# Dataset Loader: Genius Song Lyrics (Hugging Face)

**Data Source:** https://huggingface.co/datasets/sebastiandizon/genius-song-lyrics

**Description:** The original Genius Song Lyrics dataset contains 2.76 million songs (~9 GB CSV file). To enable lightweight experimentation, this script allows you to download and save a smaller random subset (e.g., 1%) locally as CSV.

---

# 1. Preparations

## 1.1 Import Packages

In [1]:
```
from datasets import load_dataset
import pandas as pd
import os
```

## 1.2 Load original Dataset from Hugging Face

Doesn't load full Dataset but Metadata.

In [2]:
```
dataset = load_dataset("sebastiandizon/genius-song-lyrics", split="train")
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP downlo
ad. For better performance, install the package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
song_lyrics%202.csv:  81%|########1 | 7.36G/9.07G [00:00<?, ?B/s]
```

```
C:\Users\pelle\SynologyDrive\Projekte\Text-Analytics-Project\.venv\Lib\site-packages\huggingface_hub\file_download.py:143: UserWarning: `huggingface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\pelle\.cache\huggingface\hub\datasets--sebastiandizon--genius-song-lyrics. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development
  warnings.warn(message)
Generating train split: 0 examples [00:00, ? examples/s]
Loading dataset shards:   0%|          | 0/19 [00:00<?, ?it/s]
```

## 1.2 Configuration

- Define the subset percentage to download (e.g., 1%, 5%, 10%). Note: Hugging Face only accepts integer percentages for split notation.
- Define output directory and file name.
- Create directory if it doesn't exist.

In [3]:
```python
subset_fraction = 1
subset_size = int(len(dataset) * subset_fraction / 100)

output_dir = "data/raw"
os.makedirs(output_dir, exist_ok=True)

output_path = os.path.join(output_dir, f"lyrics_subset_{subset_fraction}pct.csv")
```

# 2. Load and Save Subset

## 2.1 Load Subset of Dataset

- Set Seed for reproducibility.
- Take a random sample of the Dataset.

In [4]:
```
dataset = dataset.shuffle(seed=42)

dataset_small = dataset.select(range(subset_size))

print(f"Dataset loaded successfully with {len(dataset_small):,} entries.")
```

```
Dataset loaded successfully with 51,348 entries.
```

## 2.2 Convert to pandas DataFrame

In [5]:
```
df = dataset_small.to_pandas()

print(f"DataFrame shape: {df.shape}")
print(f"Number of Songs: {len(df):,} | Artists: {df['artist'].nunique():,} | Genres: {df['tag'].nunique():,}")
```

```
DataFrame shape: (51348, 11)
Number of Songs: 51,348 | Artists: 39,461 | Genres: 6
```

## 2.3 Save Subset locally

In [6]:
```
df.to_csv(output_path, index=False)

print(f"Subset saved to: {output_path}")
```

```
Subset saved to: data/raw\lyrics_subset_1pct.csv
```

# 3. Preview of the dataset

Showing the first few rows of the dataset and the distribution of genres.

In [7]:
```
df.head()
```

Out[7]:

| | title | tag | artist | year | views | features | lyrics | id | language_cld3 | language_ft | language |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 Is Better | rap | Chris Travis | 2017 | 4437 | {} | [Intro]\nBitch I'm clean\nTwo sticks like Chow... | 3036329 | en | en | en |
| **1** | Scottie | rap | KrJ | 2012 | 89 | {} | My old girl left me on her old bull shit\nSo I... | 72180 | en | en | en |
| **2** | Pirate Password | rock | The never land pirate band | 2011 | 175 | {} | [Intro: spoken]\nAvast there matey haha\nIf a ... | 2122100 | en | en | en |
| **3** | Indri | rock | Puta Volcano | 2015 | 14 | {} | Just throw a glimpse under the shell\nGhostly ... | 6889288 | en | en | en |
| **4** | Maps | misc | ANBARDA | 2018 | 4 | {} | [Verse 1]\nI miss the taste of a sweeter life\... | 3735887 | en | en | en |

In [8]:
```
print("\nGENRE DISTRIBUTION")
print("=" * 60)
category_counts = df['tag'].value_counts().sort_values(ascending=False)

for tag,count in category_counts.items():
    pct = (count / len(df)) * 100
    print(f"{tag}: {count:,} songs ({pct:.2f}%)")
```

```
GENRE DISTRIBUTION
============================================================
pop: 21,438 songs (41.75%)
rap: 17,175 songs (33.45%)
rock: 8,001 songs (15.58%)
rb: 1,894 songs (3.69%)
misc: 1,860 songs (3.62%)
country: 980 songs (1.91%)
```

In [ ]: