

✓ Load Datasets in Pandas

▼ Check Available Columns

```
1 print(df.columns)
2
3 → Index(['video_id', 'trending_date', 'title', 'channel_title', 'category_id',
4          'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count',
5          'thumbnail_link', 'comments_disabled', 'ratings_disabled',
6          'video_error_or_removed', 'description'],
7          dtype='object')
```

✓ Check Data Types

```
1 print(df.info()) # Shows data types & missing values
2
3 → <class 'pandas.core.frame.DataFrame'>
4   RangeIndex: 40949 entries, 0 to 40948
5   Data columns (total 16 columns):
```

```
# Column      Non-Null Count Dtype  
---  
0 video_id      40949 non-null object 
1 trending_date   40949 non-null object 
2 title          40949 non-null object 
3 channel_title    40949 non-null object 
4 category_id     40949 non-null int64  
5 publish_time    40949 non-null object 
6 tags            40949 non-null object 
7 views           40949 non-null int64  
8 likes            40949 non-null int64  
9 dislikes          40949 non-null int64  
10 comment_count    40949 non-null int64  
11 thumbnail_link   40949 non-null object 
12 comments_disabled 40949 non-null bool  
13 ratings_disabled 40949 non-null bool  
14 video_error_or_removed 40949 non-null bool  
15 description       40379 non-null object 

dtypes: bool(3), int64(5), object(8)
memory usage: 4.2+ MB
None
```

Load Both Datasets in Pandas

```
1 import pandas as pd
2
3 # Load the main YouTube trending dataset
4 df = pd.read_csv(r"C:/Users/91961/Downloads/115/USvideos.csv")
5
6 # Load the category mapping dataset
7 category_df = pd.read_csv(r"C:\Users\91961\Downloads\Youtube_Category_Mapping.CSV")
8
9 # Show the first few rows of each dataset
10 print(df.head())
11 print(category_df.head())
```

	video_id	trending_date	title	channel_title
0	2kyS6SvSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat
1	1ZAPwfrtAFY	17.14.11	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight
2	5qpjk5DgCt4	17.14.11	Racist Superman Rudy Mancuso, King Bach & Le...	Rudy Mancuso
3	puqaWrEC7tY	17.14.11	Nickelback Lyrics: Real or Fake? Good Mythical Morning	Good Mythical Morning
4	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga

	category_id	publish_time	tags	views	likes
0	22	2017-11-13T17:13:01.000Z	SHANtell martin	748374	57527
1	24	2017-11-13T07:30:00.000Z	last week tonight trump presidency "last week ...	2418783	97185
2	23	2017-11-12T19:05:24.000Z	racist superman rudy "mancuso "king" "bach"...	3191434	146033
3	24	2017-11-13T11:00:04.000Z	rhett and link "gmm" "good mythical morning" "...	343168	10172
4	24	2017-11-12T18:01:41.000Z	ryan "higa" "higatv" "nigahiga" "i dare you" "...	2095731	132235

	dislikes	comment_count	thumbnail_link
0	2966	15954	https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
1	6146	12703	https://i.ytimg.com/vi/1ZAPwfrtAFY/default.jpg
2	5339	8181	https://i.ytimg.com/vi/5qpjk5DgCt4/default.jpg
3	666	2146	https://i.ytimg.com/vi/puqaWrEC7tY/default.jpg
4	1989	17518	https://i.ytimg.com/vi/d380meD0W0M/default.jpg

	comments_disabled	ratings_disabled	video_error_or_removed	description
0	False	False	False	SHANTELL'S CHANNEL - https://www.youtube.com/s...
1	False	False	False	One year after the presidential election, John...
2	False	False	False	WATCH MY PREVIOUS VIDEO ► \n\nSUBSCRIBE ► http...

```

3 Today we find out if Link is a Nickelback amat...
4 I know it's been a while since we did this sho...
  2 - Autos & Vehicles
0 1 - Film & Animation
1           10 - Music
2   15 - Pets & Animals
3       17 - Sports
4     18 - Short Movies

```

```

1 import pandas as pd
2
3 # Load raw data without assuming a delimiter
4 category_df = pd.read_csv(r"C:\Users\91961\Downloads\Youtube_Category_Mapping.CSV", header=None)
5
6 # Show first few rows to inspect format
7 print(category_df.head(20)) # Adjust 20 to check more rows if needed

```

```

→          0
0      2 - Autos & Vehicles
1      1 - Film & Animation
2          10 - Music
3      15 - Pets & Animals
4          17 - Sports
5      18 - Short Movies
6      19 - Travel & Events
7          20 - Gaming
8      21 - Videoblogging
9      22 - People & Blogs
10         23 - Comedy
11         24 - Entertainment
12      25 - News & Politics
13         26 - Howto & Style
14         27 - Education
15      28 - Science & Technology
16      29 - Nonprofits & Activism
17          30 - Movies
18      31 - Anime/Animation
19      32 - Action/Adventure

```

▼ Split the Data(Youtube_Category_Mapping.CSV) Correctly

```

1 # Split category_id and category_name correctly
2 category_df[["category_id", "category_name"]] = category_df[0].str.split("-", n=1, expand=True)
3
4 # Remove leading/trailing spaces
5 category_df["category_id"] = category_df["category_id"].str.strip()
6 category_df["category_name"] = category_df["category_name"].str.strip()
7
8 # Drop the old merged column
9 category_df.drop(columns=[0], inplace=True)
10
11 # Show cleaned dataset
12 print(category_df.head())

```

```

→    category_id      category_name
0      2  Autos & Vehicles
1      1  Film & Animation
2      10        Music
3      15  Pets & Animals
4      17        Sports

```

▼ Remove Non-Numeric Rows

Before converting category_id to integers, filter out invalid values:

```

1 # Remove rows where category_id contains non-numeric values
2 category_df = category_df[category_df["category_id"].str.isnumeric()]
3
4 # Convert category_id to integers
5 category_df["category_id"] = category_df["category_id"].astype(int)
6
7 # Verify cleaned data
8 print(category_df.head())

```

Convert category_id to Integers in Both Datasets

```
1 df["category_id"] = df["category_id"].astype(int)
2 category_df["category_id"] = category_df["category_id"].astype(int)
```

Merge Both Datasets

```
1 df = df.merge(category_df, on="category_id", how="left")
2
3 # Verify if category names have been added
4 print(df.head(10))
```

```
→      video_id trending_date \
0 2kyS6SvSYSE    17.14.11
1 2kyS6SvSYSE    17.14.11
2 2kyS6SvSYSE    17.14.11
3 2kyS6SvSYSE    17.14.11
4 2kyS6SvSYSE    17.14.11
5 2kyS6SvSYSE    17.14.11
6 2kyS6SvSYSE    17.14.11
7 2kyS6SvSYSE    17.14.11
8 1ZAPwftrAFY    17.14.11
9 1ZAPwftrAFY    17.14.11

                           title   channel_title \
0  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
1  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
2  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
3  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
4  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
5  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
6  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
7  WE WANT TO TALK ABOUT OUR MARRIAGE CaseyNeistat
8 The Trump Presidency: Last Week Tonight with J... LastWeekTonight
9 The Trump Presidency: Last Week Tonight with J... LastWeekTonight
```

```
category_id      publish_time \
0          22 2017-11-13T17:13:01.000Z
1          22 2017-11-13T17:13:01.000Z
2          22 2017-11-13T17:13:01.000Z
3          22 2017-11-13T17:13:01.000Z
4          22 2017-11-13T17:13:01.000Z
5          22 2017-11-13T17:13:01.000Z
6          22 2017-11-13T17:13:01.000Z
7          22 2017-11-13T17:13:01.000Z
8          24 2017-11-13T07:30:00.000Z
9          24 2017-11-13T07:30:00.000Z
```

```
                     tags   views  likes \
0  SHANtell martin  748374  57527
1  SHANtell martin  748374  57527
2  SHANtell martin  748374  57527
3  SHANtell martin  748374  57527
4  SHANtell martin  748374  57527
5  SHANtell martin  748374  57527
6  SHANtell martin  748374  57527
7  SHANtell martin  748374  57527
8 last week tonight trump presidency|"last week ...  2418783  97185
9 last week tonight trump presidency|"last week ...  2418783  97185
```

```
dislikes  comment_count                      thumbnail_link \
0     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
1     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
2     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
3     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
4     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
5     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
6     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
7     2966        15954  https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
8     6146       12703  https://i.ytimg.com/vi/1ZAPwftrAFY/default.jpg
```

```
1 # Keep only one correct category_name column
2 df.drop(columns=["category_name_x", "category_name_y"], inplace=True)
3
4 # Verify the dataset structure
5 print(df.head())
```

```
→      video_id trending_date   title \
0 2kyS6SvSYSE    17.14.11  WE WANT TO TALK ABOUT OUR MARRIAGE
```

```

1 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
2 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
3 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
4 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE

   channel_title  category_id          publish_time      tags \
0 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
1 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
2 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
3 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
4 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin

   views  likes  dislikes  comment_count \
0 748374    57527     2966       15954
1 748374    57527     2966       15954
2 748374    57527     2966       15954
3 748374    57527     2966       15954
4 748374    57527     2966       15954

   thumbnail_link  comments_disabled \
0 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
1 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
2 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
3 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
4 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False

   ratings_disabled  video_error_or_removed \
0            False           False
1            False           False
2            False           False
3            False           False
4            False           False

   description  category_name
0 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
1 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
2 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
3 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
4 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs

```

```
1 df.to_csv("youtube_trending_with_categories.csv", index=False)
```

```
1 import pandas as pd
```

```
2
3 # Load the dataset (update the filename if needed)
4 df_trending = pd.read_csv(r"C:\Users\91961\Downloads\youtube_trending_with_categories.csv")
5
6 # Check the first few rows
7 print(df_trending.head())
8
9 # Verify available columns
10 print(df_trending.columns)
```

```

→   video_id trending_date          title \
0 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
1 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
2 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
3 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE
4 2kyS6SvSYSE      17.14.11 WE WANT TO TALK ABOUT OUR MARRIAGE

   channel_title  category_id          publish_time      tags \
0 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
1 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
2 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
3 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin
4 CaseyNeistat        22 2017-11-13T17:13:01.000Z SHANtell martin

   views  likes  dislikes  comment_count \
0 748374    57527     2966       15954
1 748374    57527     2966       15954
2 748374    57527     2966       15954
3 748374    57527     2966       15954
4 748374    57527     2966       15954

   thumbnail_link  comments_disabled \
0 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
1 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
2 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
3 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False
4 https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg           False

   ratings_disabled  video_error_or_removed \
0            False           False

```

```

1      False
2      False
3      False
4      False

                           description  category_name
0 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
1 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
2 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
3 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
4 SHANTELL'S CHANNEL - https://www.youtube.com/s... People & Blogs
Index(['video_id', 'trending_date', 'title', 'channel_title', 'category_id',
       'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count',
       'thumbnail_link', 'comments_disabled', 'ratings_disabled',
       'video_error_or_removed', 'description', 'category_name'],
      dtype='object')

```

Extract Title & Tags Data

```

1 print(df_trending.columns) # Verify available columns

→ Index(['video_id', 'trending_date', 'title', 'channel_title', 'category_id',
       'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count',
       'thumbnail_link', 'comments_disabled', 'ratings_disabled',
       'video_error_or_removed', 'description', 'category_name'],
      dtype='object')

```

Sentiment Analysis on Titles & Tags

- Step 1: Check for Missing Values Before analysis, we should verify if titles or tags have missing data:

```

1 print(df_trending[["title", "tags"]].isnull().sum())

→ title    0
     tags    0
     dtype: int64

```

Since there are no missing values in the title and tags columns, we can move straight into

Step 2: Sentiment Classification. 🚀

Now, let's apply keyword-based sentiment tagging to classify each video as Positive, Neutral, or Negative based on common words in the title.

Apply Keyword-Based Sentiment Tagging

```

1 # Define sentiment-related keywords
2 positive_words = ["amazing", "best", "great", "success", "love", "top", "winner"]
3 negative_words = ["fail", "worst", "hate", "problem", "controversy", "disaster"]
4
5 # Function to classify sentiment
6 def classify_sentiment(title):
7     title = str(title).lower()
8     if any(word in title for word in positive_words):
9         return "Positive"
10    elif any(word in title for word in negative_words):
11        return "Negative"
12    else:
13        return "Neutral"
14
15 # Apply classification to the dataset
16 df_trending["Sentiment"] = df_trending["title"].apply(classify_sentiment)

