dataset(s)

Dataset: Trending YouTube Video Statistics

Source: https://www.kaggle.com/datasnaek/youtube-new (https://www.kaggle.com/datasnaek/youtube-new)

`USvideos.csv` is a daily record of the top trending US YouTube videos.

- The dataset contains the information of 40949 non-unique videos, or 6351 unique videos (since some videos are trending for more than one day)
- There are 16 features, however, we mainly focus on a few of them after data cleaning.
  - `video_id`
  - `trending_date` : yy.mm.dd
  - `title`
  - `channel_title`
  - `category_id`
  - `publish_time`
  - `tags` : all tags of the video, space-delimited
  - `views` : view count
  - `likes` : like count
  - `dislikes` : dislike count
  - `comment_count`
  - `thumbnail_link` : https link to the thumbnail image
  - `comments_disabled` : True or False
  - `ratings_disabled` : True or False
  - `video_error_or_removed` : True or False
  - `description`
- We had considered doing sentiment analysis on each video's description, but it turns out that the description is usually not a good representation of the overall sentiment of a video — many descriptions are mainly the links to the uploader's channel or previous videos. It makes more sense to analyze the transcription of those videos rather than the description.

In [2]:
```
df = pd.read_csv('USvideos.csv')
print("There are", np.shape(df)[0], "videos")
print("There are", np.shape(df)[1], "features")
df.head()
```

```
There are 40949 videos
There are 16 features
```

Out[2]:

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	c
0	2kyS6SvSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z	SHANtell martin	748374	57527	2966	
1	1ZAPwfrtAFY	17.14.11	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13T07:30:00.000Z	last week tonight trump presidency|"last week ...	2418783	97185	6146	
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	racist superman|"rudy"|"mancuso"|"king"|"bach"...	3191434	146033	5339	
3	puqaWrEC7tY	17.14.11	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13T11:00:04.000Z	rhett and link|"gmm"|"good mythical morning"|"...	343168	10172	666	
4	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga	24	2017-11-12T18:01:41.000Z	ryan|"higa"|"higatv"|"nigahiga"|"i dare you"|"...	2095731	132235	1989	

Dataset: Categories JSON

Source: https://www.kaggle.com/datasnaek/youtube-new (https://www.kaggle.com/datasnaek/youtube-new)

`US_category_id.json` establishes one-to-one correspondence between `category_id` and name of the category. There are 32 different categories. However, some unpopular categories never shows up in `USvideos.csv` .

In [3]:
```
def load_category():
    category_dataframe = pd.read_json('US_category_id.json')
    clean_df = pd.DataFrame(columns=['id', 'name'])
    for i in range(len(category_dataframe)):
        item = category_dataframe.loc[i]['items']
        clean_df = clean_df.append({'id': int(item['id']), 'name':item['snippet']['title']}, ignore_index=True)
    clean_df = clean_df.set_index('id')
    print('There are', len(clean_df), 'different categories')
    return clean_df

category_df = load_category()
display(category_df.head())
```

```
There are 32 different categories
```

id	name
1	Film & Animation
2	Autos & Vehicles
10	Music
15	Pets & Animals
17	Sports

Dataset: Crawled Transcriptions

We crawled the transcriptions of all 6351 videos using `youtube_transcript_api` .

Source: https://pypi.org/project/youtube-transcript-api/ (https://pypi.org/project/youtube-transcript-api/)

```
from youtube_transcript_api import YouTubeTranscriptApi
import pickle
import csv
import pandas as pd

def getText(item):
    return item['text']

def parseTr(lst):
    return list(map(getText, lst))

df = pd.read_csv('USvideos.csv')
trdict = {}

count = 0

for i in df['video_id']:
    count = count + 1
    try:
        tr = parseTr(YouTubeTranscriptApi.get_transcript(i, languages=['en']))
        trdict[i] = tr
    except:
        trdict[i] = None

    if(count % 1000 == 0):
        pickle.dump(trdict, open('transcription_' + str(count / 1000) + '.p', 'wb'))
        trdict = {}

pickle.dump(trdict, open('transcription_' + str(count) + '.p', 'wb'))
```

`transcription.p` stores a dictionary whose key-value is video_id-transcription(string array). We drop the time information of the transcription, analyzing only the text part.

Concept Explained: Video-wise Analysis

Concatenate all lines of the transcription of a video and then run `textblob` on it. End up getting a 2D point (overall polarity, overall subjectivity) from each video. Further analysis is done by directly observing the distribution of 2D points.

Pros:

- More representative of each video's overall sentiment
- Give insight into whether videos in certain video categories are more positive/neutral/negative/varying than an average video
- Easier visualization

Concept Explained: Line-wise Analysis

Run `textblob` on each line of the transcription of a video. End up getting two distributions (line-wise polarity and subjectivity) from each video. Further analysis is done by adding up normalized distributions.

Pros:

- A closer look at how `textblob` works for short sentences
- Give insight into whether the uploaders in specific video categories tend to say more or less neutral/positive/negative sentence
- Open to more pre-processing techniques (drop very short sentence, drop invalid results, normalization)

In [4]:
```
tr = pickle.load(open('transcription.p', 'rb'))
print("Here is an example transcription: ")
print(list(tr.values())[2][0:10])
```

```
Here is an example transcription:
['[Music]', 'hello confetti club it is fixing and', "today is such an exciting day I'm so", 'freakin excited it is a ve
ry special', 'youtube birthday but not my actual', 'YouTube birthday my actual YouTube', 'birthday is August something
but today', 'is the fateful day I have received my', '100,000 subscribers YouTube play button', "mmm and so I'm going t
o do a cheeky"]
```

Data Cleaning & Pre-processing

After cleaning the data, we combine the original features with the features we get from the transcripts.

There are 12 features now.

- `video_id`
- `title`
- `category_id`
- `views`
- `likes`
- `dislikes`
- `comment_count`
- `transcripts`
- `polarity` : the polarity of all the lines concatenated, for video-wise analysis
- `subjectivity` : the subjectivity of all the lines concatenated, for line-wise analysis
- `line_polarity` : the distribution of the polarity per line over [-1,1], for line-wise analysis
- `line_subjectivity` : the distribution of the subjectivity per line over [0,1], for line-wise analysis

We clean the data by

- Dropping videos without transcription
- Dropping sentences with 0 polarity & 0 subjectivity for line-wise analysis (means fail to analyze)
- Normalizing the distributions for line-wise analysis

