```
from datasets import load_dataset

dataset = load_dataset("BubuDavid/Selena-Gomez-With-Lyrics-And-Spotify-Audio-Features", data_files="songs_data.json")
```

```
Downloading and preparing dataset json/BubuDavid--Selena-Gomez-With-Lyrics-And-Spotify-Audio-Features to /root/.cache/huggin
Downloading data files: 100%  1/1 [00:00<00:00, 49.22it/s]
Extracting data files: 100%  1/1 [00:00<00:00, 37.19it/s]
Dataset json downloaded and prepared to /root/.cache/huggingface/datasets/BubuDavid___json/BubuDavid--Selena-Gomez-With-Lyric
100%  1/1 [00:00<00:00, 39.29it/s]
```

```
import pandas as pd
df = pd.DataFrame(dataset["train"])
```

```
df.head()
```

	spotify_id	title	lyrics	release_day	album	artists	features	popularity	explicit	duration_
0	0EtuSDTRJYUwlPf4y6coIz	999	Hace mucho tiempo que quiero decirte algo y no...	2021-08-26	{'available_markets': ['AD', 'AE', 'AG', 'AL',...	selena gomez, camilo	{'acousticness': 0.105, 'danceability': 0.781,...	63	False	2241
1	1r0XfrhdG6bsiS4oe1QM96	a year without rain	Can you feel me when I think about you? With e...	2010-01-01	{'available_markets': ['AD', 'AE', 'AG', 'AL',...	selena gomez & the scene	{'acousticness': 0.0102, 'danceability': 0.629...	61	False	2343
2	5MUZhMWlbypeWFeRY4sE2l	adiós	La-la-la-la, ah, ah, ah-ah, ah, la-ah La-la-la...	2021-03-12	{'available_markets': ['AD', 'AE', 'AG', 'AL',...	selena gomez	{'acousticness': 0.0768, 'danceability': 0.816...	57	False	1301
3	7iyjZ4paFWpTrJJenM0yZb	all night long	Now that I have captured your attention I wann...	1994-01-01	{'available_markets': ['CA', 'US'], 'name': 'I...	mary jane girls	{'acousticness': 0.171, 'danceability': 0.796,...	56	False	3439
4	79ncrzBkNWV1OtOswPjpmz	already missing you	Driving all night just to say goodbye Windows ...	2013-10-08	{'available_markets': ['AD', 'AE', 'AG', 'AL',...	prince royce, selena gomez	{'acousticness': 0.0686, 'danceability': 0.402...	39	False	2219

```
# df['title','lyrics', 'release_day', 'album', 'popularity']
# print(df[['title', 'lyrics', 'release_day', 'album', 'popularity']])
df = df[['title', 'lyrics', 'artists','release_day', 'album', 'popularity']]
```

```
df.head()
```

	title	lyrics	artists	release_day	albu
0	999	Hace mucho tiempo que quiero decirte algo y no...	selena gomez, camilo	2021-08-26	{'available_markets': ['AD', 'AE', 'AG', 'AL',.

```
df.drop(df.columns.difference(['title', 'lyrics', 'artists' ,'release_day', 'album', 'popularity']), axis=1, inplace=True)
df.head()
```

	title	lyrics	artists	release_day	albu
0	999	Hace mucho tiempo que quiero decirte algo y no...	selena gomez, camilo	2021-08-26	{'available_markets': ['AD', 'AE', 'AG', 'AL',.
1	a year without rain	Can you feel me when I think about you? With e...	selena gomez & the scene	2010-01-01	{'available_markets': ['AD', 'AE', 'AG', 'AL',.
2	adiós	La-la-la-la, ah, ah, ah-ah, ah, la-ah La-la-la...	selena gomez	2021-03-12	{'available_markets': ['AD', 'AE', 'AG', 'AL',.
3	all night long	Now that I have captured your attention I wann...	mary jane girls	1994-01-01	{'available_markets': ['CA', 'US'], 'name': 'I.
4	already missing you	Driving all night just to say goodbye Windows ...	prince royce, selena gomez	2013-10-08	{'available_markets': ['AD', 'AE', 'AG', 'AL',.

```
print(df[['album']])
df.drop(['album'], axis=1, inplace=True)
```

```
                                                album
0    {'available_markets': ['AD', 'AE', 'AG', 'AL',...
1    {'available_markets': ['AD', 'AE', 'AG', 'AL',...
2    {'available_markets': ['AD', 'AE', 'AG', 'AL',...
3    {'available_markets': ['CA', 'US'], 'name': 'I...
4    {'available_markets': ['AD', 'AE', 'AG', 'AL',...
..                                                 ...
159  {'available_markets': ['AD', 'AE', 'AG', 'AL',...
160  {'available_markets': ['AD', 'AE', 'AG', 'AL',...
161  {'available_markets': ['CA', 'MX', 'US'], 'nam...
162  {'available_markets': ['AD', 'AE', 'AG', 'AL',...
163  {'available_markets': ['AD', 'AE', 'AG', 'AL',...

[164 rows x 1 columns]
```

```
df.head(15)
```

	title	lyrics	artists	release_day	popularity
0	999	Hace mucho tiempo que quiero decirte algo y no...	selena gomez, camilo	2021-08-26	73
1	a year without rain	Can you feel me when I think about you? With e...	selena gomez & the scene	2010-01-01	61
2	adiós	La-la-la-la, ah, ah, ah-ah, ah, la-ah La-la-la...	selena gomez	2021-03-12	57
3	all night long	Now that I have captured your attention I wann...	mary jane girls	1994-01-01	56
4	already missing you	Driving all night just to say goodbye Windows ...	prince royce, selena gomez	2013-10-08	39
5	anxiety	My friends, they wanna take me to the movies I...	julia michaels, selena gomez	2019-01-24	58
6	as a blonde	I was looking in the mirror Trying to find, a ...	selena gomez & the scene	2009-01-01	32
7	b.e.a.t.	It's a big bad world But I ain't ashamed I lik...	selena gomez	2013-01-01	44
8	back to you	Took you like a shot Thought that I could chas...	selena gomez	2018-05-10	76
9	bad blood	'Cause, baby, now we got bad blood You know it...	taylor swift	2014-10-27	74
10	bad girlfriend	I know I'm not there for you or there when you...	theory of a deadman	2008-04-01	67
11	bad liar	I was walking down the street the other day Tr...	selena gomez	2017-05-18	69
12	baila conmigo	Bebé, no sé si habla' mucho español Si entiend...	selena gomez, rauw alejandro	2021-03-12	73
13	bang	My new boy used to be a model He looks way bet...	selena gomez & the scene	2011-01-01	48
14	bang a drum	You caught my eye and I'm tryin' to holler at ...	selena gomez	2008-08-26	38

```
df['title'].count()
```

```
164
```

```
df.count()
```

```
title          164
lyrics         164
artists        164
release_day    164
```

```
popularity     164
dtype: int64
```

```
df.shape
```

```
(164, 5)
```

```
# Title That contains selena gomez in it
# Result:- There are total 164 titles out of which 135 titles have name of selena gomez in it
print(df[df['artists'].str.contains("selena gomez", case=False)].count())
```

```
title          135
lyrics         135
artists        135
release_day    135
popularity     135
dtype: int64
```

```
# Printing the count of the data in which there is name of selena gomez in the artists column
df= df[df['artists'].str.contains("selena gomez", case=False)]
df.head()
df.count()
```

```
title          135
lyrics         135
artists        135
release_day    135
popularity     135
dtype: int64
```

```
df.head()
```

	title	lyrics	artists	release_day	popularity
0	999	Hace mucho tiempo que quiero decirte algo y no...	selena gomez, camilo	2021-08-26	63
1	a year without rain	Can you feel me when I think about you? With e...	selena gomez & the scene	2010-01-01	61
2	adiós	La-la-la-la, ah, ah, ah-ah, ah, la-ah La-la-la...	selena gomez	2021-03-12	57
4	already missing you	Driving all night just to say goodbye Windows ...	prince royce, selena gomez	2013-10-08	39
5	anxiety	My friends, they wanna take me to the movies I...	julia michaels, selena gomez	2019-01-24	58

```
df['release_day'] = pd.to_datetime(df['release_day'])
df['release_year'] = df['release_day'].dt.year
df['release_month'] = pd.to_datetime(df['release_day']).dt.to_period('M')
```

