

Custom Song Classification and Analysis

(Unlock the Emotions and Tailor Your Music)

By Group-15

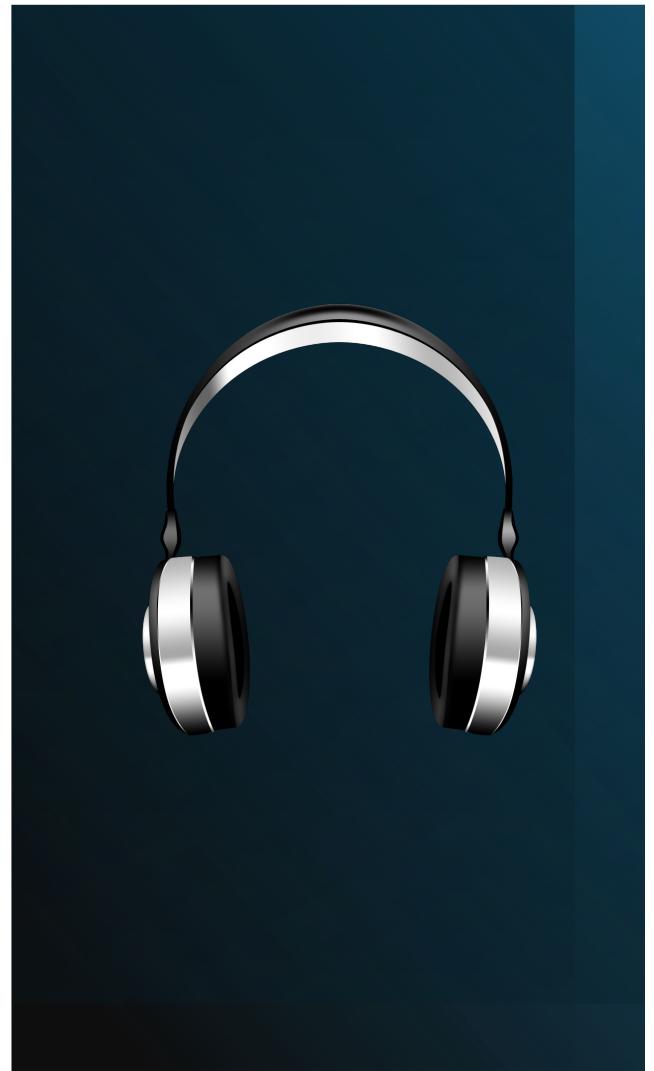
Naga Venkata Kanakalakshmi Murikipudi– 11725119

Naga Sai Sivani Tutika - 11703058



Agenda

- Introduction
- Problem Statement/Hypothesis
- Goals & Objectives
- Background & Uniqueness
- Methodology
- Dataset
- Exploratory Data Analysis
- Implementation
- Results
- Project Management
- Conclusion
- References



Introduction

- Nowadays, everyone is stressed out and searching for ways to unwind and get some peace of mind. One of the greatest things for many individuals to relax with is music. When we're feeling down, listening to the perfect music might help us feel better, forget our problems, and even improve our mood.
- **Custom Song Classification and Analysis (Unlock the Emotions and Tailor Your Music)** is an innovative project aimed at understanding the **emotional impact of songs**. By analyzing song lyrics and key features like danceability, energy, and valence, the system predicts the emotional tone of a song.
- It offers a personalized music experience that adapts to user preferences, going beyond basic **song recommendations**. The project not only categorizes songs by emotion but also **enhances user interaction** by suggesting songs based on the mood of the music. With language detection, translation, and dynamic recommendation algorithms, the system provides a more engaging and emotional music experience. It is **scalable and can handle large datasets**, ensuring a smooth experience for users.
- Two song datasets that are associated by a unique ID are used in the study. To identify each song's language, the datasets were combined using this ID. A cleaned subset of the original dataset was taken out after the language was determined, and it was used to power the recommendation system and do sentiment analysis.

Significance

The main significance of the project is providing an easy engage to the users to listen the music. As the music is a universal language which includes emotions can be expressed by lyrics - which are frequently ignored in the existing recommendation systems. We can improve the way music is categorized and suggested by using natural language processing (NLP) to analyze lyrical content. This allows us to go beyond basic acoustic parameters and concentrate on the true meaning of the song for better recommendations based on the emotions.

Background/Uniqueness

Existing Behavior:

- The current system identifies the emotion of a song using only a limited set of attributes, classifying songs based on their emotional tone.
- Popular songs are fetched based on the latest release dates and other basic attributes, with a recommendation system offering song suggestions based on user input.

Uniqueness:

- We introduced language identification for songs, which facilitates accurate translation.
- Developed a dynamic recommendation system that provides personalized song suggestions based on a given song and user input.
- Predicted the emotion of songs using a combination of attributes and lyrics.
- Highlighted the lyrics in sync with the song playing in the background.
- Implemented a scalable system capable of predicting the emotion of songs for approximately 500,000 records.
- We utilized various libraries and models to achieve optimal results in language detection, training, and testing the models.

Problem Statement

- Many songs available, it's a challenging to choose the perfect song for our current state of mind. Scrolling through endless music can be annoying, regardless of our emotions—from happiness to sadness or just a need for some more energy.
- Listeners often struggle to follow along with song lyrics while the music plays, making it difficult to sing along and fully engage with the song. This creates a gap in providing an interactive and immersive music experience.

Goals & Objectives

01

ENABLE LANGUAGE DETECTION FOR VARIABLE-LENGTH TEXTUAL DATA WITHOUT THE NEED FOR TRAINING, OPTIMIZED FOR LARGE DATASETS.

02

ENSURE SCALABLE DATA PROCESSING FOR TEXT ANALYSIS AND RETRIEVAL, CAPABLE OF HANDLING LARGE DATASETS EFFICIENTLY.

03

IMPLEMENT EMOTION PREDICTION FROM SONG LYRICS USING NATURAL LANGUAGE PROCESSING TECHNIQUES.

04

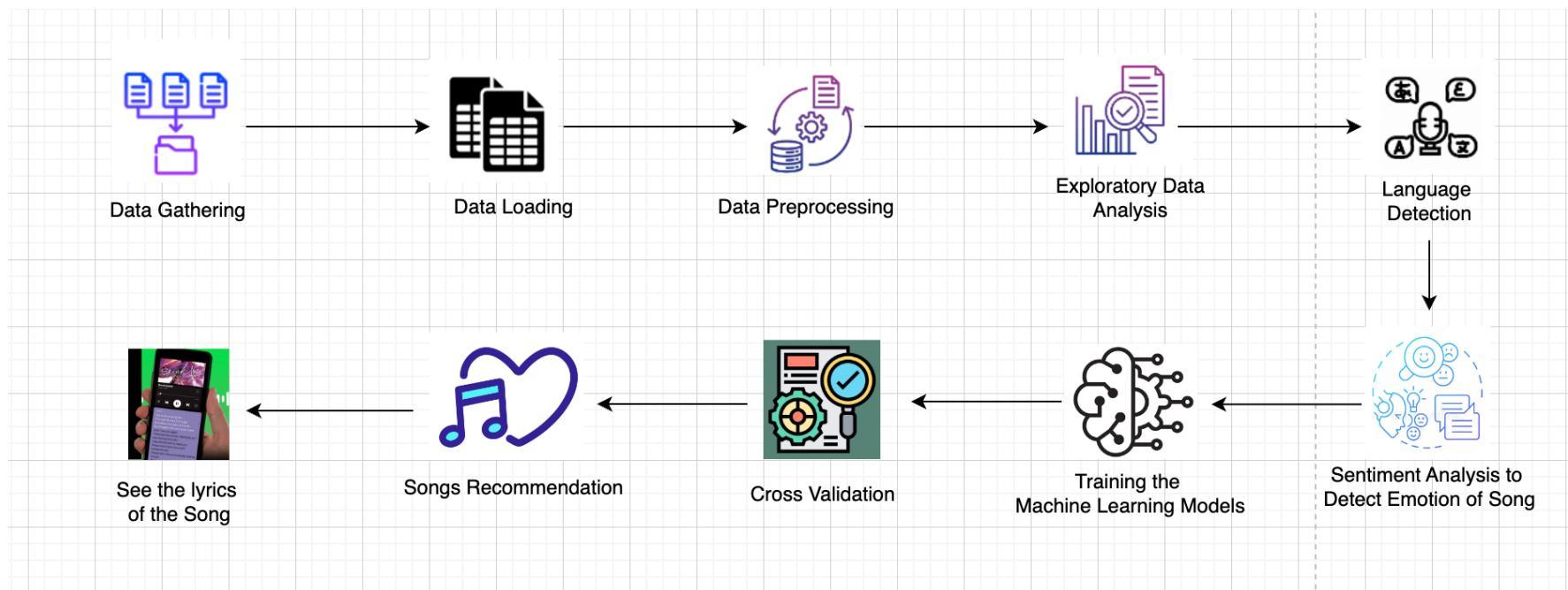
DEVELOP AN EFFICIENT MUSIC RECOMMENDATION SYSTEM BASED ON SONG ATTRIBUTES AND LYRICS.