In [5]:

```
NUM_BIN = 20

def load_videos():
    # Load all raw dataframes
    video_dataframe = pd.read_csv('USvideos.csv')
    transcript_dict = pickle.load(open('transcription.p', 'rb'))

    clean_df = pd.DataFrame()
    clean_df['video_id'] = video_dataframe['video_id'].unique()
    print('loaded video ids')

    # Linewise anlysis: bins for histogram
    binp = np.linspace(-1.0, 1.0, num=NUM_BIN+1)
    bins = np.linspace( 0.0, 1.0, num=NUM_BIN+1)

    # Process basic video stats
    video_titles = []
    video_category = []
    video_views = []
    video_likes = []
    video_dislikes = []
    video_comment_count = []
    for v_id in clean_df['video_id']:
        current_video = video_dataframe[video_dataframe['video_id'] == v_id]
        video_titles.append(current_video.iloc[0]['title'])
        video_category.append(current_video.iloc[0]['category_id'])

        # If a video was trending multiple days, we only care about the max
        video_views.append(current_video['views'].max())
        video_likes.append(current_video['likes'].max())
        video_dislikes.append(current_video['dislikes'].max())
        video_comment_count.append(current_video['comment_count'].max())

    clean_df['title'] = video_titles
    clean_df['category'] = video_category
    clean_df['views'] = video_views
    clean_df['likes'] = video_likes
    clean_df['dislikes'] = video_dislikes
    clean_df['comment_count'] = video_comment_count
    print('loaded basic video stats')

    # Process transcripts
    transcripts = []
    sentiment_polarity = []
    sentiment_subjectivity = []
    linewise_polarity = []
    linewise_subjectivity = []

    for v_id in clean_df['video_id']:
        current_transcript = ""
        pol = []
        sub = []
        if transcript_dict[v_id] != None:
            for line in transcript_dict[v_id]:
                # Linewise Analysis
                blob = TextBlob(line)
                p, s = blob.sentiment
                if p != 0.0 or s != 0.0: # Drop unrecognized sentiment
                    pol.append(p)
                    sub.append(s)
                # Videowise Analysis
                current_transcript += line + " "

        # Linewise Analysis
        linewise_polarity.append(np.histogram(pol, bins=binp)[0] / (len(pol) if len(pol) != 0 else 1))
        linewise_subjectivity.append(np.histogram(sub, bins=bins)[0] / (len(sub) if len(sub) != 0 else 1))
        # Videowise Analysis
        transcript_blob = TextBlob(current_transcript)
        p, s = transcript_blob.sentiment
        transcripts.append(current_transcript)
        sentiment_polarity.append(p)
        sentiment_subjectivity.append(s)

    clean_df['transcript'] = transcripts
    clean_df['polarity'] = sentiment_polarity
    clean_df['subjectivity'] = sentiment_subjectivity
    clean_df['linewise_polarity'] = linewise_polarity
    clean_df['linewise_subjectivity'] = linewise_subjectivity

    # Remove videos with no transcripts
    num_total = len(clean_df)
    clean_df = clean_df[clean_df['transcript'] != ""]
    num_with_transcripts = len(clean_df)
    print('{} videos total, with {} having transcripts'.format(num_total, num_with_transcripts))

    return clean_df
```

```
#df = load_videos()
#df.head()
```

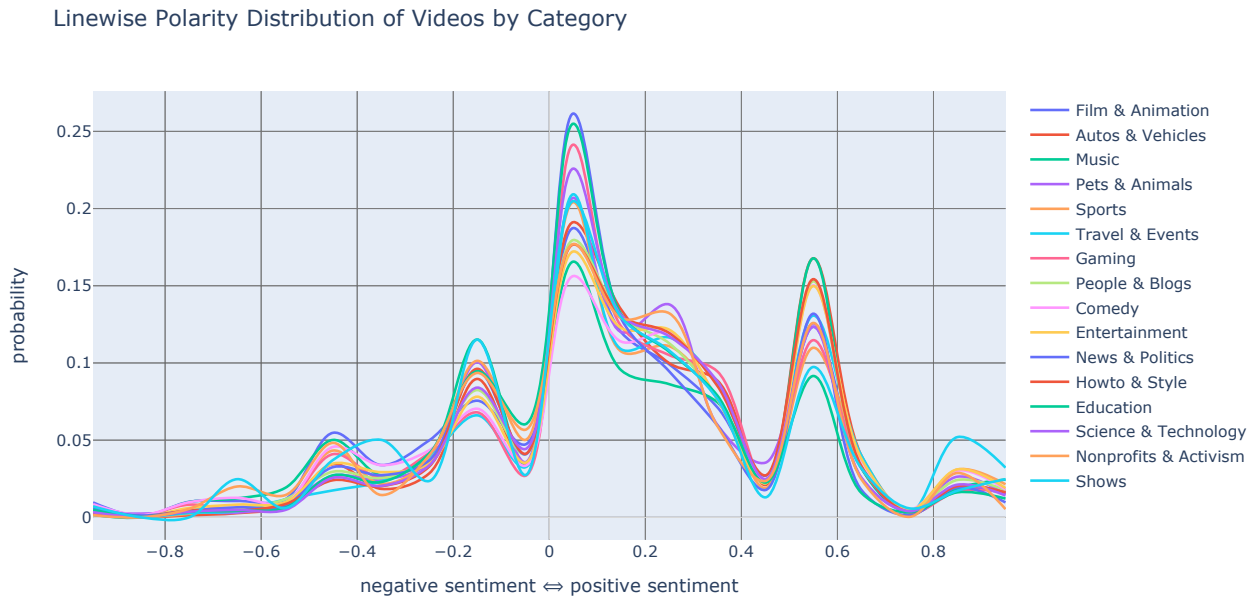
In [6]:
```
#pickle.dump(df, open("master.p", "wb"))
df = pickle.load(open("master.p", "rb"))
df.head()
```

Out[6]:

	video_id	title	category	views	likes	dislikes	comment_count	transcript	polarity	subjectivity	linewise_polarity	linewise_su
0	2kyS6SvSYSE	WE WANT TO TALK ABOUT OUR MARRIAGE	22	2564903	96321	7972	24225	[Music] Kenneth and I are headed to our therap...	0.168212	0.531523	[0.02247191011235955, 0.0, 0.0, 0.0, 0.0, 0.01...	0.022471910 0.0786516853
1	1ZAPwfrtAFY	The Trump Presidency: Last Week Tonight with J...	24	6109402	151250	11508	19820	The presidency\nof Donald Trump. The man voted...	0.055784	0.558050	[0.01858736059479554, 0.0, 0.01486988847583643...	[0.00371747211 0.0260223048
2	5qpjK5DgCt4	Racist Superman | Rudy Mancuso, King Bach & Le...	23	5315471	187303	7278	9990	Yo, I'm trying to tell you Supergirl I was lik...	0.112255	0.471696	[0.0, 0.0, 0.0, 0.0136986301369863, 0.01369863...	[0.013698630 0.0, 0.04109589
3	puqaWrEC7tY	Nickelback Lyrics: Real or Fake?	24	913268	16729	1386	3460	(bright, upbeat music) (fire crackles) W...	0.122311	0.510207	[0.012345679012345678, 0.0, 0.0123456790123456...	0.012345679 0.012345679
4	d380meD0W0M	I Dare You: GOING BALD!?	24	2819118	153395	2416	20573	Ryan: Why are we dressed like this? Greg: We'r...	0.134149	0.527256	[0.017857142857142856, 0.0, 0.0, 0.0, 0.035714...	[0.017857142 0.017857142

