Maths + AI Project
Presentation

Coldplay Lyrics Analysis

Exploring Patterns in Coldplay's Discography

Team Name: After hours
Team Members:
Harshit Agrawal: harshit.agrawal@adypu.edu.in
Sakina Farukh Ahemad: sakina.ahemad@adypu.edu.in
Krish Jain: krish.jain@adypu.edu.in
Rounit Singh: rounit.singh@adypu.edu.in

1. **We begin by uploading the dataset ColdPlay.csv which contains information about Coldplay's songs, including lyrics, release years, and albums.**

📤 Uploading the Dataset

```python
import pandas as pd
import matplotlib.pyplot as plt
from textblob import TextBlob
from collections import Counter
from google.colab import files
```

Python

```python
uploaded = files.upload()
```

Python

```
Choose files  No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving ColdPlay.csv to ColdPlay (1).csv
```

```python
data = pd.read_csv('ColdPlay.csv')
```

2. Let's take a look at the first few rows to understand the structure and contents of the dataset.

👀 Previewing the Data

```python
data.head()
```

Python

	Unnamed: 0	Artist	Title	Album	Year	Date	Lyric
0	0	Coldplay	The Scientist	A Rush of Blood to the Head	2002.0	2002-08-26	come up to meet you tell you i'm sorry you don...
1	1	Coldplay	Viva la Vida	Viva La Vida or Death and All His Friends	2008.0	2008-05-25	chris martin i used to rule the world seas wou...
2	2	Coldplay	Fix You	X&Y	2005.0	2005-06-06	chris martin when you try your best but you do...
3	3	Coldplay	Yellow	Parachutes	2000.0	2000-06-26	chris martin look at the stars look how they s...
4	4	Coldplay	Hymn for the Weekend	A Head Full of Dreams	2016.0	2016-01-25	beyoncé and said drink from me drink from me o...

3. Here, we analyze how many songs were released by Coldplay each year. This helps us identify periods of high productivity and album launches.

📊 Number of Songs Released Each Year

```python
year_column = 'Year'
year_frequency = data[year_column].value_counts()
year_frequency.sort_values()
```

Python

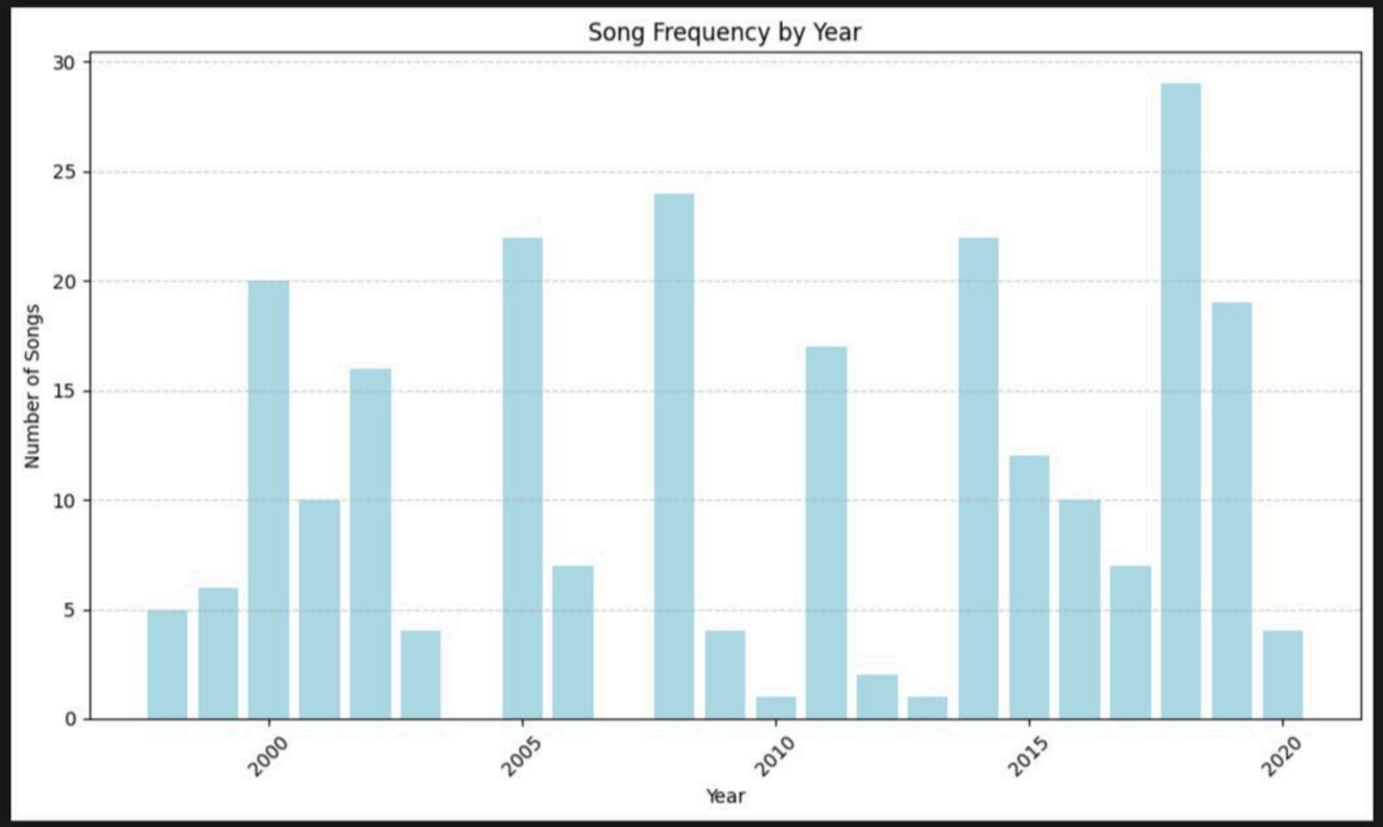
Year	count
2010.0	1
2013.0	1
2012.0	2
2020.0	4
2003.0	4
2009.0	4
1998.0	5
1999.0	6
2006.0	7
2017.0	7
2001.0	10
2016.0	10
2015.0	12
2002.0	16
2011.0	17
2019.0	19
2000.0	20
2014.0	22
2005.0	22
2008.0	24
2018.0	29