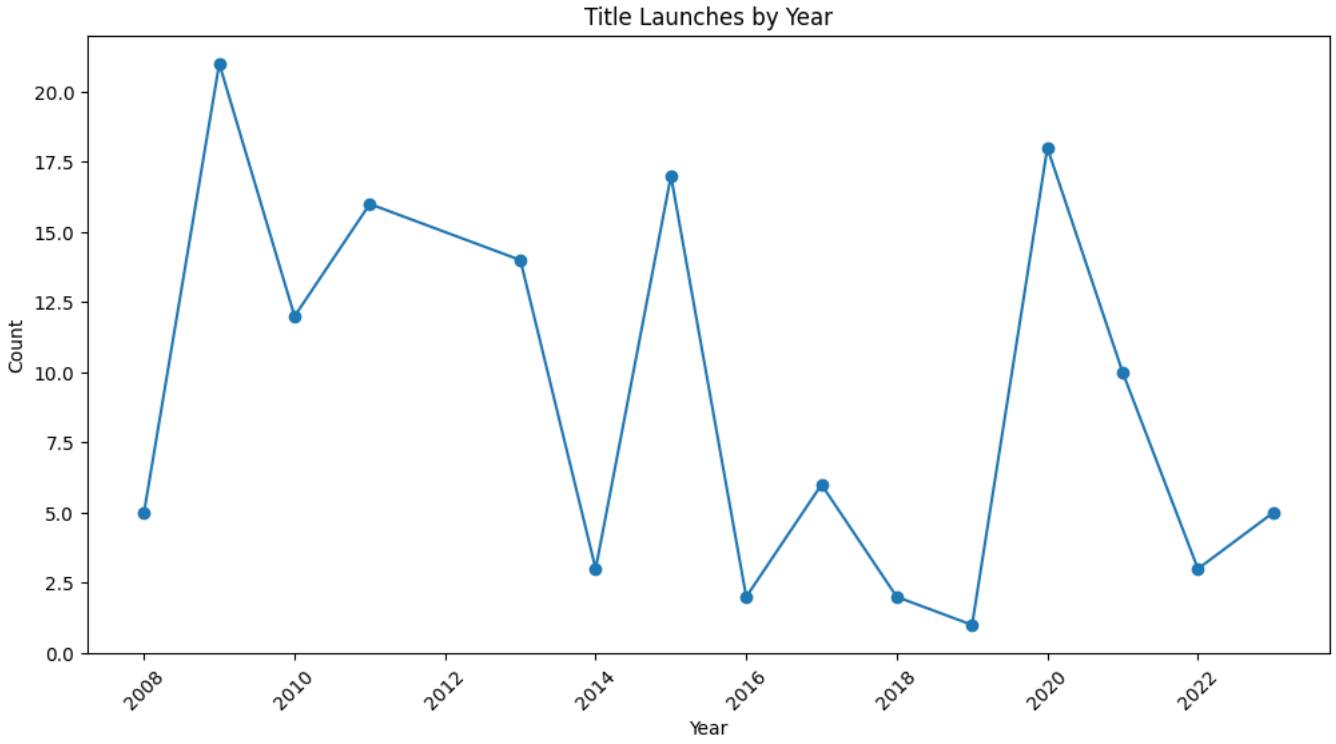
```
<ipython-input-104-bf5c597d8038>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-
  df['release_day'] = pd.to_datetime(df['release_day'])
<ipython-input-104-bf5c597d8038>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-
  df['release_year'] = df['release_day'].dt.year
<ipython-input-104-bf5c597d8038>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-
  df['release_month'] = pd.to_datetime(df['release_day']).dt.to_period('M')
```

```
df.head()
```

