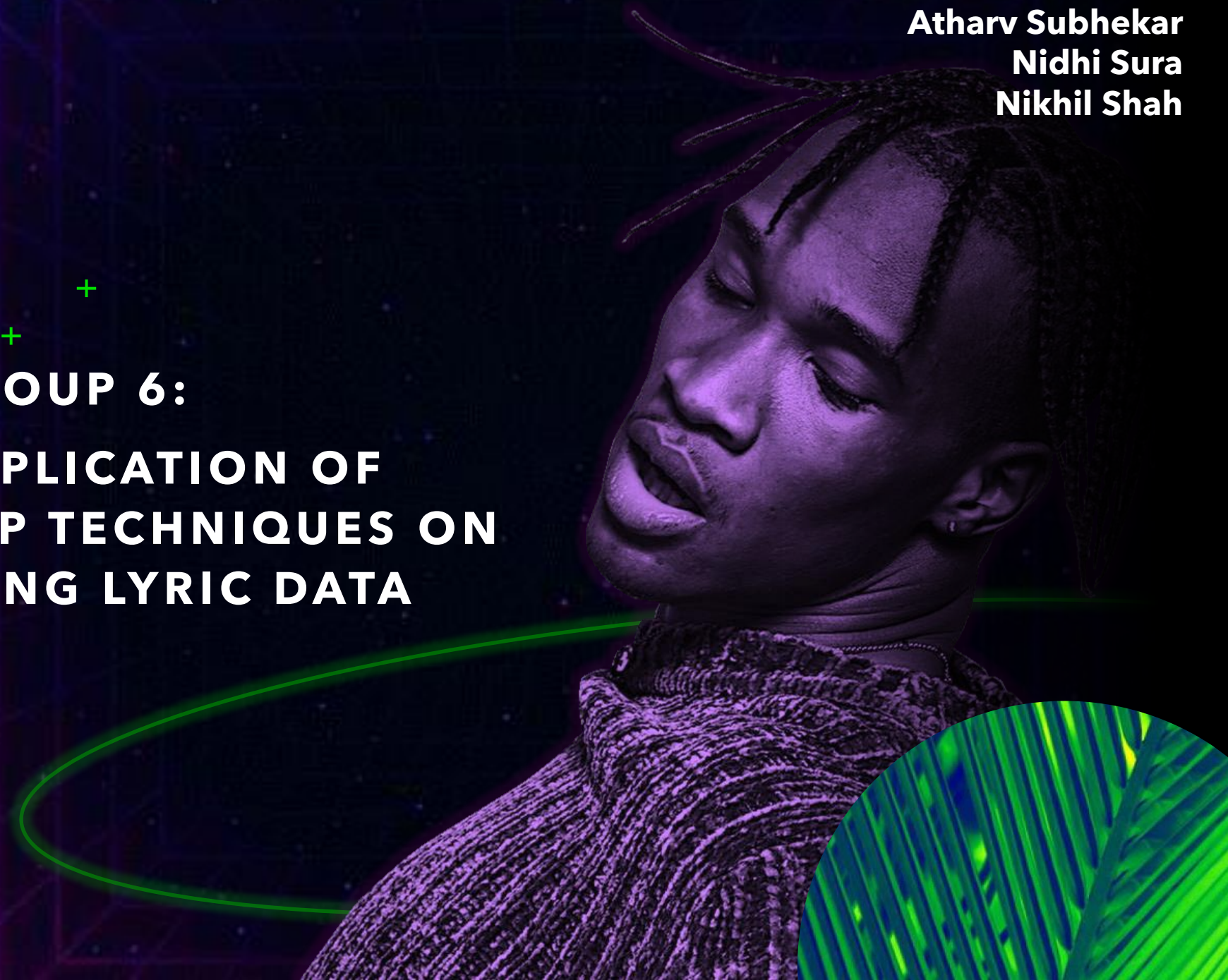