05

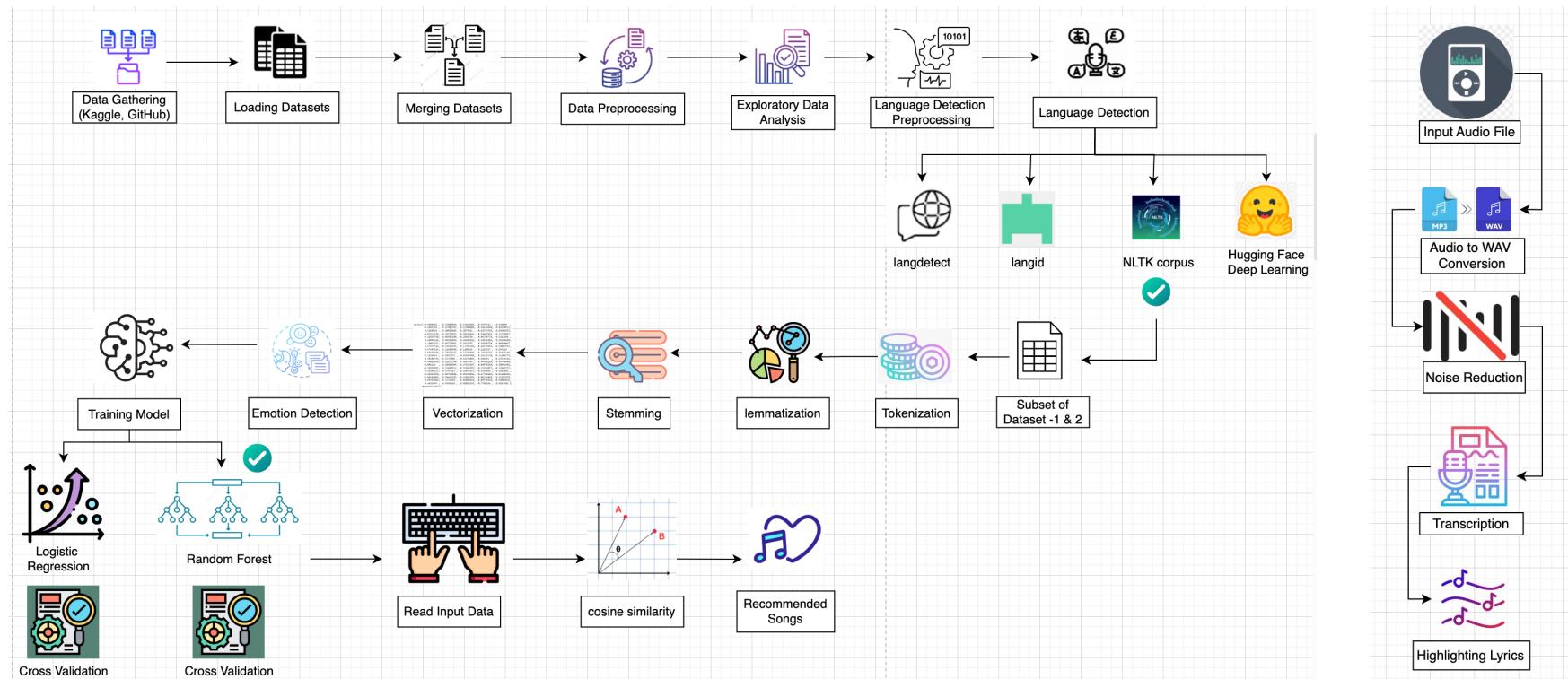
BUILD AN AUTOMATIC CAPTIONING SYSTEM THAT GENERATES CAPTIONS THROUGH SPEECH RECOGNITION FOR AUDIO DATA.

Methodology

Architecture Diagram (HLD)



Work-Flow Diagram



Architecture Models & Libraries Used

``langdetect`` Library

- ``langdetect`` library is an direct extraction of Google's language-detection library in Java into Python.
- It's trained on Wikipedia abstract xml files, uses Naïve Bayes to detect language.
- Can be used to detect multiple languages upto 49 languages with 99% accuracy.

Limitations for Language Detection:

- Failed when lyrics were short, hence limited by text length.
- Not optimized to find single language, good for multi-language detection.
- Takes more time, as it searches for all possible languages.



`langid` Library

- `langid` is a standalone Language Identification (LangID) tool.
- It's trained on 5 diverse datasets (JRC-Acquis, ClueWeb 09, Wikipedia, Reuters RCV2, Debian i18n)
- Uses Naïve Bayes, and can detect 97 languages. Faster and lighter than many libraries.

Limitations for Language Detection:

- Failed when lyrics that were short, hence limited by text length.
- Not optimized to find single language, good for multi-language detection.

あ

β

Naive Bayes Classification

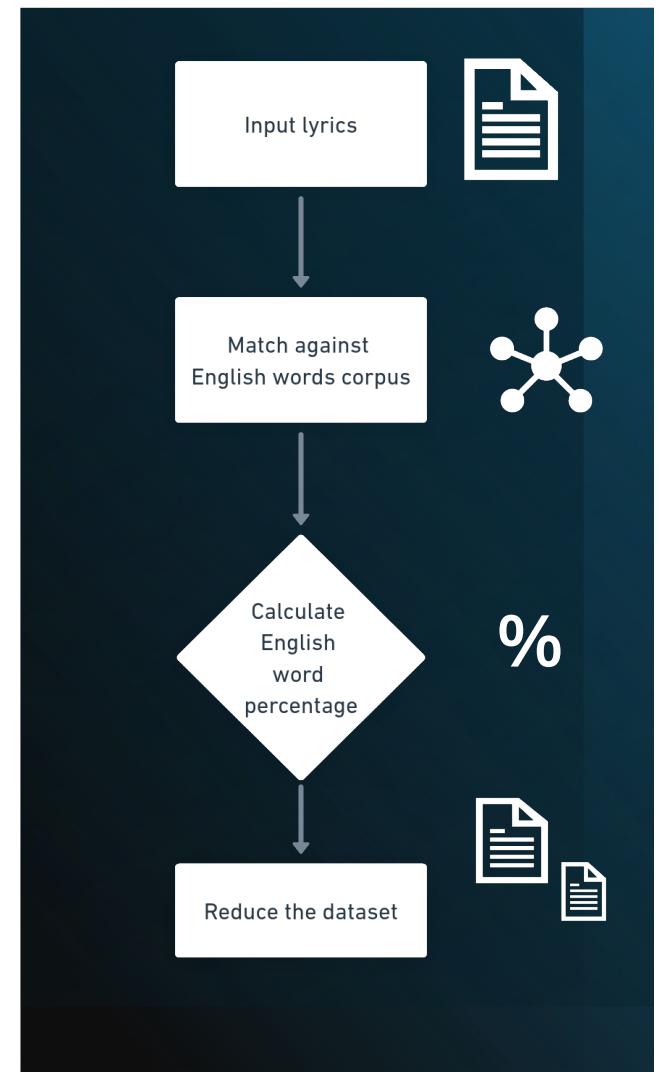
$$P(Y/X) = \frac{P(X/Y) * P(Y)}{P(X)}$$

NLTK Corpus – Similarity

- Corpus – similarity method is a simple language detection, that utilizes NLTK word corpus.
- It calculates the percentage of English words, and classifies if it's English or not.
- The best thing is it only concentrates on English language and provides faster and accurate detection of English language.
- It's faster and light on computation.
- Uses corpus like NLTK word that is not trained or based on specific domain data.
- Fast and simple to use for real-time applications.

Limitations for Language Detection:

- Not useful for multi-language classification.
- Only useful for English, unless a reliable corpus is available for other language.