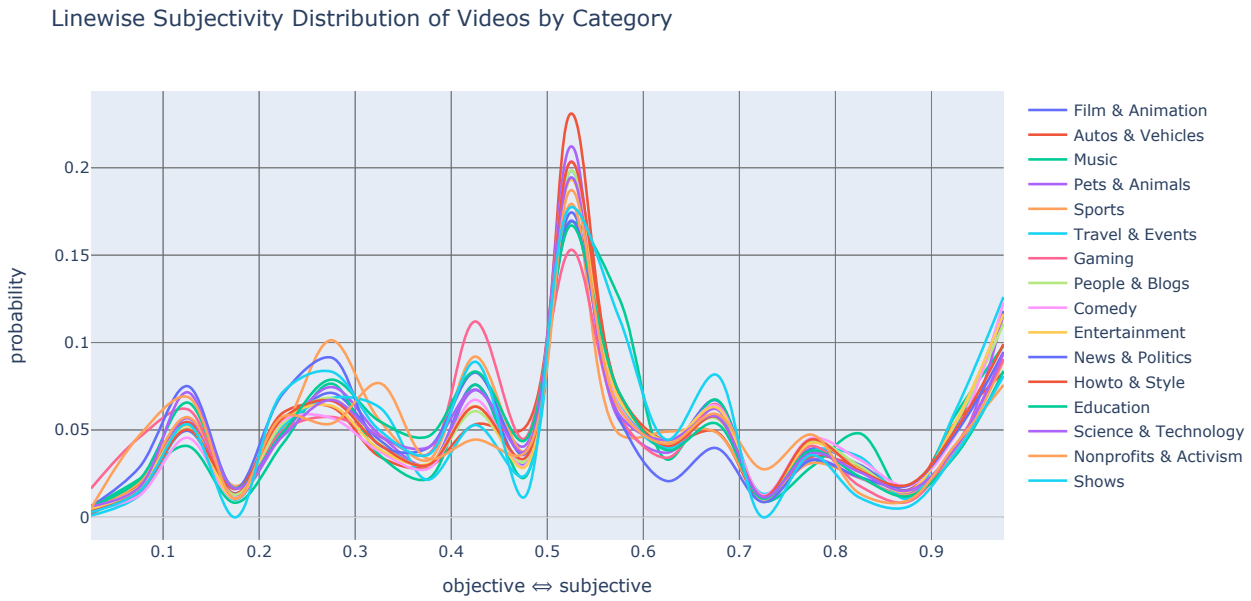
Data Visualization

In [7]:
```
"""
Compare different categories' linewise polarity distributions
"""
binp = np.linspace(-1.0, 1.0, num=NUM_BIN+1)
xp = (binp[1:] + binp[:-1]) / 2

fig = go.Figure()
for id in np.sort(df.category.unique()):
    category_dist = df[df.category == id]['linewise_polarity'].sum()
    category_dist = category_dist / sum(category_dist)
    trace = go.Scatter(x=xp, y=category_dist, line_shape='spline', name=category_df.loc[id][0])
    fig.add_trace(trace)

fig.update_layout(
    title="Linewise Polarity Distribution of Videos by Category",
    xaxis_title="negative sentiment ⇔ positive sentiment",
    yaxis_title="probability",
)
```



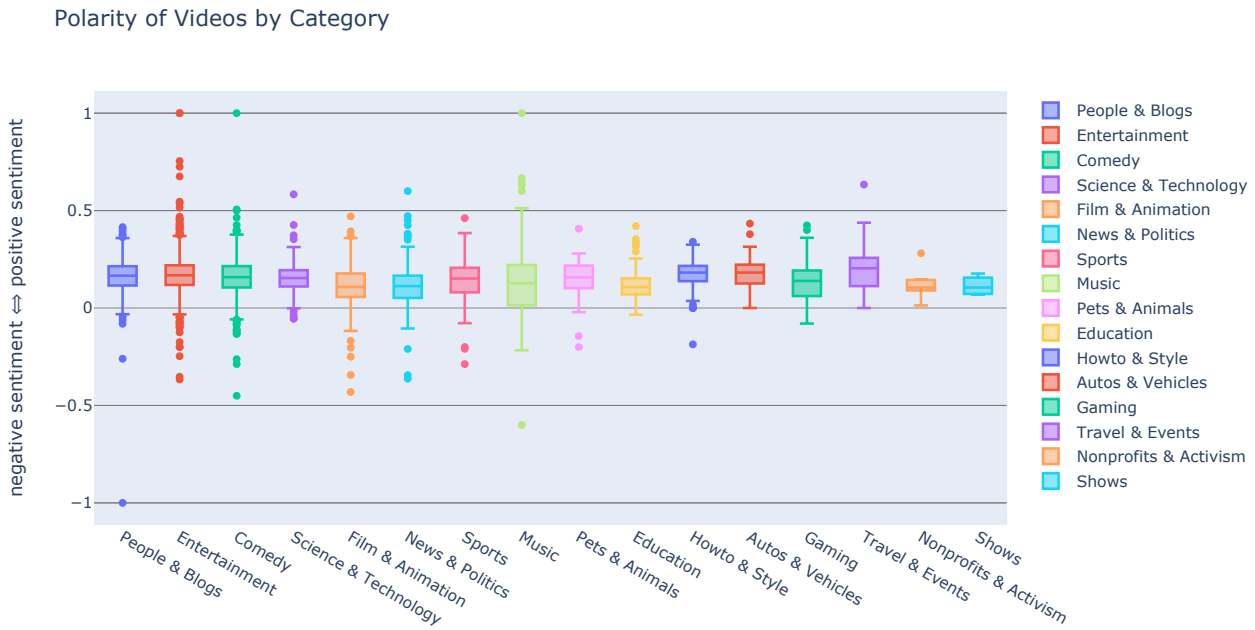
The above visualization is an interactive plot that shows the average distribution of polarity of lines within videos for all the various categories. To reveal or hide different categories, click on the category in the legend on the right. To only show a single category, double click on it in the legend. Each line represents a category of videos. The X-axis represents the polarity of videos based on each line in their transcripts with range [-1,1] negative to positive. The Y-axis represents the probability of a line from a given category having the polarity at each point on the X-axis.

In [8]:
```
"""
Compare different categories' linewise subjectivity distributions
"""
bins = np.linspace(0.0, 1.0, num=NUM_BIN+1)
xs = (bins[1:] + bins[:-1]) / 2

fig = go.Figure()
for id in np.sort(df.category.unique()):
    category_dist = df[df.category == id]['linewise_subjectivity'].sum()
    category_dist = category_dist / sum(category_dist)
    trace = go.Scatter(x=xs, y=category_dist, line_shape='spline', name=category_df.loc[id][0])
    fig.add_trace(trace)

fig.update_layout(
    title="Linewise Subjectivity Distribution of Videos by Category",
    xaxis_title="objective ⇔ subjective",
    yaxis_title="probability",
)
```



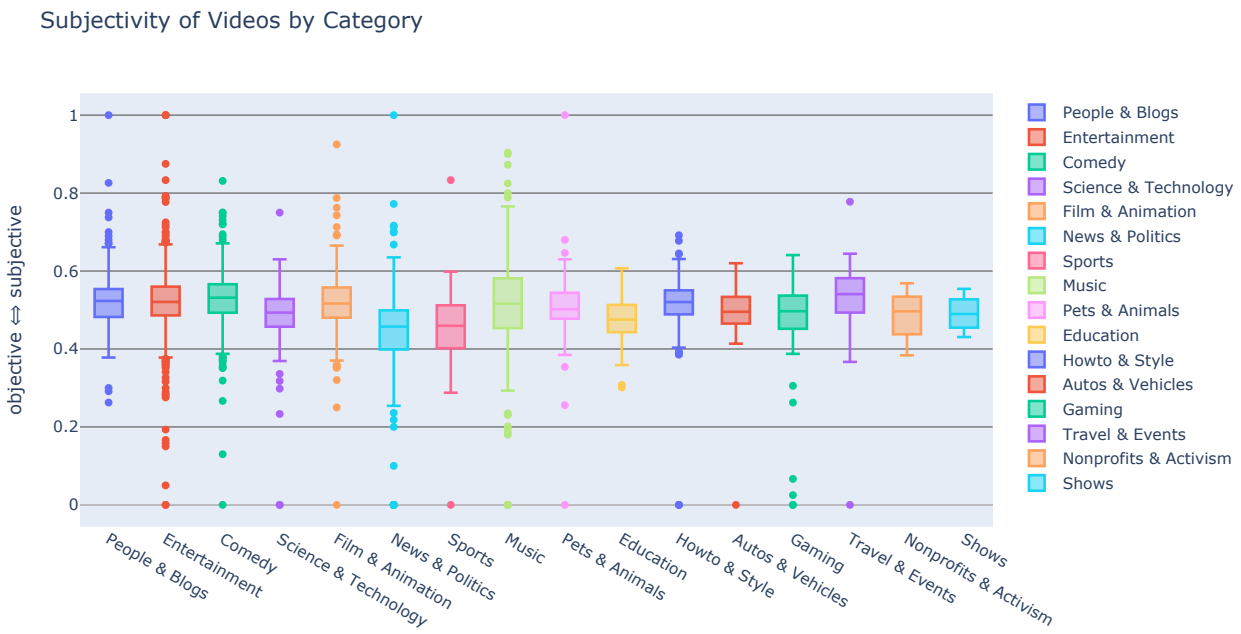
The above visualization is an interactive plot that shows the average distribution of subjectivity of lines within videos for all the various categories. To reveal or hide different categories, click on the category in the legend on the right. To only show a single category, double click on it in the legend. Each line represents a category of videos. The X-axis represents the subjectivity of videos based on each line in their transcripts with range [0,1] objective to subjective. The Y-axis represents the probability of a line from a given category having the subjectivity at each point on the X-axis.

In [9]:
```
"""
Box plots of a video's polarity for each category
"""
fig = go.Figure()
for category in df.category.unique():
    fig.add_trace(go.Box(y=df[df['category'] == category]['polarity'], name=category_df.loc[category][0]))

fig.update_layout(
    title="Polarity of Videos by Category",
    yaxis_title="negative sentiment ⇔ positive sentiment"
)
fig.show()
```



The above visualization is an interactive plot that shows the distribution of videos' overall polarity for all the various categories. To reveal or hide different categories, click on the category in the legend on the right. To only show a single category, double click on it in the legend. Each box represents a category of videos, indicated by the X-axis. The Y-axis indicates the polarity of videos based on concatenation of all lines in the transcripts with range [-1,1] negative to positive.

In [10]:
```
"""
Box plots of a video's subjectivity for each category
"""
fig = go.Figure()
for category in df.category.unique():
    fig.add_trace(go.Box(y=df[df['category'] == category]['subjectivity'], name=category_df.loc[category][0]))

fig.update_layout(
    title="Subjectivity of Videos by Category",
    yaxis_title="objective ⇔ subjective"
)
fig.show()
```

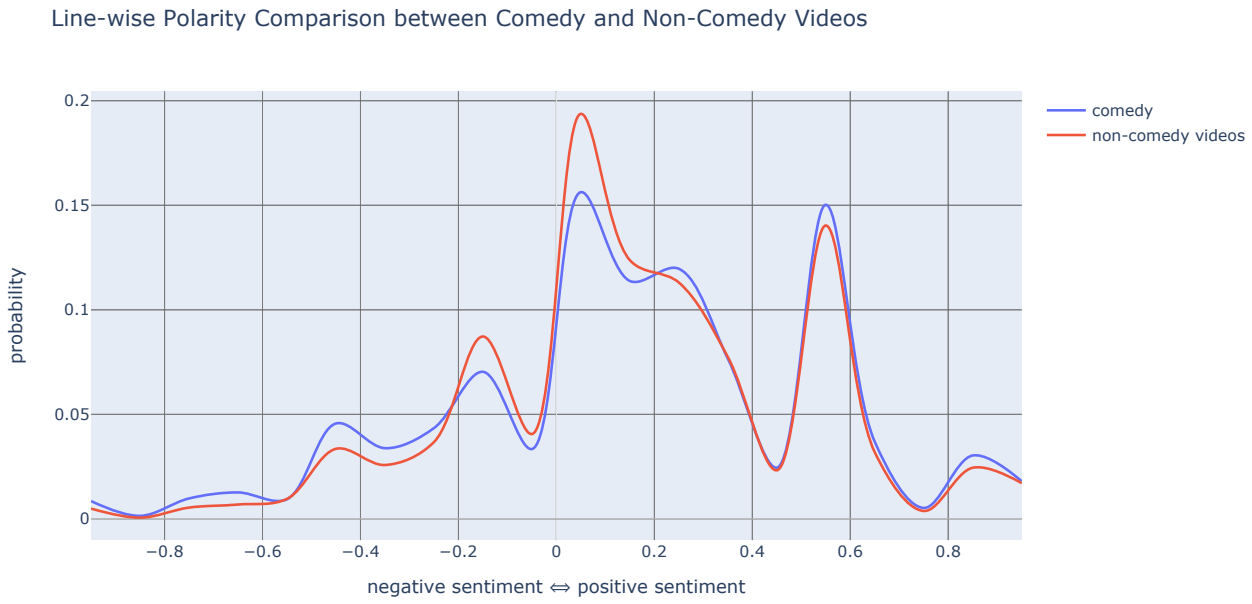


The above visualization is an interactive plot that shows the distribution of videos' overall subjectivity for all the various categories. To reveal or hide different categories, click on the category in the legend on the right. To only show a single category, double click on it in the legend. Each box represents a category of videos, indicated by the X-axis. The Y-axis indicates the subjectivity of videos based on concatenation of all lines in the transcripts with range [0,1] objective to subjective.

Data Analysis & Results

Hypothesis 1A: Comedy Videos Contain More Positive Sentences than Others

In [11]:
```
"""
Compare line-wise polarity distribution of the "comedy" (id 23) videos to that of all other videos
"""
binp = np.linspace(-1.0, 1.0, num=NUM_BIN+1)
xp = (binp[1:] + binp[:-1]) / 2

fig = go.Figure()
comedy_dist = df[df.category == 23]['linewise_polarity'].sum()
fig.add_trace(go.Scatter(x=xp, y=comedy_dist/sum(comedy_dist),
                         line_shape='spline', name="comedy")) # Normalization

non_comedy_dist = df[df.category != 23]['linewise_polarity'].sum()
fig.add_trace(go.Scatter(x=xp, y=non_comedy_dist / sum(non_comedy_dist),
                         line_shape='spline', name="non-comedy videos")) # Normalization

fig.update_layout(
    title="Line-wise Polarity Comparison between Comedy and Non-Comedy Videos",
    xaxis_title="negative sentiment ⇔ positive sentiment",
    yaxis_title="probability",
)
fig.show()
```



It can be seen that comedy videos contain **both** more positive sentences and more negative sentences than non-comedy videos.

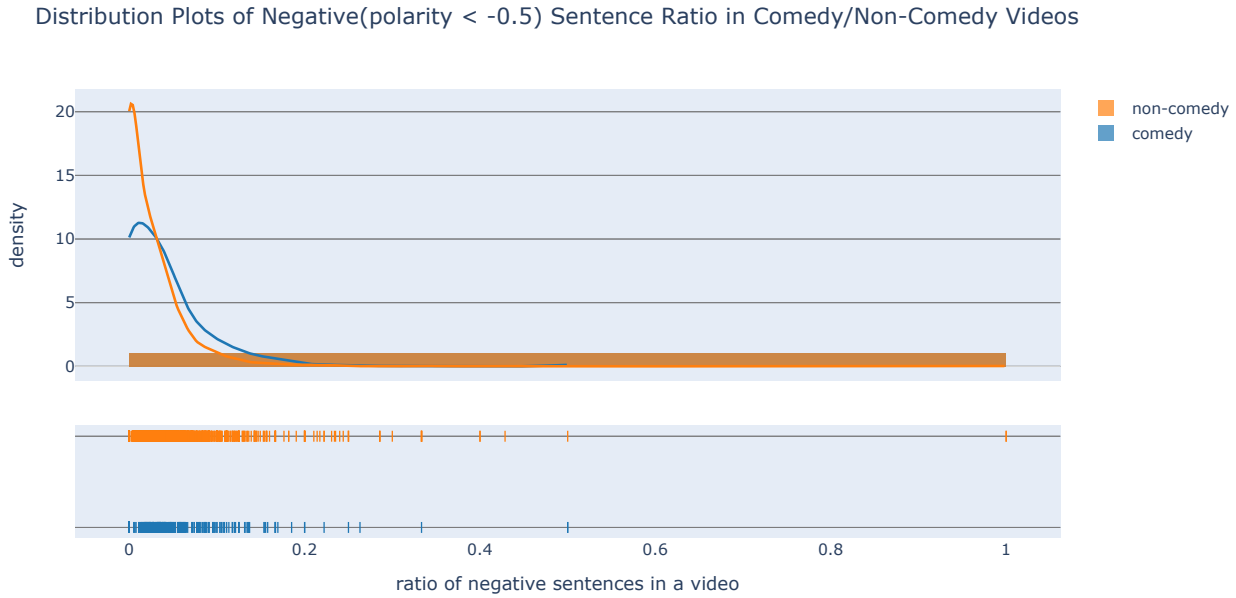
H0: the average ratio of negative(polarity < -0.5) sentences in comedy videos and in non-comedy videos are the same