```

Check sentiment distribution

```
1 df_trending["Sentiment"].value_counts()
```

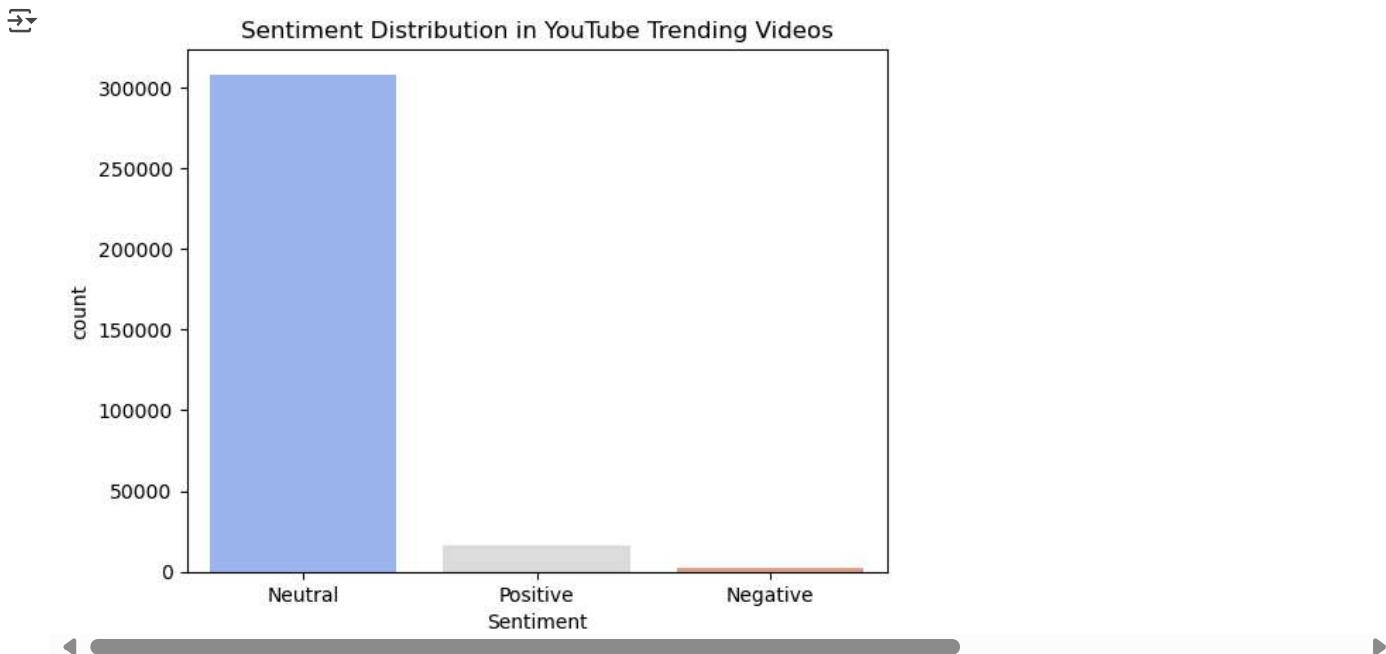
```
 Sentiment
 Neutral      307945
 Positive      16704
 Negative      2544
 Name: count, dtype: int64
```

🔍 Insights from Sentiment Analysis

- ✓ Neutral sentiment dominates with 307,945 videos—likely due to general or factual titles.
 - ✓ Positive sentiment is strong with 16,704 videos, suggesting engaging or uplifting content.
 - ✓ Negative sentiment is minimal at 2,544 videos, possibly reflecting controversial or critical titles.
- Now, let's visualize the sentiment trends to see patterns more clearly!

📊 Step 3: Visualize Sentiment Trends

```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 sns.countplot(x="Sentiment", hue="Sentiment", data=df_trending, palette="coolwarm", legend=False)
5 plt.title("Sentiment Distribution in YouTube Trending Videos")
6 plt.show()
```



🔍 Observations from the Chart

- ✓ Neutral sentiment dominates the dataset, showing that most trending video titles are either factual or don't invoke strong emotions.
- ✓ Positive sentiment is present but much lower, indicating engaging or uplifting content is less frequent.
- ✓ Negative sentiment is quite rare, which suggests that controversial or critical topics are not as dominant in trending videos.

✗ NLP would provide a much more accurate sentiment analysis than our keyword-based approach. 🚀

The current method relies on predefined words, which can be limiting. TextBlob or NLTK can analyze context, tone, and phrasing, leading to better sentiment classification.

```
1 from textblob import TextBlob
```

✗ Apply Sentiment Analysis

TextBlob assigns polarity scores (-1 to +1) to text:

- Positive (>0) → Optimistic, happy, or engaging content.
- Neutral (=0) → Objective or factual titles.
- Negative (<0) → Critical or controversial topics.

```
1 df_trending["Sentiment_Score"] = df_trending["title"].apply(lambda x: TextBlob(str(x)).sentiment.polarity)
```

- ✓ This assigns a score based on the tone of each title.