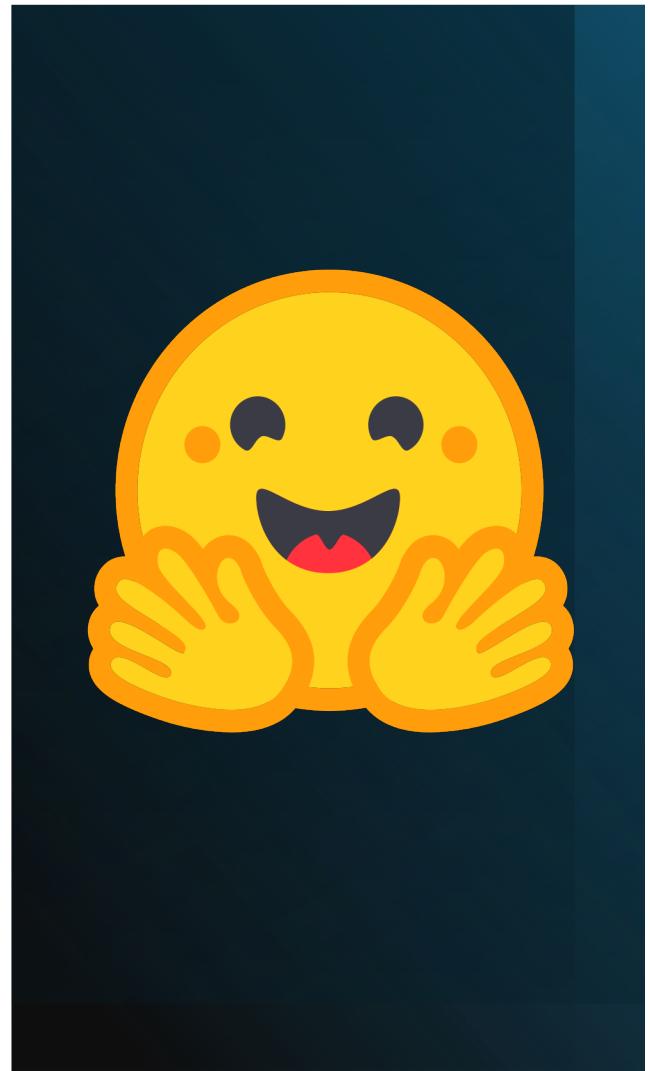


Xlm-RoBERTa – Hugging Face

- XLM-RoBERTa is a multilingual model trained on 100 different languages.
- It's a deep learning pre-trained model and optimized version of BERT for language detection.
- XLM-RoBERTa model pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages.
- It's a heavy, robust and complex model for language detection.

Limitations for Language Detection:

- Complex, heavy and takes too much of time. ~ 1 hr for 100k data.
- With lack of training data, it couldn't even predict simple sentences.

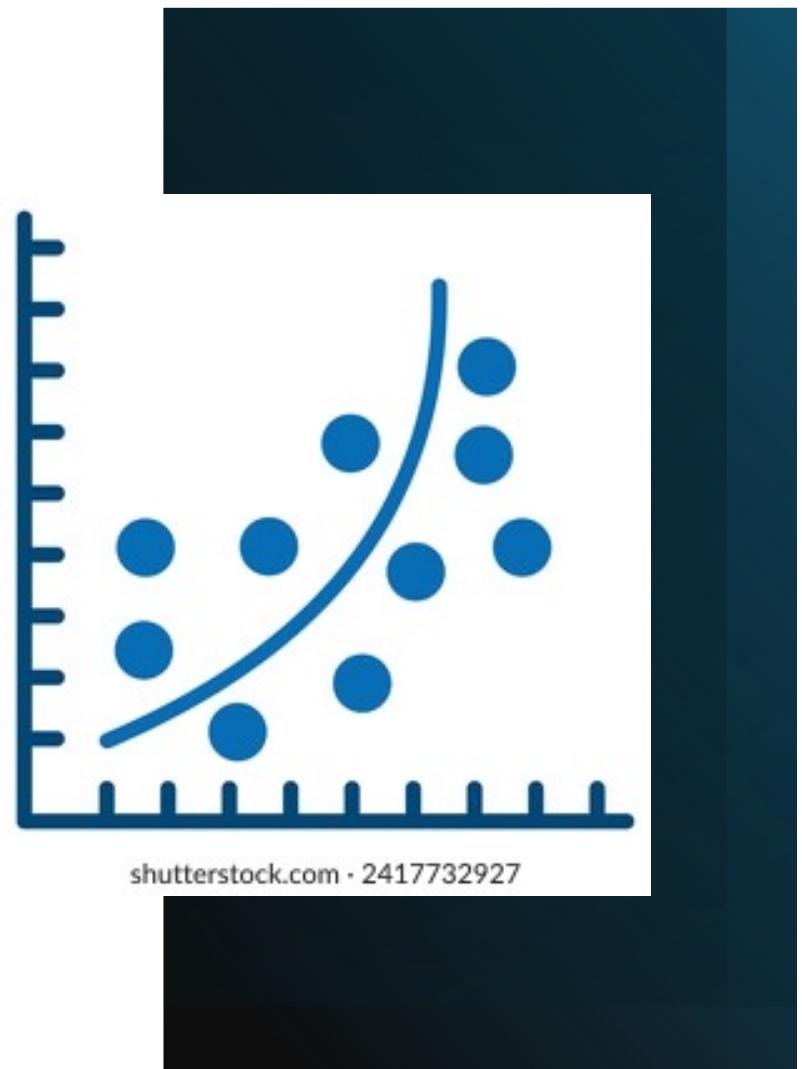


Logistic Regression

- Logistic Regression is the Statistical Method used of Classification. It models the relationship between one or more independent variables (features) and a dependent variable (target) by estimating probabilities.
- The Logistic Regression, Internally used the Sigmoid Function to map the output data.

Limitations for Emotion Detection:

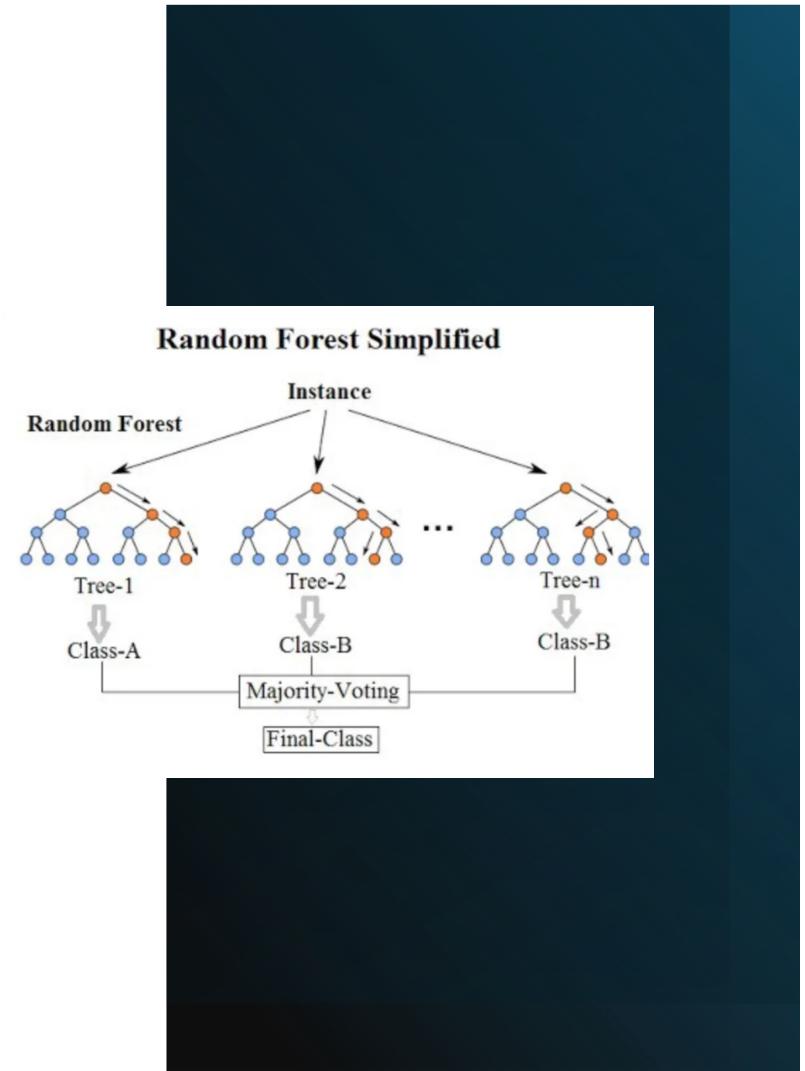
- As Emotions are often multi-class (e.g., happy, sad, angry, energetic), so need to extend logistic regression to **Multinomial Logistic Regression** to handle more than two classes.



shutterstock.com · 2417732927

Random Forest

- A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning.
- process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

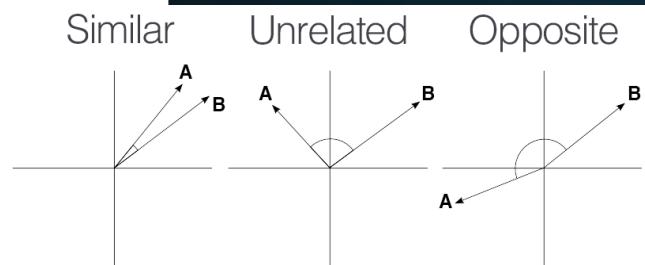


Cosine Similarity

- Cosine similarity is a measure used to determine how similar two vectors are by calculating the cosine of the angle between them. It's commonly used in text analysis, recommendation systems, and clustering to compare the similarity of documents, sentences, or other data points represented as vectors.

Limitations in Song Recommendations:

- When dealing with large datasets or high-dimensional embeddings (e.g., word embeddings of lyrics), cosine similarity may struggle to effectively differentiate between songs that are truly related vs. those with only surface-level similarities.

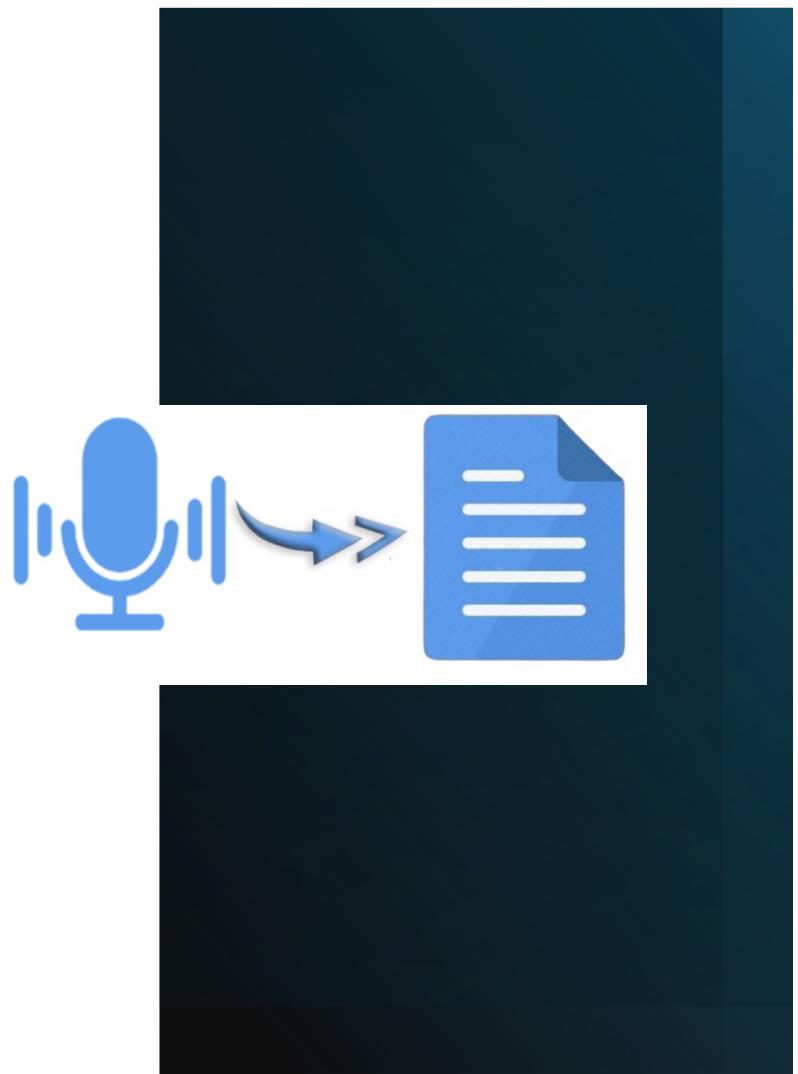


‘SpeechRecognition’ Library

- The speech_recognition library is a powerful Python module that enables easy-to-use speech-to-text conversion. It allows you to capture audio from various sources (such as microphones, audio files, or other input devices) and convert spoken words into text using different speech recognition engines.
- In your project, we used to convert songs into lyrics and highlight them, the speech_recognition library helps by changing the song's audio into text. Once the song is turned into text, it will then highlight certain words or parts of the lyrics and sync them with the music, making the lyrics appear at the right time as the song plays. This creates a fun and engaging experience for the user.

Limitations in highlighting lyrics:

- The speech_recognition library may struggle with accurately transcribing songs due to background music, unclear vocals, overlapping lyrics, and varying accents.



Dataset

Dataset Description

- The Spotify Songs dataset comprises approximately 1 million records, organized into two separate datasets.
- Dataset 1 includes song-related attributes such as danceability, energy, album name, and etc.
- Dataset 2 contains the full lyrics of each song with timestamps, using the song ID as a foreign key linking it to Dataset-1.

```
[ ] # Display the shape of the dataset
songs_dataframe_1_shape = songs_dataframe_1.shape
print(f"The shape of the dataset-1 is: {songs_dataframe_1_shape}")

songs_dataframe_2_shape = songs_dataframe_2.shape
print(f"The shape of the dataset-2 is: {songs_dataframe_2_shape}")

→ The shape of the dataset-1 is: (955320, 17)
The shape of the dataset-2 is: (36519092, 3)
```

	id	startTimeMs	words
0	000TJYIDLPM01ebX8QtIUS	42710	Siento escapar mi vida
1	000TJYIDLPM01ebX8QtIUS	70350	que triste se va, triste termina.
2	000TJYIDLPM01ebX8QtIUS	74780	Intento olvidar tu gran herida
3	000TJYIDLPM01ebX8QtIUS	87530	que siendo mortal, es también mía.
4	000TJYIDLPM01ebX8QtIUS	94990	Tal vez podré algún día surcar otra vez

```
+-----+
| Features |
+=====+
| id      |
+-----+
| name   |
+-----+
| album_name |
+-----+
| artists |
+-----+
| danceability |
+-----+
| energy   |
+-----+
| key     |
+-----+
| loudness |
+-----+
| mode    |
+-----+
| speechiness |
+-----+
| acousticness |
+-----+
| instrumentalness |
+-----+
| liveness |
+-----+
| valence  |
+-----+
| tempo   |
+-----+
| duration_ms |
+-----+
| lyrics  |
+-----+
```

Dataset Description

- We merged Dataset-1 and Dataset-2 based on the ID field and retrieved the top 100,000 records for our project.

id	name	album_name	artists	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	lyrics
0Prc5TDjAnEglqbxdY9	! UNDENIABLE	[HELLYEAH]		0.415	0.6050	7	-11.157	1	0.0575	0.00116	0.838	0.471	0.193	100.059	79500.0	He said he came from Jamaica.\nhe owned a couple houses A couple fake visas 'cause he never got his'n papers Gave up on love for me with them heart breakers But he was still getting money with the movers and the shakers. He was mixed with a'n couple things, bad boy with a couple rings Bricks in the condo and grammas'n to singin' on Jeff arm.'n baby mother tattooed 5-year bid up North when they ratified Anytime I feltin' him, helped him, put him on lock 'n' set 'em belt him Took him out to Belgium, we're home. Bitches the pretty.'n that's seldom This

```
[ ] # Display the shape of the dataset
songs_dataframe_shape = songs_dataframe.shape
print(f"The shape of the dataset is: {songs_dataframe_shape}")
```

→ The shape of the dataset is: (100000, 17)

Implementation

Loading the Data from Google Drive

>Loading the Dataset-1

```
[ ] # Use the file ID from your shareable link
file_id = '1k-ocQutBEDBokcX030aZbcXVpdCjKrt5'
url = f"https://drive.google.com/uc?id={file_id}"
output = 'songs_file.csv'
gdown.download(url, output, quiet=False)

# Step 4: Load the file into a DataFrame
songs_dataframe_1 = pd.read_csv(output)
```

→ Downloading...
From (original): <https://drive.google.com/uc?id=1k-ocQutBEDBokcX030aZbcXVpdCjKrt5>
From (redirected): <https://drive.google.com/uc?id=1k-ocQutBEDBokcX030aZbcXVpdCjKrt5&confirm=t&uuid=75bcff56-fb9e-4229-a975-b5578102f6d2>
To: /content/songs_file.csv
100%|██████████| 1.54G/1.54G [00:36<00:00, 42.0MB/s]

▶ Displaying the data in the tabular format
songs_dataframe_1[['lyrics']] = songs_dataframe_1[['lyrics']].apply(lambda x: x.splitlines()[0] if isinstance(x, str) and x.strip() else x)
songs_dataframe_1.head()

	id	name	album_name	artists	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness
0	0Prct5TDjAnEglqbxcldY9	!	UNDENIABLE	[HELLYEAH]	0.415	0.6050	7	-11.157	1	0.0575	0.00116	0.1
1	2ASl4wirkeYm3OWZxXKYuq	!!		NaN	Yxngxr1	0.788	0.6480	7	-9.135	0	0.3150	0.90000
2	69lcggVPmOr9cvPx9kLiiN	!!! - Interlude	Where I Belong EP	[Glowie]	0.000	0.0354	7	-20.151	0	0.0000	0.90800	0.1

Un Palo AI

Data Preprocessing



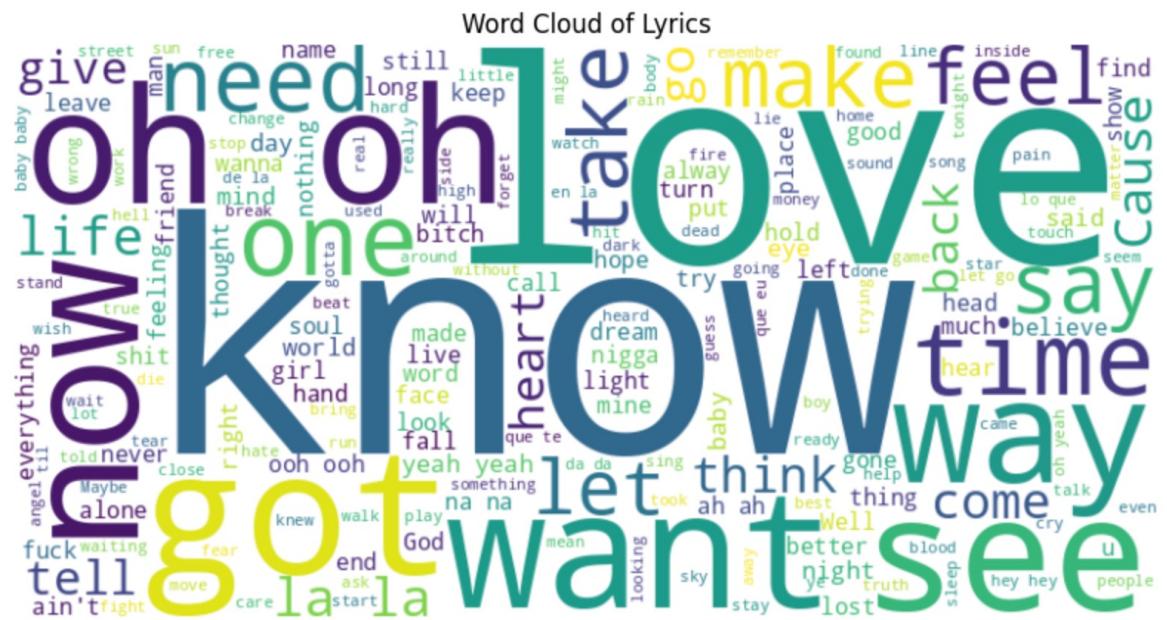
Handling Null
Values

Removing Stop
Words from Lyrics

Column	Total Records	Non-Null Count	Null Count	Dtype
id	100000	100000	0	object
name	100000	100000	0	object
album_name	100000	100000	0	object
artists	100000	100000	0	object
danceability	100000	100000	0	float64
energy	100000	100000	0	float64
key	100000	100000	0	object
loudness	100000	100000	0	float64
mode	100000	100000	0	object
speechiness	100000	100000	0	float64
acousticness	100000	100000	0	float64
instrumentalness	100000	100000	0	float64
liveness	100000	100000	0	float64
valence	100000	100000	0	float64
tempo	100000	100000	0	float64
duration_ms	100000	100000	0	float64
lyrics	100000	100000	0	object

	id	lyrics
0	0Prct5TDjAnEqlqbclY9	said came Jamaica, owned couple acres couple fake visas 'cause never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things, bald like couple rings Bricks condo grams sing-sing Left arm, baby mother tatted 5-year bid North ratted Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top toppa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, ain't it? (alright) Baby world, ain't it? Uh, got nigga home one side Best friend dyke, fucked around times momma alike, fight tell "make money", tell "make wife" tell bitch "you crazy, fuck wrong you?" excuse French, I'm long kisser try tell I'm one that's hitting say "what niggas?" say: "what niggas?" right, tonight? Put something tight, judge me, I'd get life love like brother, fuck like husband Pussy like oven, hot put tongue rub it, genie bottle Pussy wet, I'mma need goggles tell that's mine, tell "stop lying, mine else?" say worry Lil Tune holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, ain't it? Baby world, ain't it? know want boy, see tryin' keep pushing, I'mma let slide close eyes and, horizon ready, come get Yeah, yeah, yeah, yeah
1	2ASI4wirkeYm3OWZxXKYuq	Fucked bitch, running kids said never listen, yeah, yeah Yeah, yeah, yeah Tell things 'bout family Struggle words, said rambling Always playing games gambling got string I'm dancing Tell childhood TV shows Tell something could run feel lucky feel lucky, yeah I'm love Tell things you, baby know I'm love Tell things you, baby know I'm love Tell things you, baby know I'm love, oh-oh-oh Got tough heart 'cause rough start love cars love stars main dream thing see never fuck I'm seem, yeah Comin' home you, baby way flick hair makes crazy I've making much time lately never time lately I'm love Tell things you, baby know I'm love Tell things you, baby know I'm love Tell things you, baby know I'm love, oh-oh-oh I'm love, oh-oh-oh

Exploratory Data Analysis



Implementation libraries



Data Processing & Handling

pandas, numpy, joblib, Gdown



Text Processing & NLP

nltk, contractions, re, transformers
Sentiment Analysis:
GloVe, scikit-learn



Language Detection

langid, langdetect, transformers



Visualization

matplotlib, seaborn



Audio Processing & Speech Recognition

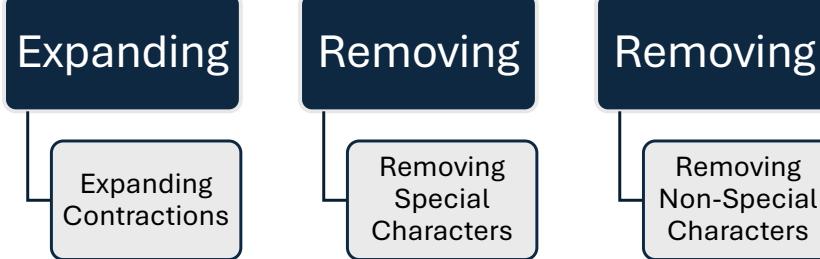
pydub, ffmpeg,
SpeechRecognition
Noise Reduction:
noisereduce



Integration & Timing

time, numpy, threading,
io, IPython.display

Language Detection Preprocessing



<code>id</code>	<code>lyrics</code>	<code>lyrics_expanded</code>	<code>lyrics_without_special_characters</code>	<code>non_english_char_count</code>
0Prct5TDjAnEglqbxclD9Y	<p>said came Jamaica, owned couple acres couple fake visas 'cause never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things, bald like couple rings Bricks condo grams sing-sing Left arm, baby mother tatted 5-year bid North ratted Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top tappa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, ain't it? (alright) Baby world, ain't it? Uh, got nigga home one side Best friend dyke, fucked around times momma alike, fight tell "make money", tell "make wife" tell bitch "you crazy, fuck wrong you?" excuse French, I'm long kisser try tell I'm one that's hitting say "what niggas?" say: "what niggas?" right, tonight? Put something light, judge me, I'd get life love like brother, fuck like husband Pussy like oven, hot put tongue rub it, genie bottle Pussy wet, I'mma need goggles tell that's mine, tell "stop lying, mine else?" say worry Lil' Tune holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, aren't they? Baby world, ain't it? know want boy, see tryin' keep pushing, I'mma let slide close eyes and, horizon ready, come get Yeah, yeah, yeah, yeah</p>	<p>said came Jamaica, owned couple acres couple fake visas because never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things bald like couple rings Bricks condo grams sing-sing Left arm, baby mother tatted 5-year bid North ratted Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top tappa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, you, are not high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, are not it? Uh, got nigga home one side Best friend dyke fucked around times momma alike fight tell make money tell make wife tell bitch you crazy fuck wrong you excuse French I am long kisser try tell I am one that is hitting say what niggas say what niggas right tonight Put something light judge me I would get life love like brother fuck like husband Pussy like oven hot put tongue rub it genie bottle Pussy wet Imma need goggles tell that is mine tell "stop lying, mine else?" say worry Lil' Tune holler me you are not high school crew slide Give whenever want whip whenever want Baby Anywhere, everywhere, baby world, are not it Baby world are not it know want boy see tryin' keep pushing Imma let slide close eyes and, horizon ready, come get Yeah yeah yeah yeah</p>	<p>said came Jamaica owned couple acres couple fake visas because never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things bald like couple rings Bricks condo grams sing-sing Left arm baby mother tatted 5-year bid North ratted Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I am Momma Dee order, call Daddy like daughters like get drunk, like sober That is top tappa never fuck beginners let play pussy lick fingers I am zone holler me you are not high school crew slide Give whenever want whip whenever want Baby Anywhere everywhere baby world are not it alright Baby world are not it Uh got nigga home one side Best friend dyke fucked around times momma alike fight tell make money tell make wife tell bitch you crazy fuck wrong you excuse French I am long kisser try tell I am one that is hitting say what niggas say what niggas right tonight Put something light judge me I would get life love like brother fuck like husband Pussy like oven hot put tongue rub it genie bottle Pussy wet Imma need goggles tell that is mine tell stop lying mine else say worry Lil' Tune holler me you are not high school crew slide Give whenever want whip whenever want Baby Anywhere everywhere baby world are not it Baby world are not it know want boy see tryin' keep pushing Imma let slide close eyes and, horizon ready come get Yeah yeah yeah yeah</p>	0

Language Detection



Language Detection -
Method-1: Using
'langdetect' Library



Language Detection -
Method-2: Using 'langid'



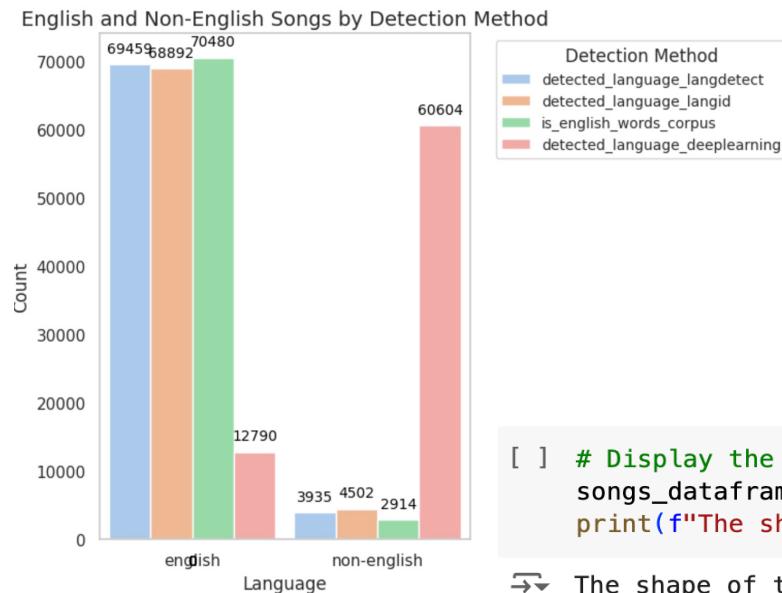
Language Detection -
Method-3: NLTK Corpus
– Similarity



Language Detection -
Method-4: Hugging face
– Xlm-roberta

Language Detection...

Best Model



USING 'LANGDETECT' LIBRARY	USING 'LANGID'	NLTK CORPUS – SIMILARITY	HUGGING FACE – XLM- ROBERTA
Run Time : 5min	Run Time: 4 min	Run Time: <1 min	Run Time: >1hour

```
[ ] # Display the shape of the dataset
songs_dataframe_cleaned_shape = songs_dataframe_cleaned.shape
print(f"The shape of the cleaned dataset is: {songs_dataframe_cleaned_shape}")
```

→ The shape of the cleaned dataset is: (73394, 19)

NLTK – Corpus Similarity

```
[ ] # Ensure the corpus is downloaded
nltk_15.download('words')

[?] [nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]  Package words is already up-to-date!
True

# Load the list of English words from the NLTK words corpus
english_words_set = set(words.words())

# Function to check the percentage of words in English
def english_word_percentage(text):
    try:
        if isinstance(text, str) and text.strip():
            # Split the text into words
            words_in_text = text.split()
            # Count words that exist in the English words set
            english_word_count = sum(1 for word in words_in_text if word.lower() in english_words_set)
            # Calculate the percentage of English words
            percentage = english_word_count / len(words_in_text)
            return percentage
    else:
        return 0 # Return 0 if the text is empty or not a string
    except Exception as e:
        print(f'Error processing text: {e}')
        return 0

# Apply the function to calculate the percentage of English words
language_detection_songs_dataframe['english_word_percentage'] = language_detection_songs_dataframe['lyrics_with']

# Create a column to indicate if the percentage meets a threshold (e.g., 50%)
threshold = 0.5
language_detection_songs_dataframe['is_english_words_corpus'] = language_detection_songs_dataframe['english_word_percentag
+ Code + Text
```

Selecting a Corpus.

Developing the method.

Threshold & Pre-processing.

Sentiment Analysis

Emotion Detection



Tokenization : The process of breaking down text into smaller units, such as words, sentences, or subwords, to facilitate analysis.



Lemmatization: Reducing a word to its base or root form, considering its meaning and context.



Stemming: Reducing a word to its root form by chopping off prefixes or suffixes, without considering its meaning.



Vectorization: The process of converting text into numerical representations (vectors) to enable machine learning models to process and analyze the text.

Tokenization

```
❶ # Function to tokenize the lyrics
def tokenize_lyrics(lyrics):
    if isinstance(lyrics, str): # Check if the lyrics is a string
        return word_tokenize(lyrics)
    else:
        return [] # Return an empty list if lyrics are not a string

# Ensure that NaN or None values are handled by converting them to empty strings
songs_dataframe_cleaned['lyrics_without_special_characters'] = songs_dataframe_cleaned['lyrics_without_special_characters'].fillna('')

# Create the new DataFrame with id, lyrics, and tokenized lyrics
tokenized_songs_dataframe = songs_dataframe_cleaned[['id', 'lyrics', 'lyrics_without_special_characters']].copy()

# Apply tokenization
tokenized_songs_dataframe['tokenized_lyrics'] = tokenized_songs_dataframe['lyrics_without_special_characters'].apply(tokenize_lyrics)
```