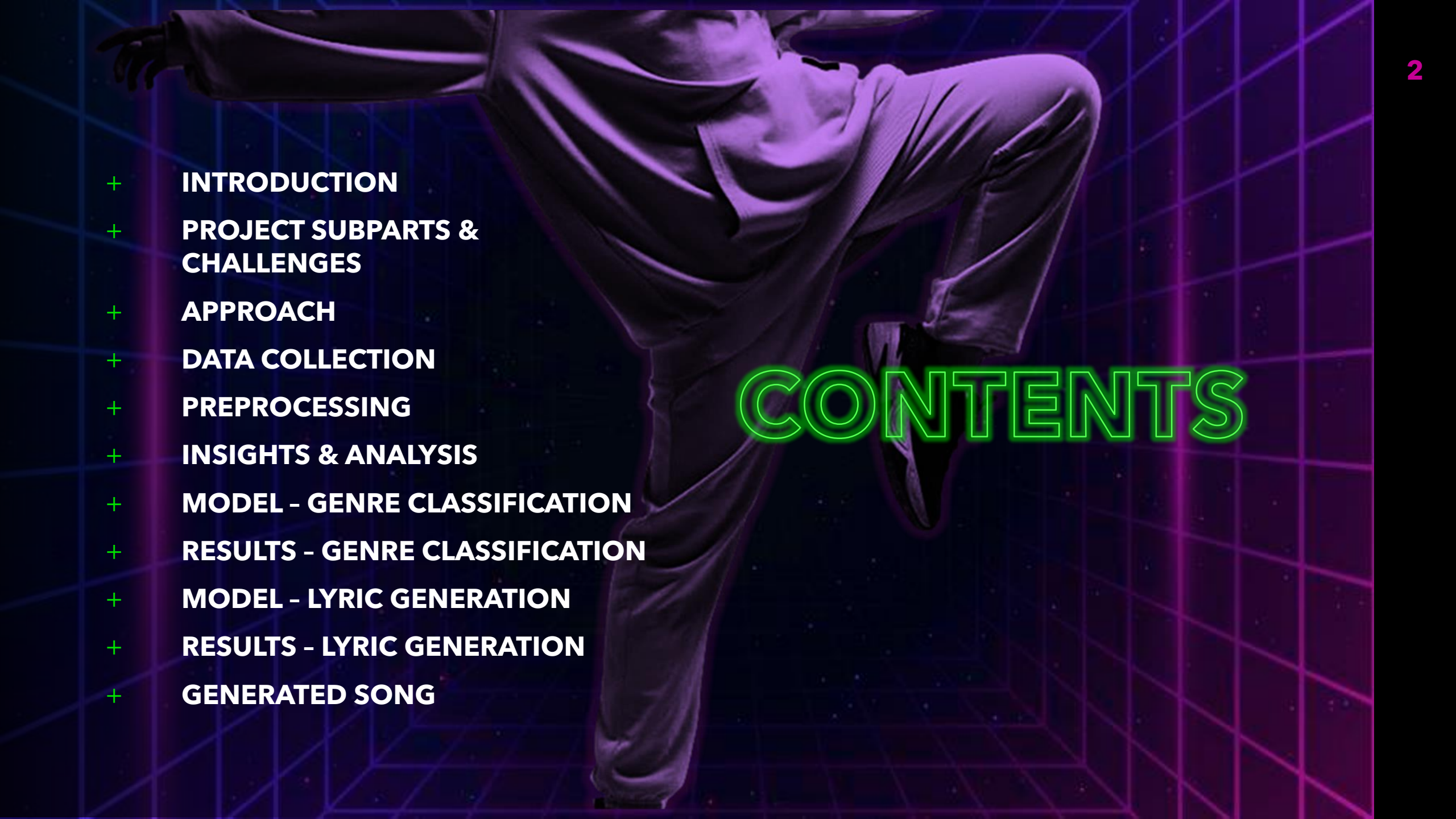


CS584 PROJECT

+ + +
GROUP 6:
**APPLICATION OF
NLP TECHNIQUES ON
SONG LYRIC DATA**

Atharv Subhekar
Nidhi Sura
Nikhil Shah



- 
- + **INTRODUCTION**
 - + **PROJECT SUBPARTS & CHALLENGES**
 - + **APPROACH**
 - + **DATA COLLECTION**
 - + **PREPROCESSING**
 - + **INSIGHTS & ANALYSIS**
 - + **MODEL - GENRE CLASSIFICATION**
 - + **RESULTS - GENRE CLASSIFICATION**
 - + **MODEL - LYRIC GENERATION**
 - + **RESULTS - LYRIC GENERATION**
 - + **GENERATED SONG**

CONTENTS

INTRODUCTION

+

+

In the dynamic world of music, a fresh perspective on genre classification has emerged—analyzing lyrics through sentiment analysis. Beyond traditional auditory features, this approach dives into the emotional essence of songs. By leveraging sentiment analysis, we explore how the sentiments expressed in lyrics can unveil distinctive emotional landscapes, paving the way for a nuanced understanding of various music genres. This brief journey explores the synergy between lyrics and sentiment analysis, revealing how emotional tones contribute to the unique identity of each musical genre.

GENRE CLASSIFICATION

- + Genre ambiguity
- + Newly emerging and evolving genres
- + Multiple-genre influences

LYRIC GENERATION

- + Maintaining genre authenticity
- + Incorporating genre-specific vocabulary
- + Creative versatility (curb stereotyping)

PROJECT SUBPART CHALLENGES



APPROACH



Data Collection - We are using the lyrics dataset from Kaggle which has all the necessary information which is mainly song and artist names, song lyrics and their genres.



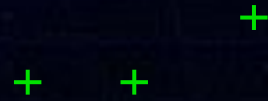
Data Preprocessing - We have implemented various techniques to convert the lyrics into tokens and dynamically labeled them using during



Data Statistics - We can analyze the basic statistics of the data set from the following histograms plots and wordclouds.



DATA COLLECTION



The data used for our project was a Kaggle Dataset,
made up of songs from 79 different genres

[Song lyrics from 79 musical genres](#)

All the data was obtained by scraping the Brazilian
website Vagalume using R.

PREPROCESSING

Stoplist and Symbols Removal

Text Cleaning Function: Created a **cleanText** function to strip whitespaces, replace newline and carriage return characters, and convert text to lowercase.

Data Loading, Deduplication and Filtering:

- Loaded artist data from a CSV file and removed duplicates based on a 'Link' column.
- Loaded lyrics data, renamed a column to match with the artist data, and then merged the two datasets on the 'Link' column.
- Filtered the merged dataset to retain only rows where the 'language' column is 'en' (English).

Genre Labeling:

- Mapped musical genres to numerical labels and applied this mapping to the 'Genres' column of the English-only dataset.
- Converted the genre labels into a NumPy array.

PREPROCESSING

Data Splitting

Data Parsing and Tokenization: Tokenized the lyrics data in the train, validation, and test sets using a **simple_tokenizer** function.

Vocabulary and GloVe Weight Matrix Construction:

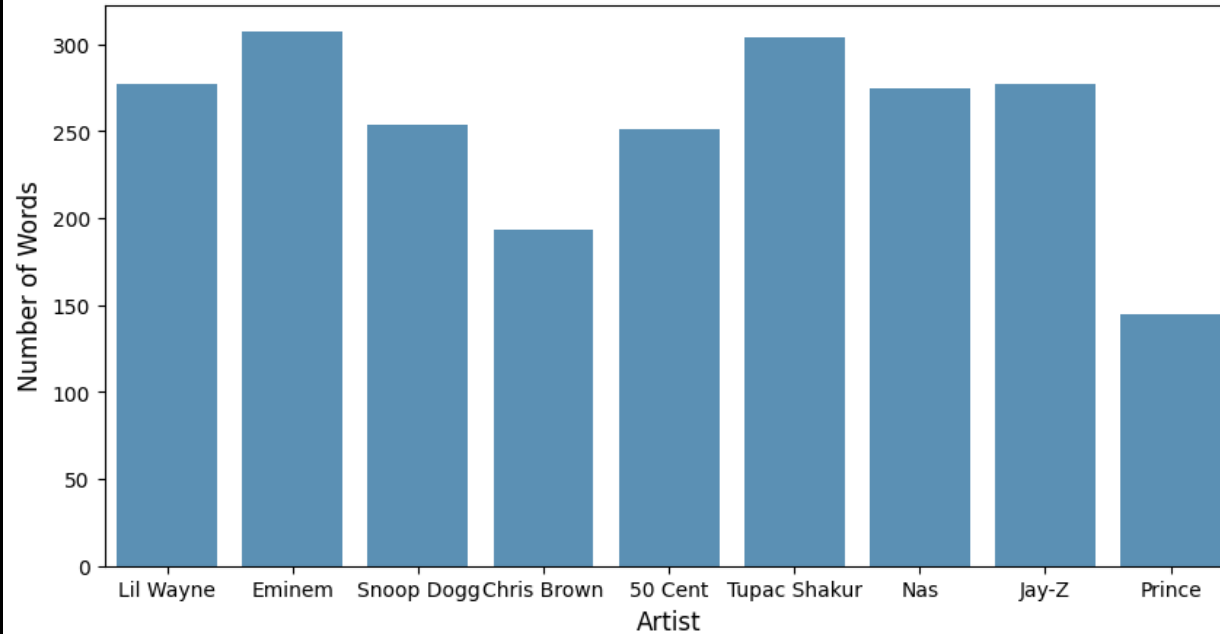
- Constructed a vocabulary using CountVectorizer on the tokenized training data, adjusting the vocabulary indices.
- Initialized a GloVe vectors matrix and populated it with vectors for each word in the vocabulary.

Data Indexing and Padding:

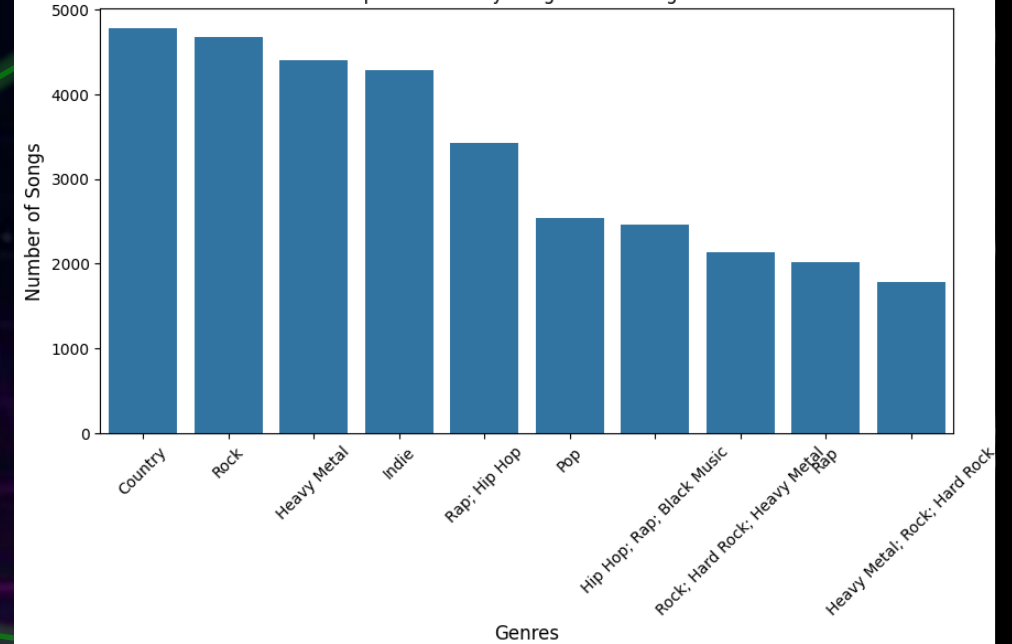
- Transformed the tokenized data into indices using the **doc_to_index** function.
- Padded the indexed data using the **pad_sequence** function for train, validation, and test datasets.