dtype: int64

Graphical Representation of how many songs were released by Coldplay each year

```
plt.figure(figsize=(10, 6))
plt.bar(year_frequency.index.astype(int), year_frequency.values, color='lightblue')
plt.xlabel('Year')
plt.ylabel('Number of Songs')
plt.title('Song Frequency by Year')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

Pytho



4. We visualize how many songs are part of each album. This gives insights into which albums are the most extensive or possibly the most promoted.

💿 Number of Songs in Each Album

```python
Album_column = 'Album'
Album_frequency = data[Album_column].value_counts()
Album_frequency.sort_values()
```

Python

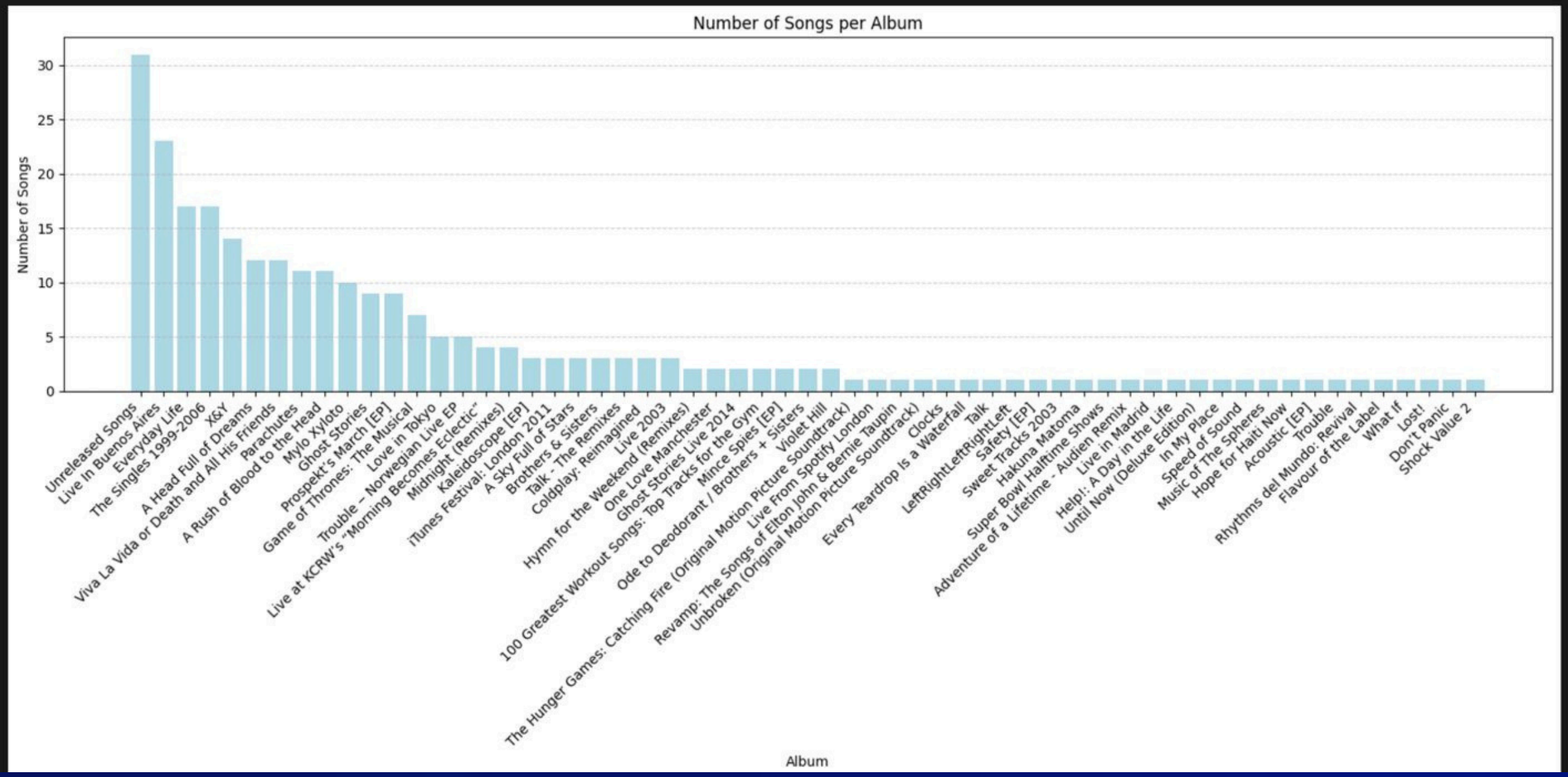
...

Album	count
The Hunger Games: Catching Fire (Original Motion Picture Soundtrack)	1
Until Now (Deluxe Edition)	1
Help!: A Day in the Life	1
Live in Madrid	1
Adventure of a Lifetime - Audien Remix	1
Shock Value 2	1
Don't Panic	1
Lost!	1
What If	1
Flavour of the Label	1
Rhythms del Mundo: Revival	1
Trouble	1
Acoustic [EP]	1
Hope for Haiti Now	1
Music of The Spheres	1
Speed of Sound	1
In My Place	1
LeftRightLeftRightLeft	1
Talk	1
Every Teardrop Is a Waterfall	1
Clocks	1
Super Bowl Halftime Shows	1
Hakuna Matoma	1
Sweet Tracks 2003	1
Safety [EP]	1

Graphical Representation how many songs are part of each album

```python
plt.figure(figsize=(16, 8))
plt.bar(Album_frequency.index, Album_frequency.values, color='lightblue')
plt.xlabel('Album')
plt.ylabel('Number of Songs')
plt.title('Number of Songs per Album')
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```

Python



5. Using the TextBlob library, we calculate sentiment polarity for each song's lyrics. The sentiment ranges from -1 (very negative) to +1 (very positive).

Sentiment Analysis of Lyrics

```
data['sentiment'] = data['Lyric'].apply(lambda x: TextBlob(str(x)).sentiment.polarity)
print(data[['Title', 'sentiment']].head(10))
```

Python

```
...                      Title  sentiment
0                The Scientist   0.102941
1                 Viva la Vida   0.119767
2                      Fix You   0.035111
3                       Yellow   0.243056
4          Hymn for the Weekend   0.028201
5           A Sky Full of Stars   0.259091
6                     Everglow  -0.055284
7      Adventure of a Lifetime   0.140196
8                      Orphans   0.021053
9                     Paradise   0.060000
```