H1: the average ratio of negative(polarity < -0.5) sentences in comedy videos are more than in other videos

We have the ratio of sentences with polarity < 0.5 per video, that is the ratio of sentences fall into one of the first 5 bins, i.e. ([-1,-0.9],[-0.9,-0.8],...,[-0.6,-0.5])

Since we are comparing average, it is natural to use Two-Sample T-Test. It requires the data to follow the normal distribution.

In [12]:
```
x = df[df.category == 23]['linewise_polarity'].map(lambda lst : sum(lst[0:5]))
y = df[df.category != 23]['linewise_polarity'].map(lambda lst : sum(lst[0:5]))
fig = ff.create_distplot([x, y], ['comedy', 'non-comedy'])

fig.update_layout(
    title="Distribution Plots of Negative(polarity < -0.5) Sentence Ratio in Comedy/Non-Comedy Videos",
    xaxis_title="ratio of negative sentences in a video",
    yaxis_title="density",
)
fig.show()
```



Since the data clearly does not follow the normal distribution, we have to use the Mann–Whitney U test (non-parametric).

In [13]:
```
mannwhitneyu(x, y, alternative='greater')
```

Out[13]:
```
MannwhitneyuResult(statistic=897154.0, pvalue=5.746452682260195e-11)
```

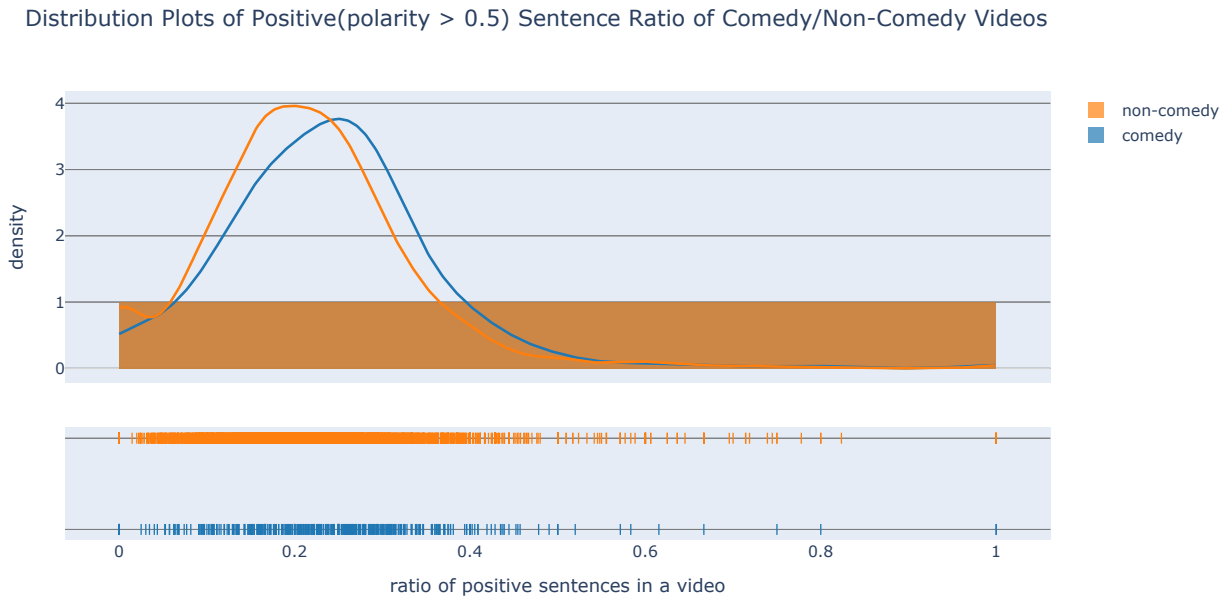
With alpha = 0.05, we can reject the null hypothesis. It can be seen that the p-value is so small that we can argue with certainty that comedy videos contain more sentences with negative sentiments.

Similarly, we shall test

H0: the average ratio of positive(polarity > 0.5) sentences in comedy videos and in non-comedy videos are the same

H1: the average ratio of positive(polarity > 0.5) sentences in comedy videos are more than in other videos

In [14]:
```
x = df[df.category == 23]['linewise_polarity'].map(lambda lst : sum(lst[-5:]))
y = df[df.category != 23]['linewise_polarity'].map(lambda lst : sum(lst[-5:]))
fig = ff.create_distplot([x, y], ['comedy', 'non-comedy'])

fig.update_layout(
    title="Distribution Plots of Positive(polarity > 0.5) Sentence Ratio of Comedy/Non-Comedy Videos",
    xaxis_title="ratio of positive sentences in a video",
    yaxis_title="density",
)
fig.show()
```



It could be seen that the distributions are almost normal, so we can apply the Two-Sample T-Test. We can't assume the variances of x and y are equal.

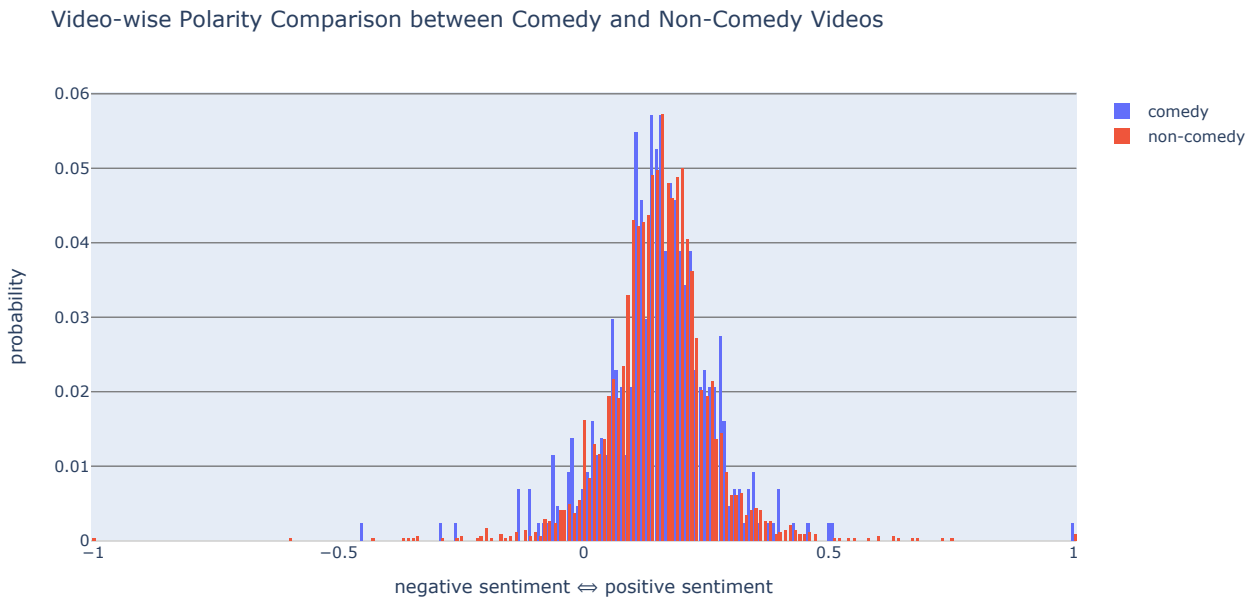
In [15]:
```
ttest_ind(x, y, equal_var=False)
```