	title	lyrics	artists	release_day	popularity	release_year	relea
0	999	Hace mucho tiempo que quiero decirte algo y no...	selena gomez, camilo	2021-08-26	63	2021	
1	a year without rain	Can you feel me when I think about you? With e...	selena gomez & the scene	2010-01-01	61	2010	

```
#Counting release of title yearly & plotting into the chart
title_counts_by_year = df.groupby(df['release_year'])['title'].count()
print(title_counts_by_year)
# Plotting the line chart
plt.figure(figsize=(12, 6))
plt.plot(title_counts_by_year.index, title_counts_by_year.values, marker='o')
plt.xlabel('Year')
plt.ylabel('Count')
plt.title('Title Launches by Year')
plt.xticks(rotation=45)
plt.show()
```

```
release_year
2008     5
2009    21
2010    12
2011    16
2013    14
2014     3
2015    17
2016     2
2017     6
2018     2
2019     1
2020    18
2021    10
2022     3
2023     5
Name: title, dtype: int64
```




```
titles_2008 = df.loc[(df['release_year'] == 2008), 'title']
print(titles_2008)
```

```
14                        bang a drum
31                      cruella de vil
45                   fly to your heart
98                        new classic
137    tell me something i dont know
Name: title, dtype: object
```

```
# Find most common words used for lyrics from all the 135 songs
```

```
#finding the average of released titles according to the release year !
# Group the data by 'release_year' and calculate the average popularity count
average_popularity_by_year = df.groupby('release_year')['popularity'].mean()

# Create a new DataFrame to hold the average yearly popularity count
average_popularity_df = pd.DataFrame({'Year': average_popularity_by_year.index, 'Average Popularity': average_popularity_by_year.v

# Display the average yearly popularity count DataFrame
print(average_popularity_df)
```

```
    Year  Average Popularity
0   2008           47.400000
1   2009           40.666667
2   2010           40.583333
3   2011           56.750000
4   2013           49.142857
5   2014           54.000000
6   2015           55.058824
7   2016           59.500000
8   2017           65.000000
9   2018           77.000000
10  2019           58.000000
11  2020           61.666667
12  2021           55.000000
13  2022           66.666667
14  2023            1.400000
```