INSIGHTS & ANALYSIS

Top 10 Artists by Words/Song



Top 10 Genres by Song Count in English

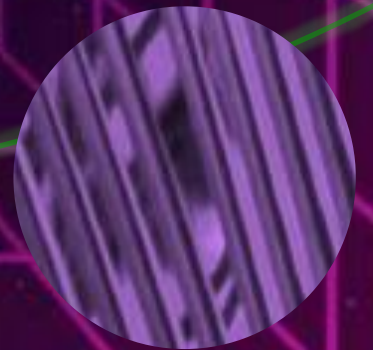


Axé; Dance; Pop/Rock Word Count



THE MODEL

- The motivation to use Long Short-Term Memory (LSTM) model stems from the fact that lyrics are inherently sequential in nature, and the similarity between two lyrics must in at least some way be determined by the similarities between their sequences over time.
- We have made use of dense vectors to represent words. For the purpose of this project, we will be used GloVe embeddings.
- To address overfitting since the corpus is 5000 lyrics we incorporated dropout rates.
- LSTM (Long Short-Term Memory) are very good for analyzing sequences of values and predicting the next one. For example, LSTM could be a good choice if you want to predict the very next point of a given time series.



THE MODEL

- **Embedding Layer:** The model starts with an embedding layer. If a pre-trained weight matrix (such as GloVe embeddings) is provided, it uses that for embedding; otherwise, it initializes a new embedding layer with a specified vocabulary size and embedding dimension.
- **LSTM Layer:** Following the embedding layer, the model employs an LSTM (Long Short-Term Memory) layer. LSTM is useful for processing sequences (like text) as it can capture long-range dependencies and contextual information. The LSTM layer has configurable parameters like hidden dimensions, number of layers, and dropout probability.

THE MODEL

- **Attention Mechanism:** The model includes a custom attention mechanism (**Attention** class) which computes attention scores based on the LSTM's outputs. This mechanism allows the model to focus on specific parts of the input sequence when making predictions, which is particularly useful for understanding the context and nuances in text data.
- **Dropout and Fully Connected Layer:** After the attention mechanism, the model applies dropout for regularization, followed by a fully connected linear layer to map the LSTM outputs to the desired output size.
- **Output:** The final output is designed to represent the sentiment of the input text. The model can be used for binary classification (positive/negative sentiment) or multi-class sentiment classification, depending on the specified output size.

RESULTS

- **Test Loss: 46.62%, Test Accuracy: 89.70%**



LYRIC GENERATION

- **Genre** - Country Music
- **Model** - LSTM based architecture (Sequential-LSTM-Dense)
- **Seed text** - Small town boy (very common country music phrase)
- **Training** - 20 epochs (batch size = 128)
- **Tokenization** - Implemented word & character; selected word tokenization
- Why such a simple model?
- Future work!

```
Epoch 1/10  
c:\Users\nidhi\OneDrive\Documents\USA\Stevens\Academics\CS584_NLP\Project\venv\Lib\site-packages\tensorflow\python\data\ops\structured_function.py:  
    warnings.warn(  
780/780 [=====] - 257s 329ms/step - loss: 7.3980 - accuracy: 0.0233 - val_loss: 7.2211 - val_accuracy: 0.0666  
---MODEL SAVED---
```

After epoch 1: Small town boy love im love im love im love love love love love love love love love love love love love love love love

[illegible]

LYRIC GENERATION BY EPOCH

Epoch 15/20

780/780 [=====] - 259s 330ms/step - loss: 4.2735 - accuracy: 0.2696 - val_loss: 7.5499 - val_accuracy: 0.1211

---MODEL SAVED---

After epoch 15: Small town boy id say goodbye goodbye dont know dont know dont know dont know dont know dont know dont know dont know dont know don

Epoch 15/20

780/780 [=====] - 259s 330ms/step - loss: 4.2735 - accuracy: 0.2696 - val_loss: 7.5499 - val_accuracy: 0.1211

---MODEL SAVED---

After epoch 15: Small town boy id say goodbye goodbye dont know dont know dont know dont know dont know dont know dont know dont know dont know don