Out[15]:
```
Ttest_indResult(statistic=3.892545948292066, pvalue=0.0001115189210343164)
```

With alpha = 0.05, since `p/2 < alpha and t > 0` (greater-than test), we can reject the null hypothesis. It can be seen that the p-value is so small that we can argue with certainty that comedy videos contain more sentences with positive sentiments.

Hypothesis 1B: Comedy Videos Are Generally More Positive than Others

In [16]:
```
"""
Compare video-wise polarities of the "comedy" (id 23) videos to that of all other videos
"""

fig = go.Figure()
x = df[df.category == 23]['polarity']
y = df[df.category != 23]['polarity']
fig.add_trace(go.Histogram(x=x, name='comedy', histnorm='probability'))
fig.add_trace(go.Histogram(x=y, name='non-comedy', histnorm='probability'))

fig.update_layout(
    title="Video-wise Polarity Comparison between Comedy and Non-Comedy Videos",
    xaxis_title="negative sentiment ⇔ positive sentiment",
    yaxis_title="probability",
)

fig.show()
```



We can hardly say from the plot that comedy videos are more positive.

H0: the average polarity of comedy videos is the same as the average polarity of non-comedy videos.

H1: the average polarity of comedy videos is higher than the average polarity of non-comedy videos.

It can be seen from the above plot that the data follows the normal distribution. We may use Two-Sample T-Test.

In [17]:
```
ttest_ind(x, y, equal_var=False)
```

Out[17]:
```
Ttest_indResult(statistic=0.12997785229946718, pvalue=0.8966331405551706)
```

With alpha = 0.05, we fail to reject the null hypothesis.

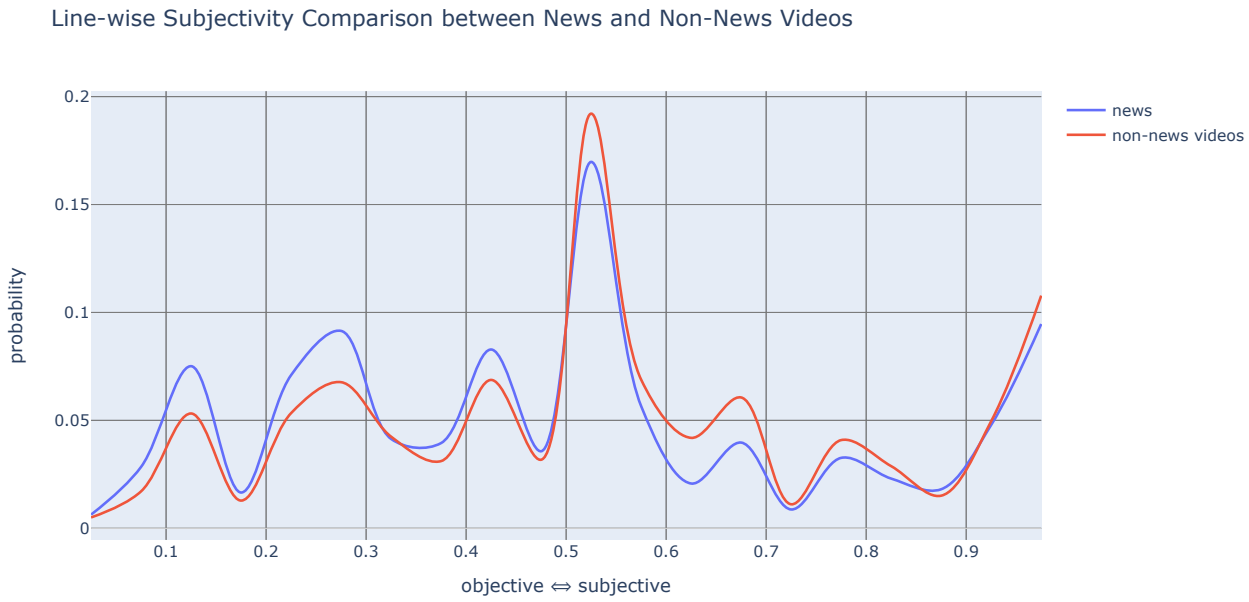
Conclusion

The tests support the hypothesis that comedy videos contain more positive sentences but fail to support the hypothesis that comedy videos are generally more positive.

It suggests that, compared to the other videos, the comedy videos *substitute* non-emotional sentences with **both positive and negative** sentences. It is an interesting discovery and it makes sense—a possible explanation is that a comedy makes audience laugh not by using a lot of "happy" words but by creating dramatic conflicts.

Hypothesis 2A: News Videos Contain More Objective Sentences than Others

In [18]:
```
"""
Compare line-wise subjectivity distribution of the "news&politics" (id 25) videos to that of all other videos
"""
bins = np.linspace(0.0, 1.0, num=NUM_BIN+1)
xs = (bins[1:] + bins[:-1]) / 2

fig = go.Figure()
news_dist = df[df.category == 25]['linewise_subjectivity'].sum()
fig.add_trace(go.Scatter(x=xs, y=news_dist/sum(news_dist), line_shape='spline', name="news"))

non_news_dist = df[df.category != 25]['linewise_subjectivity'].sum()
fig.add_trace(go.Scatter(x=xs, y=non_news_dist/sum(non_news_dist), line_shape='spline', name="non-news videos"))

fig.update_layout(
    title="Line-wise Subjectivity Comparison between News and Non-News Videos",
    xaxis_title="objective ⇔ subjective",
    yaxis_title="probability",
)
fig.show()
```



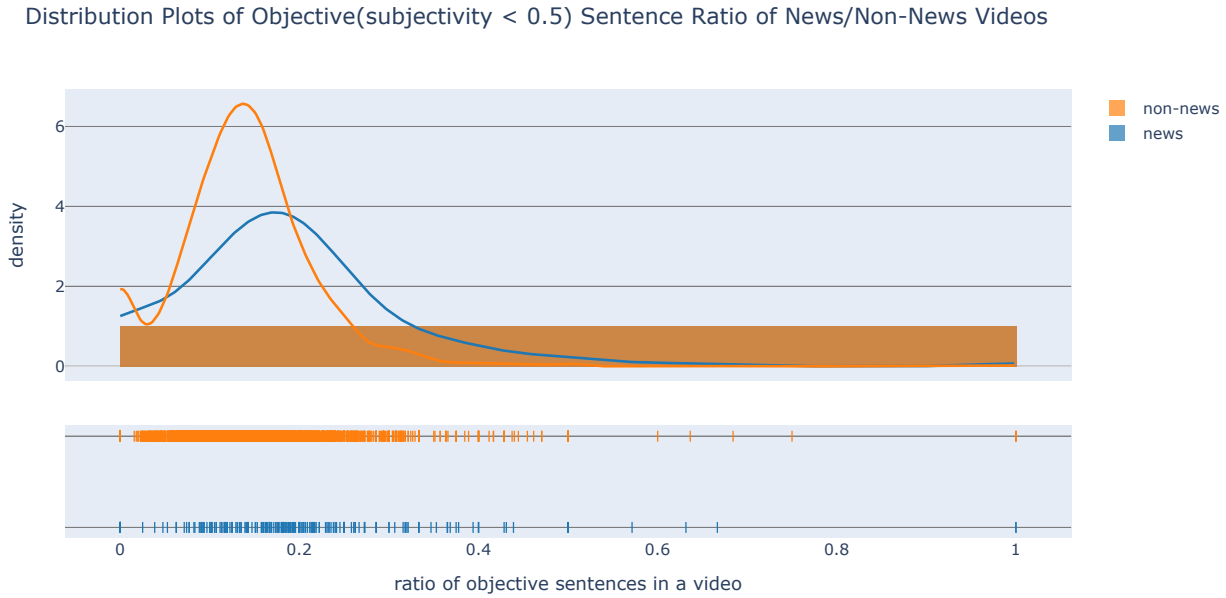
It is pretty obvious that news videos contain more objective sentences than non-news videos. The blue line is higher than the red line on the left and is lower than the red line on the right.

We still need statistical evidence.

H0: the average ratio of objective(subjectivity < 0.5) sentences in news videos and in non-news videos are the same

H1: the average ratio of objective(subjectivity < 0.5) sentences in news videos are more than in other videos

In [19]:
```
x = df[df.category == 25]['linewise_subjectivity'].map(lambda lst : sum(lst[0:5]))
y = df[df.category != 25]['linewise_subjectivity'].map(lambda lst : sum(lst[0:5]))
fig = ff.create_distplot([x, y], ['news', 'non-news'])

fig.update_layout(
    title="Distribution Plots of Objective(subjectivity < 0.5) Sentence Ratio of News/Non-News Videos",
    xaxis_title="ratio of objective sentences in a video",
    yaxis_title="density",
)
fig.show()
```



It could be seen that the distributions are almost normal, so we can apply the Two-Sample T-Test. We can't assume the variances of x and y are equal.

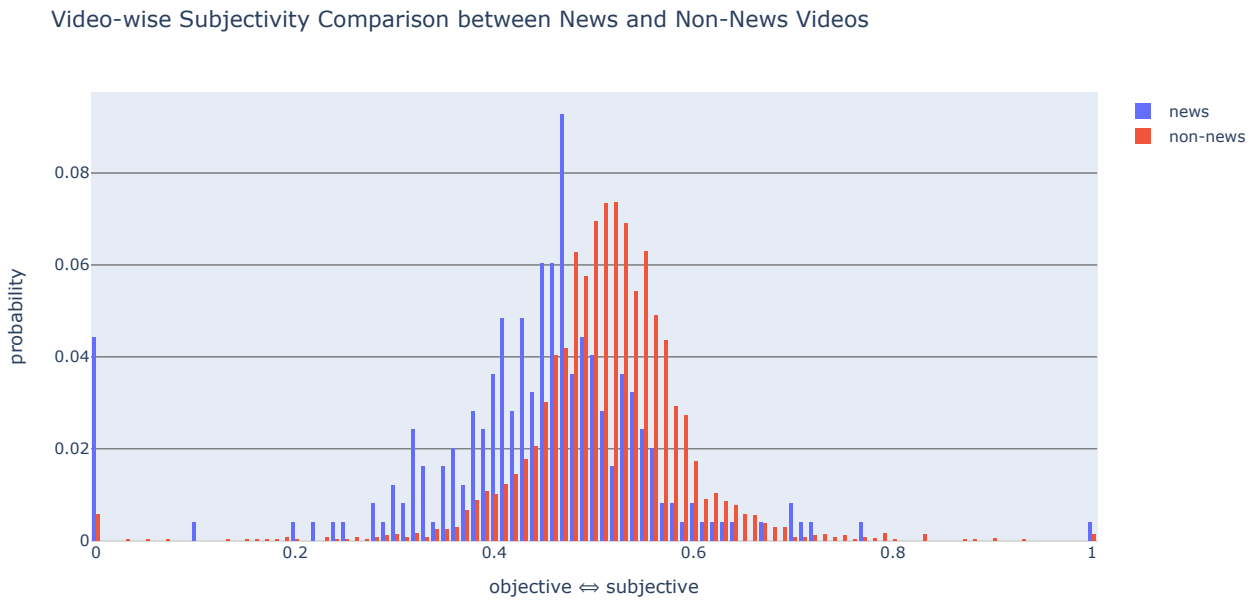
In [20]:
```
ttest_ind(x, y, equal_var=False)
```

Out[20]:
```
Ttest_indResult(statistic=5.353535258603698, pvalue=1.904292604706516e-07)
```

With alpha = 0.05, since `p/2 < alpha and t > 0` (greater-than test), we can reject the null hypothesis. It can be seen that the p-value is so small that we can argue with certainty that news videos contain more objective sentences than others.

Hypothesis 2B: News Videos Are Generally More Objective than Others

In [21]:
```
"""
Compare video-wise subjectivities of the "news&politics" (id 25) videos to that of all other videos
"""

fig = go.Figure()
x = df[df.category == 25]['subjectivity']
y = df[df.category != 25]['subjectivity']
fig.add_trace(go.Histogram(x=x, name='news', histnorm='probability'))
fig.add_trace(go.Histogram(x=y, name='non-news', histnorm='probability'))

fig.update_layout(
    title="Video-wise Subjectivity Comparison between News and Non-News Videos",
    xaxis_title="objective ⇔ subjective",
    yaxis_title="probability",
)

fig.show()
```



The plot clearly shows news videos are generally more objective than non-news videos.

H0: the average subjectivity of news videos is the same as the average subjectivity of non-news videos.

H1: the average subjectivity of news videos is less than the average subjectivity of non-news videos.

It can be seen from the above plot that the data follows the normal distribution. We may use Two-Sample T-Test.

In [22]:
```
ttest_ind(x, y, equal_var=False)
```

Out[22]:
```
Ttest_indResult(statistic=-8.751936559400812, pvalue=2.682693177870182e-16)
```

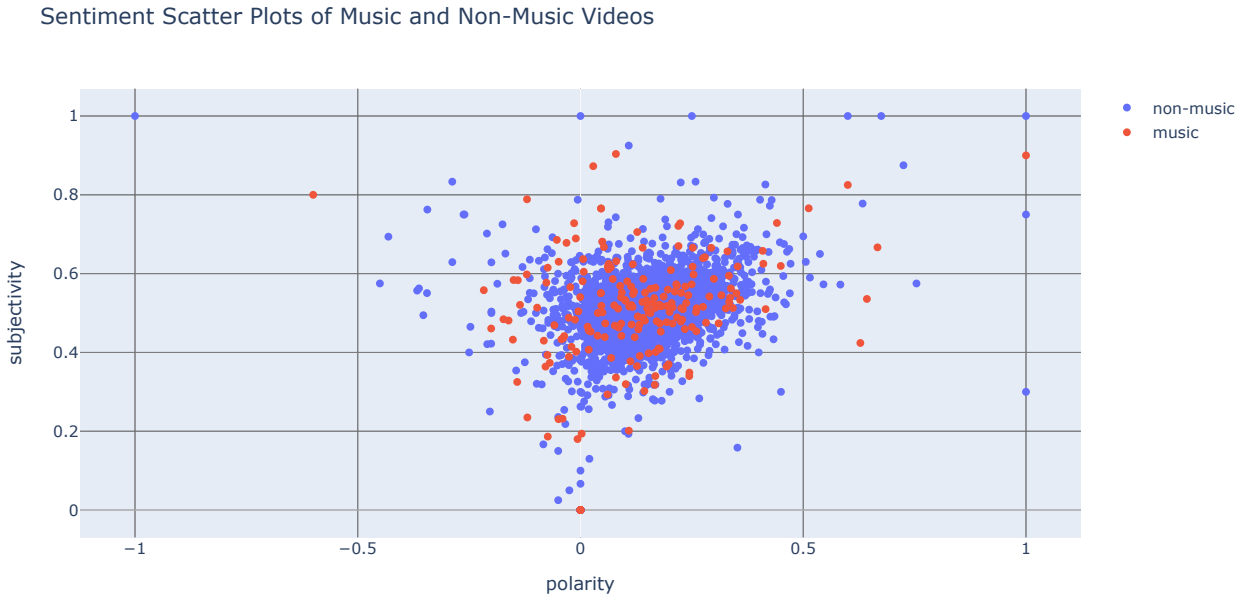
With alpha = 0.05, since `p/2 < alpha and t < 0` (less-than test), we can reject the null hypothesis. The p-value is almost 0.