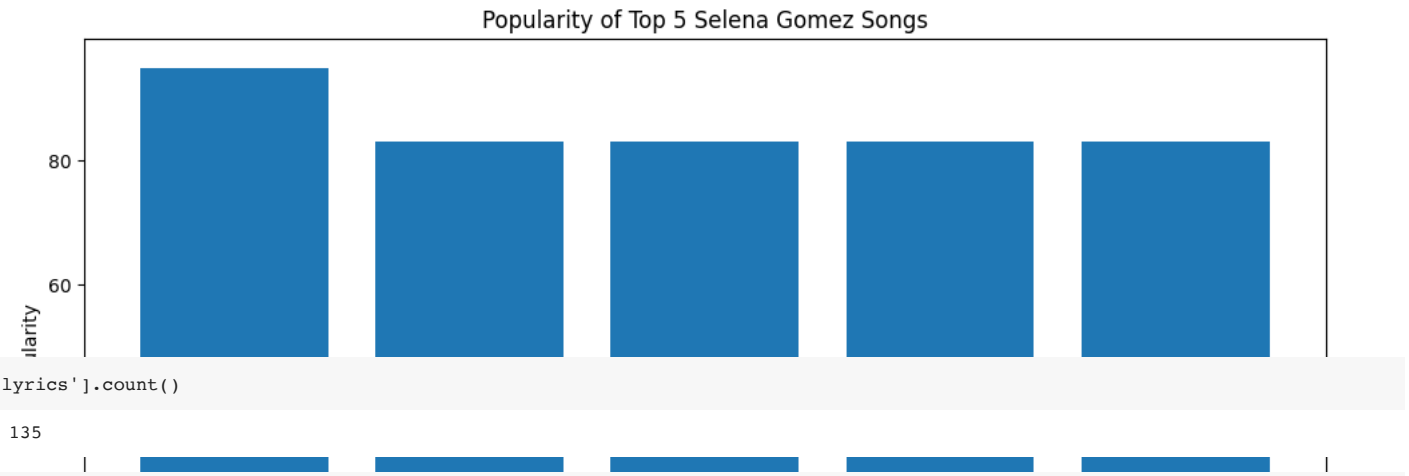
```
#Filtering till year 2022 only as found record is till year 2022 only !!
filtered_df = average_popularity_df[average_popularity_df['Year'] <= 2022]
print("Released titles according to year", title_counts_by_year)
print("Popularity according to year", filtered_df)
# Plotting the line chart for average popularity according to year (up to 2022)
plt.figure(figsize=(12, 6))
plt.plot(filtered_df['Year'], filtered_df['Average Popularity'], marker='o')
plt.xlabel('Year')
plt.ylabel('Average Popularity')
plt.title('Average Popularity of Selena Gomez\'s Titles by Year (up to 2022)')
plt.xticks(rotation=45)
plt.show()
```

```
Released titles according to year release_year
2008     5
2009    21
2010    12
2011    16
2013    14
2014     3
2015    17
2016     2
2017     6
2018     2
2019     1
2020    18
2021    10
2022     3
2023     5
Name: title, dtype: int64
Popularity according to year     Year  Average Popularity
0   2008           47.400000
1   2009           40.666667
2   2010           40.583333
3   2011           56.750000
4   2013           49.142857
5   2014           54.000000
6   2015           55.058824
7   2016           59.500000
8   2017           65.000000
9   2018           77.000000
10  2019           58.000000
11  2020           61.666667
12  2021           55.000000
```

```
sorted_data = df.sort_values('popularity', ascending=False)

# Filter the top 5 titles with highest popularity
top_5_popular_titles = sorted_data.head(5)

# Display the filtered DataFrame
print(top_5_popular_titles[['title', 'popularity']])
```

```
                       title  popularity
160               calm down          95
79      lose you to love me          83
151  when the sun goes down          83
118            sad serenade          83
81      love me like you do          83
```

```
sorted_data = top_5_popular_titles.sort_values('popularity', ascending=False)

# Plotting the bar chart for popularity of the top 5 songs
plt.figure(figsize=(12, 6))
plt.bar(sorted_data['title'], sorted_data['popularity'])

plt.xlabel('Song')
plt.ylabel('Popularity')
plt.title('Popularity of Top 5 Selena Gomez Songs')
plt.xticks(rotation=45)
plt.show()
```




```
df['lyrics'].count()
```

```
135
```

```
df.head()
```

	title	lyrics	artists	release_day	popularity	release_year	relea
0	999	Hace mucho tiempo que quiero decirte algo y no...	selena gomez, camilo	2021-08-26	63	2021	
1	a year without rain	Can you feel me when I think about you? With e...	selena gomez & the scene	2010-01-01	61	2010	
2	adiós	La-la-la-la, ah, ah, ah-ah, ah, la-ah La-la-la...	selena gomez	2021-03-12	57	2021	
4	already missing you	Driving all night just to say goodbye Windows ...	prince royce, selena gomez	2013-10-08	39	2013	
5	anxiety	My friends, they wanna take me to the movies I...	julia michaels, selena gomez	2019-01-24	58	2019	