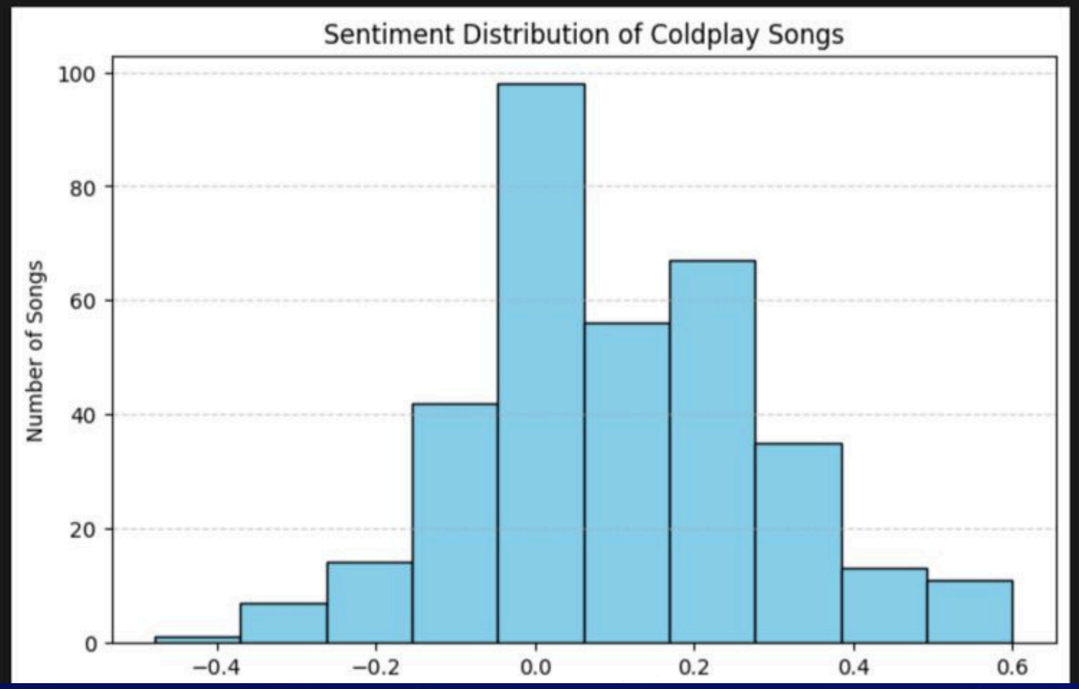
Graphical Representation sentiment polarity for each song's lyrics

Sentiment Distribution Across Songs

```python
plt.figure(figsize=(8, 5))
plt.hist(data['sentiment'], bins=10, color='skyblue', edgecolor='black')
plt.title('Sentiment Distribution of Coldplay Songs')
plt.xlabel('Sentiment Polarity')
plt.ylabel('Number of Songs')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.show()
```

Python

...



6. We create a word cloud to visualize the most frequently used words in Coldplay's lyrics, giving us a quick thematic sense of their music.

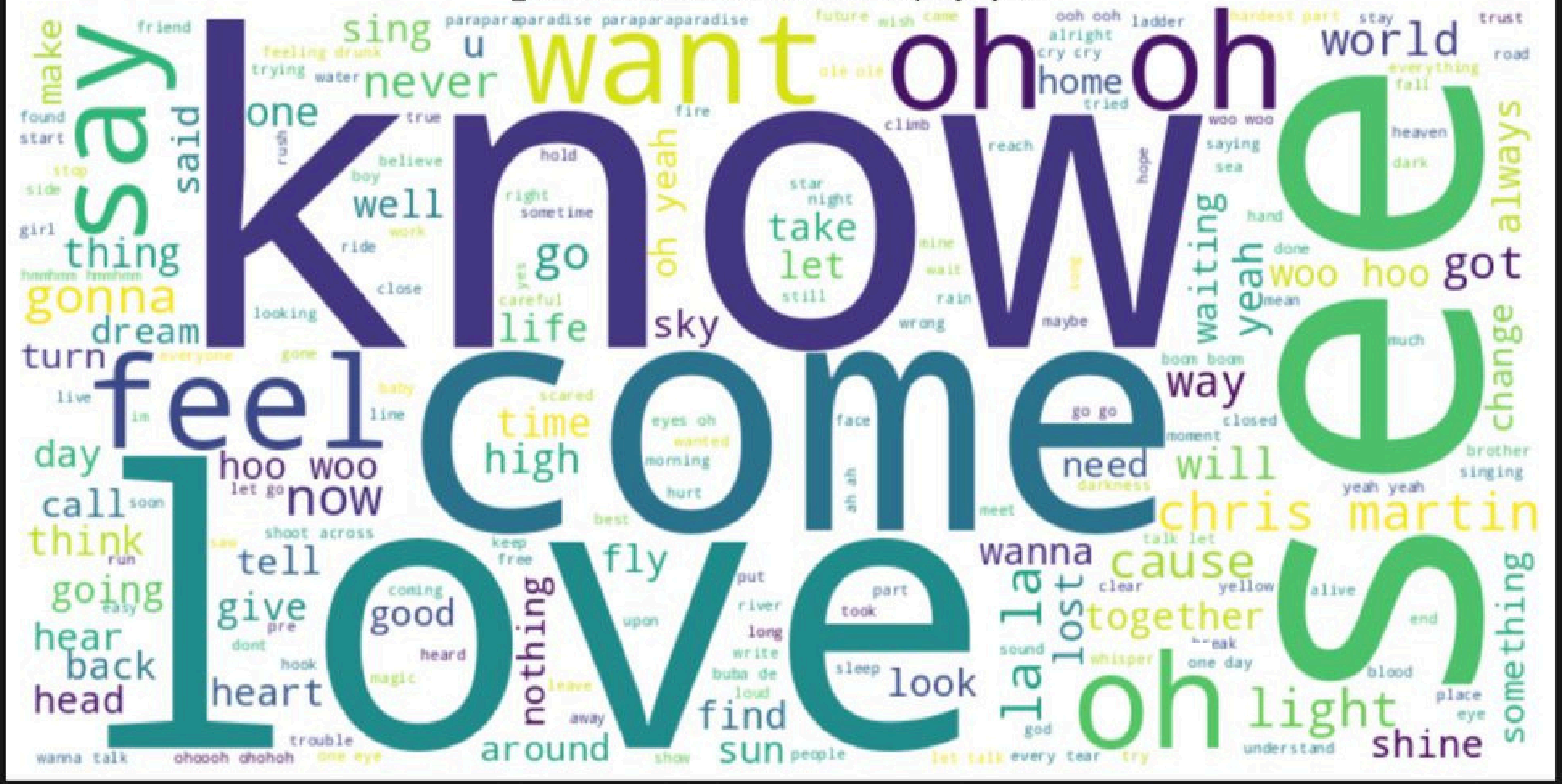
Word Cloud of Most Common Words

```python
all_lyrics = " ".join(str(lyric) for lyric in df_cleaned['Lyric'].dropna()).lower()

wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white',
    colormap='viridis'
).generate(all_lyrics)

plt.figure(figsize=(15, 7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("🌀 Most Common Words in Coldplay Lyrics", fontsize=16)
plt.tight_layout()
plt.show()
```

Python



7. We examine in which months Coldplay tends to release music. This can uncover patterns such as seasonal releases or end-of-year drops.

📅 Release Month Distribution

```python
if 'Date' in data.columns:
    data['Month'] = pd.to_datetime(data['Date'], errors='coerce').dt.month
    print(data['Month'].value_counts().sort_index())
```

Python

```
Month
1.0      5
2.0      8
3.0     14
4.0     10
5.0     24
6.0     31
7.0     15
8.0     21
9.0      8
10.0    23
11.0    33
12.0    50
Name: count, dtype: int64
```

8. This section shows the number of words in each song's lyrics. It helps identify which songs are longer or more lyrically dense.

📝 Lyrics Word Count per Song

```python
data['word_count'] = data['Lyric'].apply(lambda x: len(str(x).split()))
print(data[['Title', 'word_count']].sort_values(by='word_count', ascending=False))
```

Python

```
                          Title  word_count
125  Game of Thrones: The Musical        1780
184  Up&Up (Live in Buenos Aires)         607
15                         Up&Up         495
203                   Billie Jean         467
129          Bitter Sweet Symphony         417
..                             ...         ...
230                         Lukas           1
255                          Aiko           1
234            Famous Old Painters           1
322  Midnight (Jon Hopkins Remix)           1
342                Family Portrait           1

[344 rows x 2 columns]
```

🔍 KEY INSIGHTS FROM THE ANALYSIS

- Coldplay has consistently released songs over the years with notable peaks during album cycles.
- Albums like Parachutes and A Head Full of Dreams have a higher number of songs.
- Common lyric themes include “love,” “light,” “sky,” and “heart,” suggesting an emotional and uplifting musical style.

👥 TEAM CONTRIBUTIONS

- Harshit Agrawal: Led the exploratory data analysis (EDA), styled the visualizations, and managed the GitHub upload.
- Sakina Farukh Ahemad: Focused on cleaning the data and planning the visual narratives.
- Krish Jain: Designed the final presentation and helped summarize the key findings.
- Rounit Singh: Created the word cloud and handled the text-based sentiment analysis.

-

THANK YOU 🙏🙏

SEE YOU IN NEXT SEMESTER SIR :)