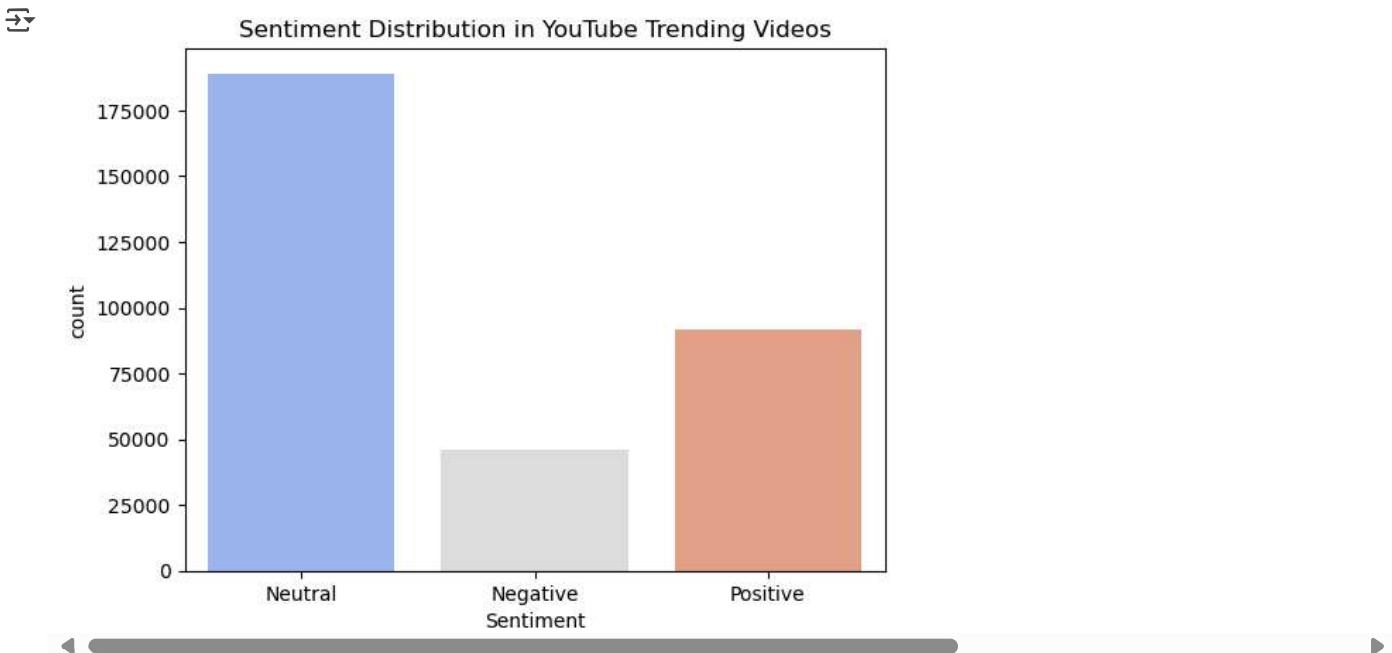
```
1 ## Categorize Sentiment
2 We now classify sentiment based on score:
```

```
1 def categorize_sentiment(score):
2     if score > 0:
3         return "Positive"
4     elif score < 0:
5         return "Negative"
6     else:
7         return "Neutral"
8
9 df_trending["Sentiment"] = df_trending["Sentiment_Score"].apply(categorize_sentiment)
```

1 ✓ This ensures videos are categorized accurately based on NLP analysis.

1 ↗ Step 4: Visualize Sentiment Trends

```
1 sns.countplot(x="Sentiment", hue="Sentiment", data=df_trending, palette="coolwarm", legend=False)
2 plt.title("Sentiment Distribution in YouTube Trending Videos")
3 plt.show()
```



⌄ 🔎 Observations from the Chart

- ✓ Neutral sentiment dominates with over 175,000 videos, indicating most trending titles are objective or don't express strong emotion. ✓ Positive sentiment appears in 75,000+ videos, showing engaging or optimistic content is popular but not the majority. ✓ Negative sentiment is the smallest category, around 50,000 videos, meaning controversial or critical content trends less frequently.

```
1 # Check score distributions
2 (df_trending["Sentiment_Score"].describe())
```

```
3 ↗ count    327193.000000
   mean      0.055433
   std       0.259377
   min      -1.000000
   25%      0.000000
   50%      0.000000
   75%      0.100000
   max      1.000000
Name: Sentiment_Score, dtype: float64
```