Epoch 20/20

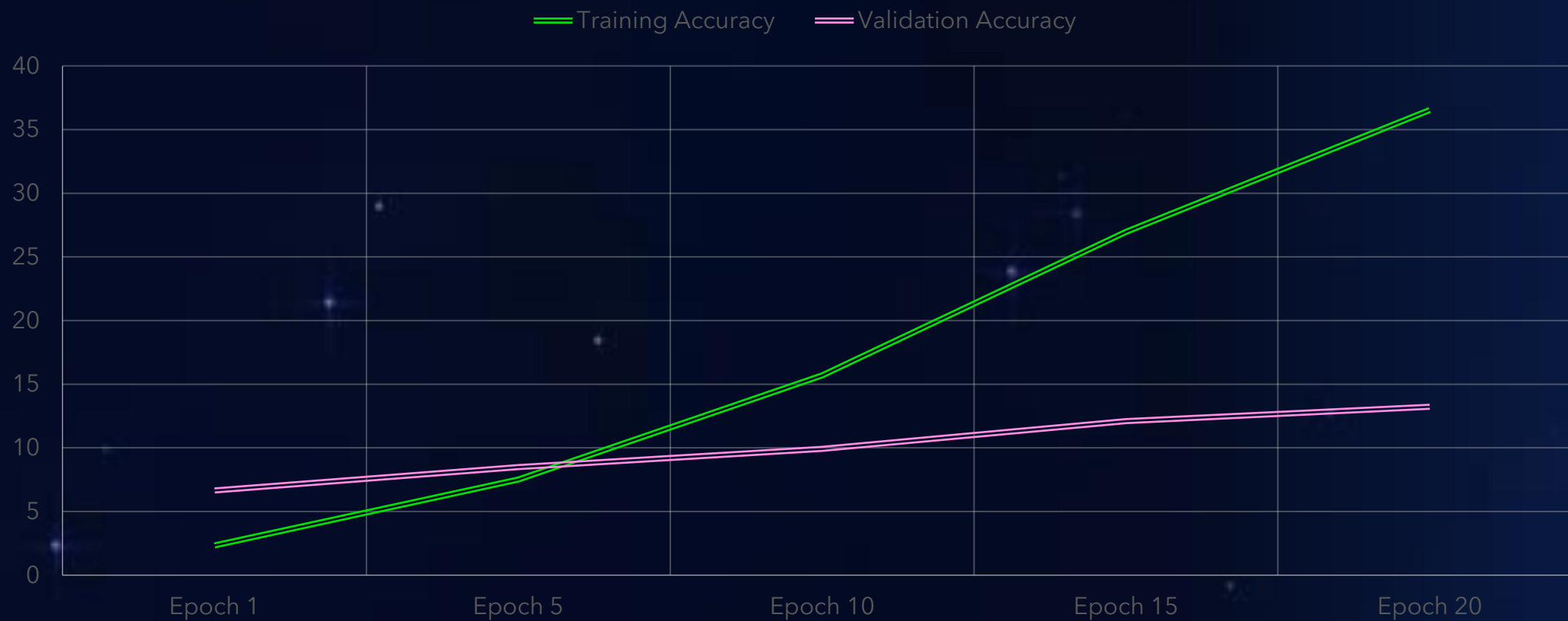
780/780 [=====] - 323s 412ms/step - loss: 3.5905 - accuracy: 0.3654 - val_loss: 7.9409 - val_accuracy: 0.1322

---MODEL SAVED---

After epoch 20: Small town boy man way back home small town southern man well im goin back town two kinds cherries two kinds fairies two kinds babi

MODEL TRAINING

ACCURACY BY EPOCH



GENERATED LYRICS

Epoch 1:

*Small town boy love
im love im love im love
love love love love
love love love love
love love love love
love love love love
love love love love
love love love love
love love love love
love love love love
love love love love
love love love love
love love love
love love love*

Epoch 20:

*Small town boy man
im gon na see nobody
thats gon na break rusty cage
run run back back back
tennessee homesick
midnight homesick blues
mama aint got ta go back
way back Tennessee
i'm goin back
tennessee homesick blues
got ta kick back relax
tennessee homesick blues
got ta kick back relax*

THANK YOU