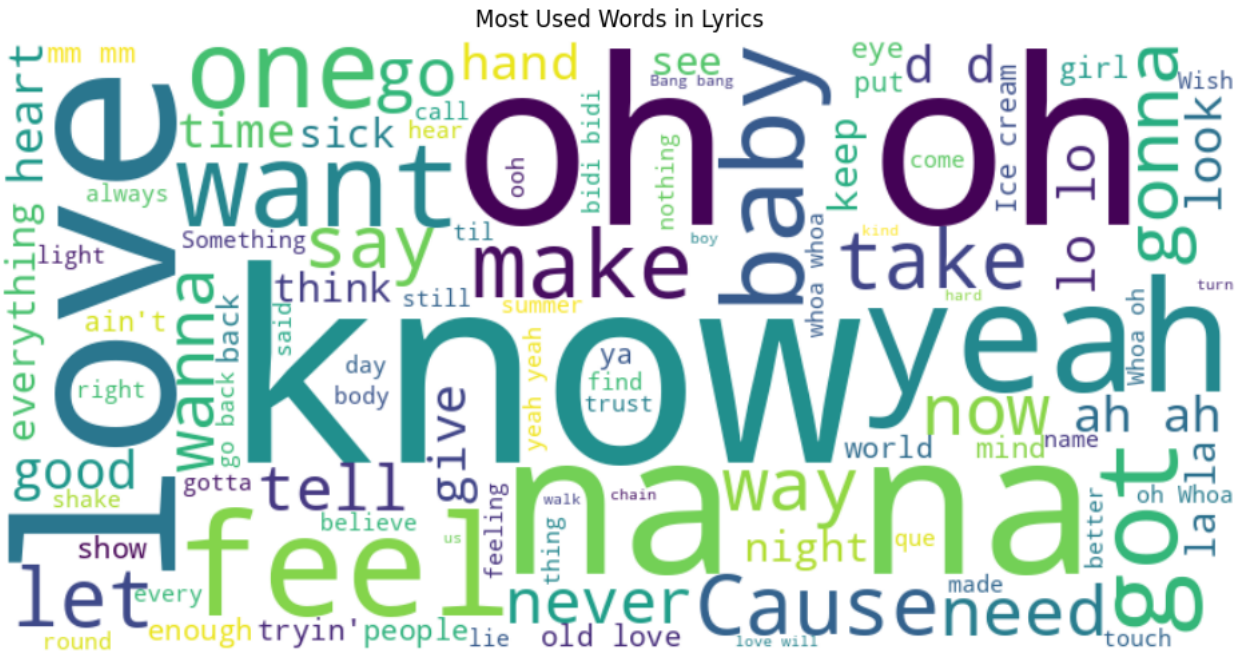
```
from wordcloud import WordCloud
```

```
#Finding words that are highest used by selena gomez from the lyrics of her 135 titles
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Assuming you have a DataFrame named 'df' containing the lyrics column

# Combine all the lyrics into a single string
all_lyrics = ' '.join(df['lyrics'])

# Generate a word cloud
wordcloud = WordCloud(width=800, height=400, max_words=100, background_color='white').generate(all_lyrics)

# Get the word frequencies from the word cloud
word_frequencies = wordcloud.process_text(all_lyrics)

# Sort the words based on frequency in descending order
sorted_words = sorted(word_frequencies.items(), key=lambda x: x[1], reverse=True)

# Print the most used words and their frequencies
for word, frequency in sorted_words[:10]:
    print(f"{word}: {frequency}")
```

```
know: 466
oh oh: 341
love: 304
na na: 264
yeah: 209
feel: 178
want: 175
baby: 157
got: 151
one: 141
```

```
# Plot the word cloud
plt.figure(figsize=(12, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
```

```
plt.title('Most Used Words in Lyrics')
plt.show()
```




```
# count of songs for the highest used words from the lyrics
# Combine all the lyrics into a single string
all_lyrics = ' '.join(df['lyrics'])

# Generate a word cloud
wordcloud = WordCloud(width=800, height=400, max_words=100, background_color='white').generate(all_lyrics)

# Get the word frequencies from the word cloud
word_frequencies = wordcloud.process_text(all_lyrics)

# Sort the words based on frequency in descending order
sorted_words = sorted(word_frequencies.items(), key=lambda x: x[1], reverse=True)

# Get the highest used words (top 10 in this case)
highest_used_words = [word for word, _ in sorted_words[:10]]

# Create a new column 'word_count' to store the count of each word in the lyrics
df['word_count'] = df['lyrics'].apply(lambda x: sum(1 for word in x.lower().split() if word in highest_used_words))

# Group the data by word and calculate the count of songs for each word
songs_count_by_word = df.groupby('word_count')['title'].count()

# Print the count of songs for each word
for word in highest_used_words:
    count = songs_count_by_word.get(len(word), 0)
    print(f"{word}: {count} songs")
```

```
know: 6 songs
oh oh: 4 songs
love: 6 songs
na na: 4 songs
yeah: 6 songs
feel: 6 songs
want: 6 songs
baby: 6 songs
got: 8 songs
one: 8 songs
```

✓ 2s completed at 3:34 AM