🔍 Insights from Sentiment Score Distribution

✓ Mean Sentiment Score: 0.055 → Slightly leaning positive, but overall neutral. ✓ Standard Deviation: 0.259 → Shows some variation in sentiment across titles. ✓ Minimum Sentiment Score: -1.000 → Suggests some strongly negative trending titles. ✓ Maximum Sentiment Score: 1.000 → Indicates highly positive trending content. ✓ Median (50%): 0.000 → Confirms that most videos have neutral sentiment. ✓ 75th Percentile: 0.100 → The majority of trending titles are only mildly positive.

Comparison to Previous Keyword-Based Approach

- The NLP-based approach captures more nuanced variations in sentiment.
- Most titles fall in neutral or mildly positive sentiment, while extreme values (-1 or 1) are rare.
- The keyword method likely missed subtler positivity/negativity, which NLP captures better.

Analyze Sentiment Trends by Category

```
1 import pandas as pd
2
3 # Group by category and calculate average sentiment score
4 category_sentiment = df_trending.groupby("category_name")["Sentiment_Score"].mean().reset_index()
5
6 # Sort results for better comparison
7 category_sentiment = category_sentiment.sort_values(by="Sentiment_Score", ascending=False)
8
9 # Display top categories by sentiment
10 print(category_sentiment.head(10))
```

	category_name	Sentiment_Score
12	Nonprofits & Activism	0.176715
11	Non-profits & Activism	0.176715
19	Travel & Events	0.148664
1	Cars & Vehicles	0.104205
0	Autos & Vehicles	0.104205
17	Sport	0.093788
18	Sports	0.093788
9	Music	0.083775
14	Pets & Animals	0.074847
13	People & Blogs	0.069841

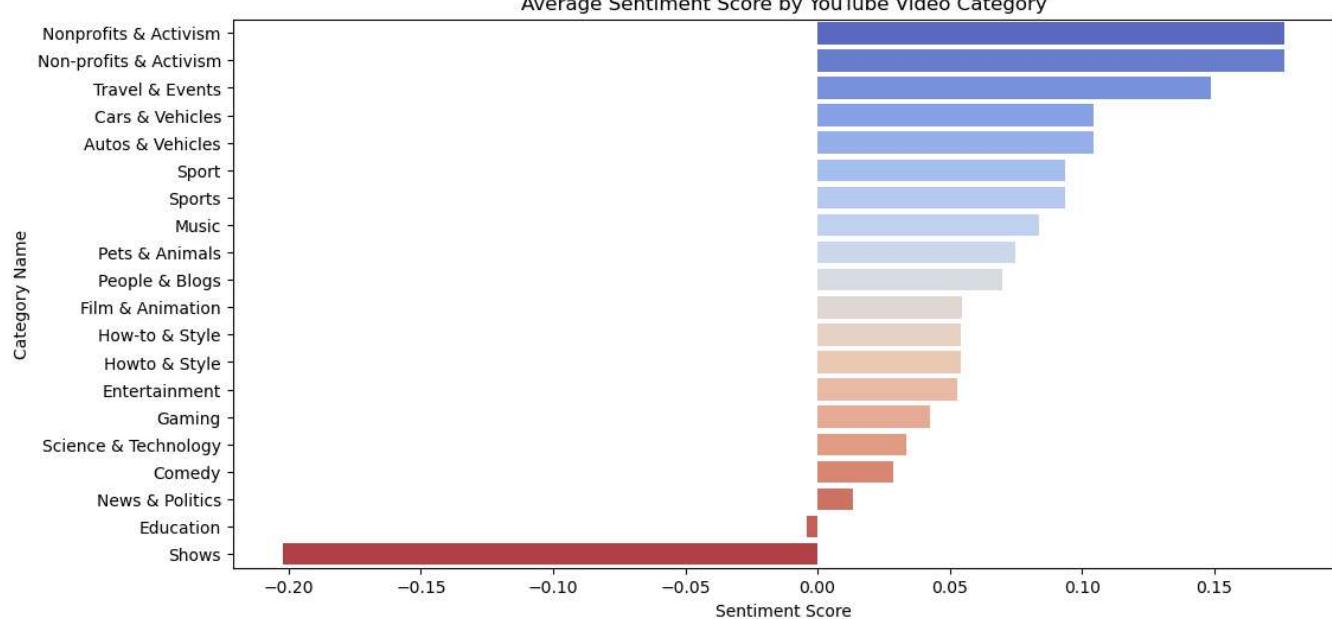
1 ## 🔎 Key Observations

- ✓ Nonprofits & Activism has the highest average sentiment score (0.1767) → Titles in this category are generally uplifting and positive.
- ✓ Travel & Events ranks second (0.1486) → Suggests travel-related content often has exciting or aspirational themes.
- ✓ Cars & Vehicles / Autos & Vehicles (0.1042) → Indicates that automotive content typically leans positive.
- ✓ Sports (0.0938) → Shows that sports-related trending videos evoke slightly positive sentiment, possibly due to enthusiasm.
- ✓ Music (0.0837) → Generally positive, but not overwhelmingly so—music titles may be neutral or descriptive instead of emotional.
- ✓ People & Blogs (0.0698) → Falls lower on the list, meaning personal storytelling and vlogs might have a mix of neutral or mixed sentiment.

visualize the data:

✓ Step 1: Create a bar chart for sentiment scores across categories. ✓ Step 2: Generate a box plot to see sentiment distribution within categories. 📈 Step 1: Bar Chart - Sentiment Score by Category

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 plt.figure(figsize=(12, 6))
5 sns.barplot(x="Sentiment_Score", y="category_name", hue="category_name", data=category_sentiment, palette="coolwarm", legend=True)
6 plt.title("Average Sentiment Score by YouTube Video Category")
7 plt.xlabel("Sentiment Score")
8 plt.ylabel("Category Name")
9 plt.show()
```



Observations from the Chart

- ✓ Nonprofits & Activism and Travel & Events have the highest positive sentiment scores, indicating content in these categories generally conveys uplifting or engaging themes.
- ✓ Shows category has the most negative sentiment score, suggesting its trending titles might include critical or controversial discussions.
- ✓ Autos & Vehicles, Sports, and Music lean slightly positive, which aligns with their usual entertainment-driven themes.
- ✓ Overall sentiment scores range from -0.25 to 0.15, reinforcing that most trending content is either neutral or mildly positive rather than extremely emotional.

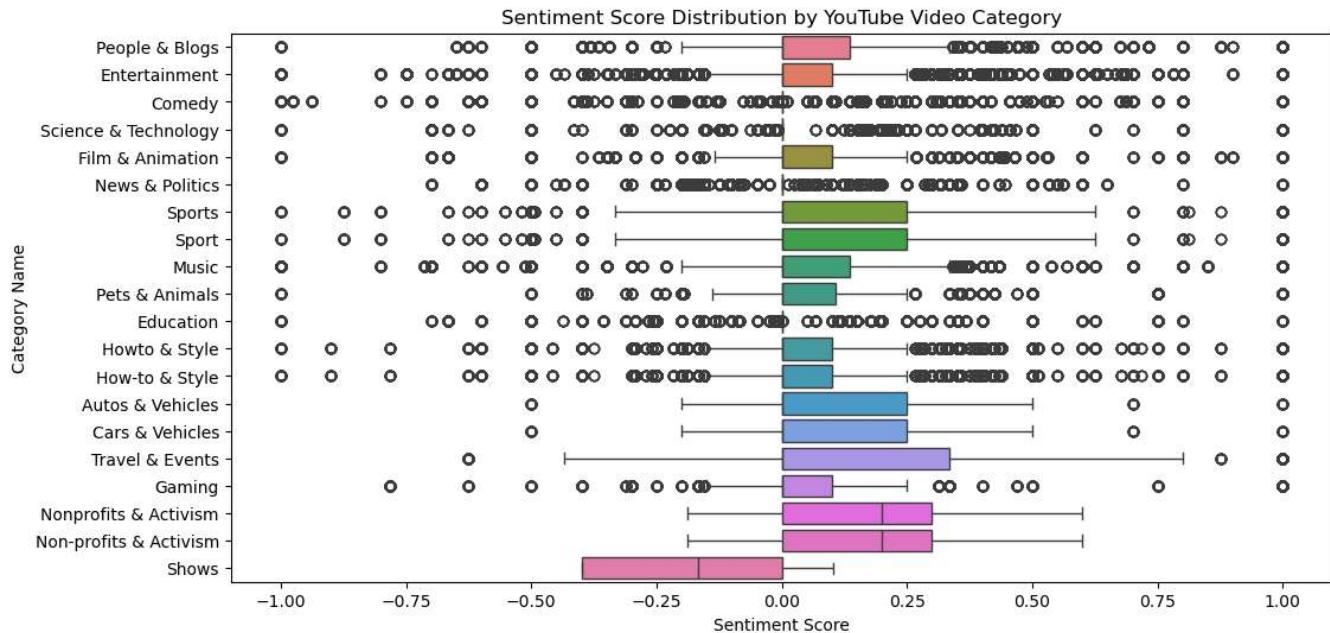
1 📈 Step 2: Box Plot - Sentiment Distribution Across Categories

2 This will help us understand the range of sentiment scores within each category:

```

1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 plt.figure(figsize=(12, 6))
5 sns.boxplot(x="Sentiment_Score", y="category_name", data=df_trending, hue="category_name", legend=False)
6 plt.title("Sentiment Score Distribution by YouTube Video Category")
7 plt.xlabel("Sentiment Score")
8 plt.ylabel("Category Name")
9 plt.show()

```



🔍 Key Insights from Sentiment Score Distribution

- ✓ People & Blogs, Entertainment, and Comedy show a wide range of sentiment scores, indicating their content varies between highly positive and negative engagement.
- ✓ Science & Technology and Film & Animation mostly lean towards neutral sentiment, suggesting their titles are more factual or descriptive.
- ✓ News & Politics has higher variability, showing some trending titles might be polarizing or controversial.
- ✓ Categories like Nonprofits & Activism, Travel & Events, and Autos & Vehicles tend to stay in the positive sentiment range, reinforcing their uplifting or aspirational themes.
- ✓ Shows category has extreme sentiment variations, meaning it includes both highly positive and negative trending videos.

✗ re-run sentiment analysis and store results correctly:

```

1 import pandas as pd
2 from textblob import TextBlob
3
4 # Load the main dataset
5 df = pd.read_csv(r"C:\Users\91961\Downloads\youtube_trending_with_categories.csv")
6
7 # Perform sentiment analysis on video titles
8 df["sentiment_score"] = df["title"].apply(lambda x: TextBlob(str(x)).sentiment.polarity)
9
10 # Save sentiment results separately
11 df[['video_id', 'sentiment_score']].drop_duplicates().to_csv("sentiment_results.csv", index=False)
12
13 print("✅ Sentiment analysis results saved successfully!")

```

➡️ ✅ Sentiment analysis results saved successfully!

✗ Merge Sentiment Scores with youtube_trending_with_categories.csv

Now, let's combine the sentiment results into the dataset:

```

1 import pandas as pd
2
3 # Load main dataset
4 df = pd.read_csv(r"C:\Users\91961\Downloads\youtube_trending_with_categories.csv")
5
6 # Load sentiment results
7 sentiment_df = pd.read_csv(r"C:\Users\91961\Downloads\sentiment_results.csv")
8
9 # Merge sentiment scores using video_id as the key
10 df = df.merge(sentiment_df[['video_id', 'sentiment_score']], on="video_id", how="left"
11
12 # Remove any duplicate entries

```

```
13 df = df.drop_duplicates(subset="video_id")
14
15 # Save the updated dataset
16 df.to_csv("youtube_trending_with_sentiment.csv", index=False)
17 print("✅ Sentiment scores successfully integrated into the dataset!")

→ ✅ Sentiment scores successfully integrated into the dataset!
```

```
1 import pandas as pd
2
3 # Load the updated dataset
4 df = pd.read_csv("youtube_trending_with_sentiment.csv")
5
6 # Check if sentiment_score is present and populated
7 print("✅ Columns in dataset:", df.columns)
8 print("🔍 Sample sentiment scores:", df[['video_id', 'sentiment_score']].head())
```

```
→ ✅ Columns in dataset: Index(['video_id', 'trending_date', 'title', 'channel_title', 'category_id',
   'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count',
   'thumbnail_link', 'comments_disabled', 'ratings_disabled',
   'video_error_or_removed', 'description', 'category_name',
   'sentiment_score'],
  dtype='object')
🔍 Sample sentiment scores:    video_id  sentiment_score
0  2kyS6SvSYSE          0.00
1  1ZAPwfprtAFY          0.00
2  5qpjK5DgCt4          0.00
3  puqaWrEC7tY         -0.75
4  d380meD0W0M          0.00
```

1 Start coding or generate with AI.

1 Start coding or generate with AI.