

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
1 import pandas as pd
2 df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Dataset/youtube.csv')
```

<ipython-input-25-6bca16ac4cb8>:2: DtypeWarning: Columns (1,2,3,4,6,7,8,9,10,15,16,17) have mixed types. Specify dtype option on im
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Dataset/youtube.csv')

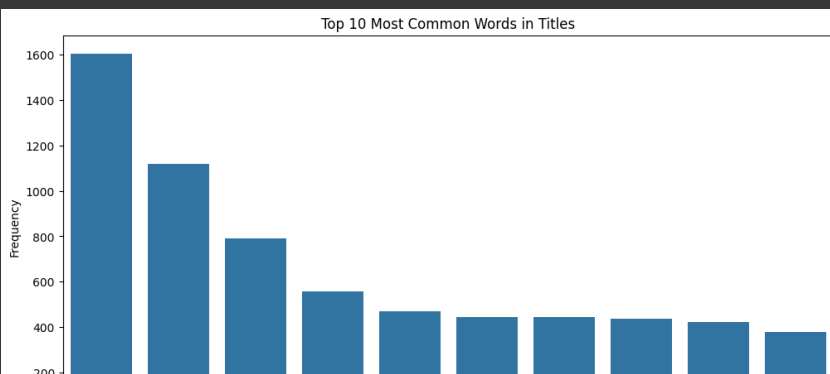
```
1 import nltk
2 from nltk.tokenize import word_tokenize
3 from nltk.corpus import stopwords
4 from collections import Counter
5 import matplotlib.pyplot as plt
6 import seaborn as sns
```

```
1 import nltk
2 nltk.download('stopwords')
3 nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
True
```

```
1 def preprocess_text(text):
2     if isinstance(text, str):
3         tokens = word_tokenize(text.lower())
4         tokens = [word for word in tokens if word.isalpha() and word not in stop_words]
5         return tokens
6     else:
7         return []
8
9 df['title_tokens'] = df['title'].apply(preprocess_text)
10 df['tags_tokens'] = df['tags'].apply(preprocess_text)
```

```
1 # Flatten the lists of tokens
2 all_title_tokens = [token for sublist in df['title_tokens'] for token in sublist]
3 all_tags_tokens = [token for sublist in df['tags_tokens'] for token in sublist]
4
5 # Get the most common words in titles and tags
6 top_title_words = Counter(all_title_tokens).most_common(10)
7 top_tags_words = Counter(all_tags_tokens).most_common(10)
8
9 # Plot the most common words in titles
10 plt.figure(figsize=(12, 6))
11 sns.barplot(x=[word[0] for word in top_title_words], y=[word[1] for word in top_title_words])
12 plt.title('Top 10 Most Common Words in Titles')
13 plt.xlabel('Words')
14 plt.ylabel('Frequency')
15 plt.show()
16
17 # Plot the most common words in tags
18 plt.figure(figsize=(12, 6))
19 sns.barplot(x=[word[0] for word in top_tags_words], y=[word[1] for word in top_tags_words])
20 plt.title('Top 10 Most Common Words in Tags')
21 plt.xlabel('Words')
22 plt.ylabel('Frequency')
23 plt.show()
```



```

1 import seaborn as sns
2
3 #Channel Analysis: Top Channels by views
4 top_channels_views = df.groupby('channel_title')['views'].sum().sort_values(ascending = False).head(10)
5
6 # Channel Analysis: Top Channels by Likes
7 top_channels_likes = df.groupby('channel_title')['likes'].sum().sort_values(ascending=False).head(10)
8
9 # Channel Analysis: Top Channels by Comments
10 top_channels_comments = df.groupby('channel_title')['comment_count'].sum().sort_values(ascending=False).head(10)

```

```

1 #Plotting
2 plt.figure(figsize = (15,10))
3 plt.subplot(2,2,1)
4 top_channels_views.plot(kind = 'bar', color = 'skyblue')
5 plt.title('Top Channels by Views')
6 plt.xlabel('Channel')
7 plt.ylabel('Total Views')
8
9 # Top Channels by Likes
10 plt.subplot(2, 2, 2)
11 top_channels_likes.plot(kind='bar', color='lightgreen')
12 plt.title('Top Channels by Likes')
13 plt.xlabel('Channel')
14 plt.ylabel('Total Likes')
15
16 # Top Channels by Comments
17 plt.subplot(2, 2, 3)
18 top_channels_comments.plot(kind='bar', color='gold')
19 plt.title('Top Channels by Comments')
20 plt.xlabel('Channel')
21 plt.ylabel('Total Comments')
22
23 plt.tight_layout()
24 plt.show()

```