	<code>id</code>	<code>lyrics</code>	<code>lyrics_without_special_characters</code>	<code>tokenized_lyrics</code>
0	0PrctSTDjAnEqlqbxdY9	said came Jamaica, owned couple acres couple fake visas 'cause never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things, bald like couple rings Bricks condo grams sing- sing Left arm, baby mother tattooed 5-year bid North rated Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top toppa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, you, ain't high school crew, slide Give whenever want whip whenever want Baby Anywhere, everywhere, baby world, ain't it? (adult) Baby world, now it? Uh, got nigga home one side Best friend dyke, fucked around times momma alike, fight tell "make money", tell "make wife", tell bitch "you crazy, fuck wrong you?" excuse French, I'm long kisser try tell I'm one that's hitting say "what niggas?" say: "what niggas?" right, tonight? Put something tight, judge me, I'd get life love like brother, fuck like husband Pussy like oven, hot put tongue rub it, genie bottle Pussy wet, I'mma need goggles tell that's mine, tell "stop lying, mine else?" say worry Lil Tune holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, ain't it? Baby world, ain't it? know want boy, see tryin' keep pushing, I'mma let slide close eyes and, horizon ready, come get Yeah, yeah, yeah, yeah	said came Jamaica owned couple acres couple fake visas because never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things bald like couple rings Bricks condo grams singsing Left arm baby mother tattooed 5year bid North rated Anyway felt him helped him put lock seatbelt Took Belgium welcome Bitches pretty that's seldom box better box held I am Momma Dee order call Daddy like daughters like get drunk like sober That is topoppa never fuck beginners let play pussy lick fingers I am zone holler me you you are not high school crew slide Give whenever want whip whenever want Baby Anywhere everywhere baby world are not it Baby world are not it Uh got nigga home one side Best friend dyke fucked around times momma alike fight tell make money tell make wife tell bitch you crazy fuck wrong you excuse French I am long kisser try tell I am one that is hitting say what niggas say what niggas right tonight Put something tight judge me I would get life love like brother fuck like husband Pussy like oven hot put tongue rub it genie bottle Pussy wet Imma need goggles tell that is mine tell stop lying mine else say worry Lil Tune holler me you you are not high school crew slide Give whenever want whip whenever want Baby Anywhere everywhere baby world are not it Baby world are not it know want boy see tryin' keep pushing Imma let slide close eyes and horizon ready come get Yeah yeah yeah yeah	[said, came, Jamaica, owned, couple, acres, couple, fake, visas, because, never, got, papers, Gave, love, fucking, heart, breakers, getting, money, movers, shakers, mixed, couple, things, bald, like, couple, rings, Bricks, condo, grams, singsing, Left, arm, baby, mother, tattooed, 5year, bid, North, rated, Anyway, feel, him, helped, him, put, lock, seatbelt, Took, Belgium, welcome, Bitches, pretty, that, is, seldom, box, better, box, held, I, am, Momma, Dee, order, call, Daddy, like, daughters, like, get, drunk, like, sober, That, is, top, toppa, never, fuck, beginners, let, play, pussy, lick, fingers, I, am, zone, holler, me, you, you, are, not, high, school, crew, ...]

Lemmatization

```
# Initialize WordNetLemmatizer
lemmatizer = WordNetLemmatizer()

# Function to lemmatize the tokens
def lemmatize_tokens(tokens):
    return [lemmatizer.lemmatize(token) for token in tokens]

# Create the new DataFrame with id, lyrics, tokenized_lyrics, and lemmatized_lyrics
lemmatized_songs_dataframe = tokenized_songs_dataframe[['id', 'lyrics', 'tokenized_lyrics']].copy()

# Apply lemmatization
lemmatized_songs_dataframe['lemmatized_lyrics'] = lemmatized_songs_dataframe['tokenized_lyrics'].apply(lemmatize_tokens)

# Display the DataFrame in tabular format after lemmatization
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
pd.set_option('display.max_colwidth', None)
display(lemmatized_songs_dataframe.head(3))
```

	id	lyrics	tokenized_lyrics	lemmatized_lyrics
0	0Prct5TDjAnEglqbxcldY9	said came Jamaica, owned couple acres couple fake visas 'cause never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things, bald like couple rings Bricks condo grams sing-sing Left arm, baby mother tatted 5-year bid North ratted Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top toppa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, ain't it? (alright) Baby world, ain't it? Uh, got nigga home one side Best friend dyke, fucked around times momma alike, fight tell "make money", tell "make wife" tell bitch "you crazy, fuck wrong you?" excuse French, I'm long kisser try tell I'm one that's hitting say "what niggas?" say: "what niggas?" right, tonight? Put	[said, came, Jamaica, owned, couple, acres, couple, fake, visas, because, never, got, papers, Gave, love, fucking, heart, breakers, getting, money, movers, shakers, mixed, couple, things, bald, like, couple, rings, Bricks, condo, grams, sing-sing, Left, arm, baby, mother, tatted, 5year, bid, North, ratted, Anyway, felt, him, helped, him, put, lock, seatbelt, Took, Belgium, welcome, Bitches, pretty, that, is, seldom, box, better, box, held, I, am, Momma, Dee, order, call, Daddy, like,	[said, came, Jamaica, owned, couple, acre, couple, fake, visa, because, never, got, paper, Gave, love, fucking, heart, breaker, getting, money, mover, shaker, mixed, couple, thing, bald, like, couple, ring, Bricks, condo, gram, singsing, Left, arm, baby, mother, tattooed, 5year, bid, North, ratted, Anyway, felt, him, helped, him, put, lock, seatbelt, Took, Belgium, welcome, Bitches, pretty, that, is, seldom, box, better, box, held, I, am, Momma, Dee, order, call, Daddy, like, daughter,

Stemming (Porter Stemmer)

```
[ ] # Initialize the PorterStemmer
stemmer = PorterStemmer()

# Function to stem the lemmatized tokens
def stem_tokens(tokens):
    return [stemmer.stem(token) for token in tokens]

# Create the new DataFrame with id, lyrics, tokenized_lyrics, lemmatized_lyrics, and stemmed_lyrics
stemmed_songs_dataframe = lemmatized_songs_dataframe[['id', 'lyrics', 'tokenized_lyrics', 'lemmatized_lyrics']].copy()

# Apply stemming
stemmed_songs_dataframe['stemmed_lyrics'] = stemmed_songs_dataframe['lemmatized_lyrics'].apply(stem_tokens)

# Display the DataFrame in tabular format after Stemming
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
pd.set_option('display.max_colwidth', None)
display(stemmed_songs_dataframe.head(3))
```

		id	lyrics	tokenized_lyrics	lemmatized_lyrics	stemmed_lyrics
0	0Prct5TDjAnEglqbxcldY9		said came Jamaica, owned couple acres couple fake visas 'cause never got paper. Gave love fucking heart breakers getting money movers shakers mixed couple things, bald like couple rings Bricks condo grams sing- sing Left arm, baby mother tatted 5-year bid North rated Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top toppa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, you, ain't high school crew, slide Give whenever want, whip whenever want Baby Anywhere, everywhere, baby world, ain't it? (alright) Baby world, ain't it? Uh, got nigga home one side Best friend dyke, fucked around times momma alike, fight tell "make money", tell "make wife" tell bitch "you crazy, fuck wrong you?" excuse French, I'm long kisser try tell I'm one that's hitting say	[said, came, Jamaica, owned, couple, acres, couple, fake, visas, because, never, got, papers, Gave, love, fucking, heart, breakers, getting, money, movers, shakers, mixed, couple, things, bald, like, couple, rings, Bricks, condo, grams, singsing, Left, arm, baby, mother, tatted, 5year, bid, North, rated, Anyway, felt, him, helped, him, put, lock, seatbelt, Took, Belgium, welcome, Bitches, pretty, that, is, seldom, box, better, box, held, I, am. Momma. Dee. order. call. Daddy.	[said, came, jamaica, own, coupl, acr, coupl, fake, visa, becaus, never, got, paper, gave, love, fuck, heart, breaker, get, money, mover, shaker, mix, coupl, thing, bald, like, coupl, ring, brick, condo, gram, sings, left, arm, babi, mother, tat, 5year, bid, north, rat, anyway, felt, him, help, him, put, lock, seatbelt, took, belgium, welcom, bitch, pretti, that, is, seldom, box, better, box, held, i, am. momma. dee. order. call. daddi.	