Conclusion

The tests support both that news videos are generally more objective and that news videos contain more objective sentences, with high significance. This is consistent with our common sense.

Hypothesis 3: Music Videos Have More Varying Sentiments than Others

In [23]:
```
df_music    = df[df.category==10]
df_nonmusic = df[df.category!=10]

fig = go.Figure()
fig.add_trace(go.Scatter(x=df_nonmusic['polarity'], y=df_nonmusic['subjectivity'], mode='markers', name='non-music'))
fig.add_trace(go.Scatter(x=df_music['polarity'], y=df_music['subjectivity'], mode='markers', name='music'))

fig.update_layout(
    title="Sentiment Scatter Plots of Music and Non-Music Videos",
    xaxis_title="polarity",
    yaxis_title="subjectivity",
)
fig.show()
```



It could be seen that although there are a lot more non-music videos than music videos, the sentiments of the music videos are as widespread as that of the non-music videos.

The plot does not exhibit a correlation between videos' subjectivity and polarity in either group. Therefore, we may analyze the variance of polarity and subjectivity separately.

H0: the polarity of music videos and that of non-music videos have equal variance.

H1: the polarity of music videos and that of non-music videos have different variance.

Since the data seems to follow the normal distribution, we may use Bartlett's Test for testing equal variance.

In [24]:
```
x = df_music['polarity']
y = df_nonmusic['polarity']
print("The variance of music videos' polarity is", np.var(x))
print("The variance of non-music videos' polarity is", np.var(y))
bartlett(x, y)
```

```
The variance of music videos' polarity is 0.029981194771833534
The variance of non-music videos' polarity is 0.010101879652775963
```

Out[24]:
```
BartlettResult(statistic=161.4969198260581, pvalue=5.328421925913653e-37)
```

With alpha = 0.05, we can reject the null hypothesis. Since the variance of music videos is much greater than the variance of non-music videos, we could argue that the polarity of music videos is more varying than that of other videos.

H0: the subjectivity of music videos and that of non-music videos have equal variance.

H1: the subjectivity of music videos and that of non-music videos have different variance.

In [25]:
```
x = df_music['subjectivity']
y = df_nonmusic['subjectivity']
print("The variance of music videos' subjectivity is", np.var(x))
print("The variance of non-music videos' subjectivity is", np.var(y))
bartlett(x, y)
```

```
The variance of music videos' subjectivity is 0.022676838975062855
The variance of non-music videos' subjectivity is 0.007292277432133567
```

Out[25]:
```
BartlettResult(statistic=178.1918836027104, pvalue=1.2028632927850301e-40)
```

With alpha = 0.05, we can reject the null hypothesis. Since the variance of music videos is much greater than the variance of non-music videos, we could argue that the subjectivity of music videos is more varying than that of other videos.

Conclusion

The tests support our hypothesis that the overall sentiments of music videos are more varying than others. It makes sense as most music videos have completely different types of lyrics, and we would expect lyrics to be generally more sensational than daily speeches.

Ethics & Privacy

For our project, the dataset that we used came directly from Kaggle, a public platform that contains data that can be accessed by any individual. These datasets are open-sourced under the Apache 2.0 license, and therefore, we provide a citation to our Youtube videos dataset in the references section. Additionally, the data being used adheres to the Safe Harbour rules because all unique identifiers of Youtube users have been removed. We are focusing primarily on the view count, transcription from the videos, and category_id within our project. We are able to ethically transcribe the videos because of the Youtube Privacy Policy which states that videos that contain content that breach the Safe Harbour rules (aka contain unique information about individuals) will be removed from their website. Therefore, it is safe to assume that videos do not contain any personal information.

We also believe that we have eliminated bias throughout our data analysis process due to the fact that our dataset is solely composed of the statistics of each video, such as the number of views, categories, likes, etc. Additionally, the transcription of the video is precisely the content that is being presented to us, so we are not adjusting the data when analyzing sentiment.

Furthermore, we acknowledge that the use of the dataset for our project is solely to find correlations between the sentiments of different Youtube video categories in the United States, and that does not necessarily mean causation. We recognize that our data only consists of trending videos in the United States which does not holistically analyze the correlations between videos and sentiment because various other videos, that are not as popular, are not considered in the data. Therefore, we understand that other lesser popular videos may change the results of observations. However, the implications of our data help us understand the differences in sentiment for typically popular videos in each category. We evaluated the correlations solely based on the analysis found within our data, and by no means do we mean any bias against specific Youtube videos.

Conclusion & Discussion

Throughout our project, we were able to come to various conclusions that supported our initial hypothesis and answered our research question. For our project, we analyzed over fifteen categories of videos on Youtube's Trending page and predicted the sentiment differences between three specific categories individually (comedy, news, and music videos) when compared to all other categories. We concluded that there do exist differences between each of these categories when compared with all others as comedy videos tend to contain more polar statements in general (both positive and negative), news videos are definitely more object, and the variance of sentiment is greater in music videos than non-music videos.

Parts of our data both support and reject our original hypothesis. For example, our original hypothesis claimed that comedy videos contained more positive sentences than others. Through analyzing the line-wise polarity distribution throughout the video, we found that our p-value was much smaller than our alpha, and therefore, we could conclude that comedy videos contain more positive sentences and supported our hypothesis. However, when analyzing if comedy videos provide a generally greater positive sentiment than other videos, we found that comedy videos tend to be both more positively and negatively polar in content, or more extreme polarity than other categories. This was an interesting find as it seems to reflect the fact that the generic "lighthearted humor" that we associate with comedy has transitioned to somewhat of more extreme positive and dark humor. Additionally, our hypothesis that news videos contain more objective sentences than other categories was supported by our line-wise and video-wise subjectivity analysis. Lastly, our hypothesis the variance of sentiment is greater in music videos than videos in other categories was also supported by the analysis we did comparing the polarity and subjectivity of all music videos versus non-music videos.

However, there are various drawbacks and limitations throughout our study. First off, we lost quite a bit of data to analyze due to missing transcriptions of some videos, so our analysis only represents the videos with transcripts. Additionally, we understand that the sentiment in some words in phrases may get lost in translation as the tone and context of the speaker are not necessarily considered. With these limitations considered, our original research question can still be answered as our data analysis demonstrates that there are significant differences in sentiment between categories of trending Youtube videos in the United States.

In [ ]:
```
```