Vectorization (GloVe)

		id	lyrics	tokenized_lyrics	lemmatized_lyrics	stemmed_lyrics	glove_vector	glove_vector_numeric
0	0Prct5TDjAnEqlqbxclY9		<p>said came Jamaica, owned couple acres couple fake visas 'cause never got papers Gave love fucking heart breakers getting money movers shakers mixed couple things, bald like couple rings Bricks condo grams sing-sing Left arm, baby mother tattooed 5-year bid North rated Anyway felt him, helped him, put lock, seat-belt Took Belgium, welcome Bitches pretty, that's seldom box better box held I'm Momma Dee order, call Daddy like daughters like get drunk, like sober That's top toppa, never fuck beginners let play pussy lick fingers, I'm zone holler me, you, ain't high school crew slide Give whenever want, whip whenever want Baby anywhere, everywhere, baby world, ain't it? (alright) Baby world, ain't it? Uh, got nigga home one side Best friend dyke, fucked around times momma alike, fight tell "make money", tell "make wife" tell bitch "you crazy, fuck wrong you?" excuse French, I'm long kisser try tell I'm one that's hitting say "what niggas?" say: "what niggas?" right, tonight? Put something tight, judge me, I'd get life love like brother, fuck like husband Pussy like oven, hot put tongue rub it, genie bottle Pussy wet, I'mma need someone tell that's mine, tell</p>	[said, came, Jamaica, owned, couple, acres, couple, fake, visas, because, never, got, papers, Gave, love, fucking, heart, breakers, getting, money, movers, shakers, mixed, couple, things, bald, like, couple, rings, Bricks, condo, grams, singsing, Left, arm, baby, mother, tattooed, 5year, bid, North, rated, Anyway, felt, him, helped, him, put, lock, seatbelt, Took, Belgium, welcome, Bitches, pretty, that, is, seldom, box, better, box, held, I, am, Momma, Dee, order, call, Daddy, like, daughters, like, get, drunk, like, sober, That, is, top, toppa, never, fuck, beginner, beginners, let, play, pussy, lick, fingers, I, am, zone, holler, me, you, you, are, not, high, school, crew, ...]	[said, came, jamaica, own, coupl, acr, coupl, fake, visa, becaus, never, got, paper, Gave, love, fuck, heart, breaker, get, money, mover, shaker, mix, coupl, thing, bald, like, coupl, ring, brick, condo, gram, sings, left, arm, babi, mother, tat, 5year, bid, north, rat, anyway, felt, him, help, him, put, lock, seatbelt, took, belgium, welcom, bitch, pretti, that, is, seldom, box, better, box, held, I, am, Momma, Dee, order, call, Daddy, like, daughter, like, get, drunk, like, sober, That, is, top, toppa, never, fuck, beginn, let, play, pussy, lick, finger, I, am, zone, holler, me, you, you, are, not, high, school, crew, ...]	[0.08974653, 0.10316204, 0.09011229, -0.28711596, 0.30701685, -0.07443138, -0.28884256, 0.14675121, -0.1762103, 0.15497473, -0.11771619, 0.2690226, -0.26836237, 0.0510136, 0.49038538, 0.28612027, 0.28377014, 0.04239313, -0.051632795, -0.35785016, -0.17742904, 0.27705163, 0.3541656, 0.21059477, 0.39794052, -1.3599495, -0.55661505, 0.2109708, 0.4130193, 0.27924544, -0.29406372, 0.08967351, -0.0951646, -0.059202153, 0.1536655, 0.0027032122, 0.2598489, -0.23379707, -0.11190175, 0.049095]		

Emotion of Song



Get the Emotion of the Song

```
def categorize_emotion(row):
    energy = row['energy']
    valence = row['valence']
    liveliness = row['liveness']
    danceability = row['danceability']

    # Define logic for categorizing based on attribute thresholds
    if valence > 0.6 and energy > 0.6 and danceability > 0.6:
        return 'Happy'
    elif valence < 0.4 and energy < 0.4 and danceability < 0.4:
        return 'Sad'
    elif energy > 0.7 and valence < 0.4 and liveliness > 0.5:
        return 'Angry'
    elif energy < 0.5 and valence > 0.5 and danceability > 0.4:
        return 'Relaxed'
    elif energy > 0.7 and danceability > 0.7:
        return 'Energetic'
    else:
        return 'Neutral' # Use this for songs that don't fit clear categories

songs_dataframe_cleaned = songs_dataframe_cleaned.copy()

# Apply the function to the DataFrame
songs_dataframe_cleaned['song_emotion'] = songs_dataframe_cleaned.apply(categorize_emotion, axis=1)

# Display the DataFrame with the new emotion categories
print(songs_dataframe_cleaned[['id', 'song_emotion']].head())
```

```
id song_emotion
0 0Prct5TDjAnEgIqbxxldy9 Neutral
1 2ASl4wirkeYm30WZxXKYuq Neutral
2 691cgvPm0r9cvPv9xLiiN Sad
3 4U7dIZg1s9pjdpqZy0fm Happy
4 4v1IBp3Y3rpkWmWzIkYju Happy
```

```
[ ] # Get the count of each emotion in the 'song_emotion' column
emotion_counts = final_songs_dataframe['song_emotion'].value_counts()

# Display the count of each category
print(emotion_counts)
```

```
song_emotion
Neutral      52084
Happy        8964
Relaxed      5400
Sad          3630
Angry         1681
Energetic     1635
Name: count, dtype: int64
```



Training Machine Learning Models

Logistic Regression

```
# Initialize the Logistic Regression model
logreg = LogisticRegression(max_iter=1000)

# Train the model
logreg.fit(X_train_scaled, y_train)

# Make predictions on the test set
y_pred = logreg.predict(X_test_scaled)

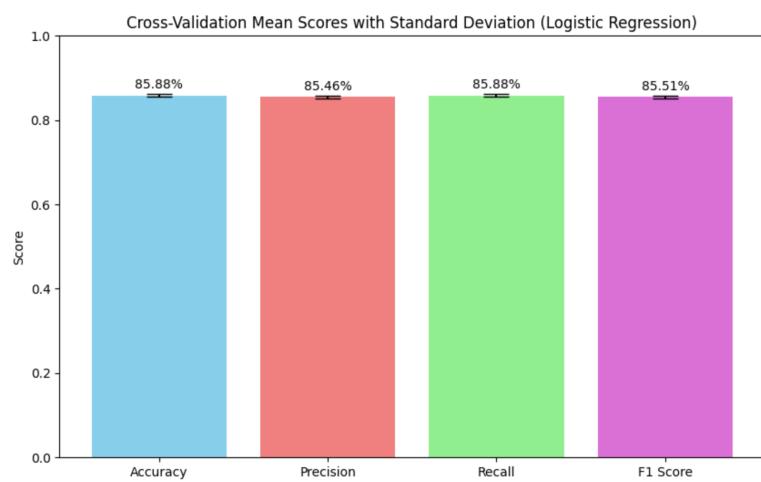
# Evaluate the model performance
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted') # Use 'weighted' for multi-class
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

metrics = {
    "accuracy": accuracy,
    "precision": precision,
    "recall": recall,
    "f1-score": f1
}
# Update the performance_metrics dictionary using a map
performance_metrics["logistic_regression"].update(metrics)
```

```
[ ] # Print the metrics
print("Logistic Regression Performance Metrics:")
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")
```

```
↳ Logistic Regression Performance Metrics:
Accuracy: 0.8601
Precision: 0.8563
Recall: 0.8601
F1-Score: 0.8572
```

Logistic Regression...



Random Forest

```
❶ # Initialize the Random Forest model
rf = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
rf.fit(X_train_scaled, y_train)

# Make predictions on the test set
y_pred_rf = rf.predict(X_test_scaled)

# Evaluate the model performance
accuracy_rf = accuracy_score(y_test, y_pred_rf)
precision_rf = precision_score(y_test, y_pred_rf, average='weighted') # Use 'weighted' for multi-class
recall_rf = recall_score(y_test, y_pred_rf, average='weighted')
f1_rf = f1_score(y_test, y_pred_rf, average='weighted')

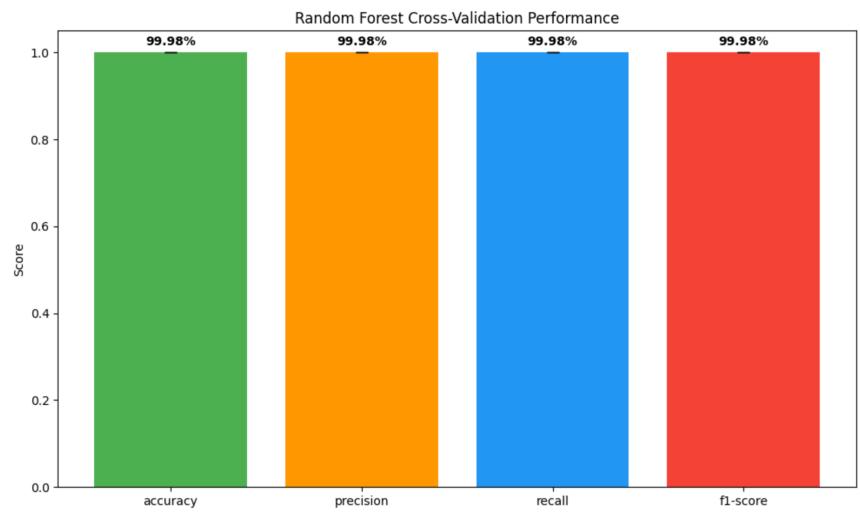
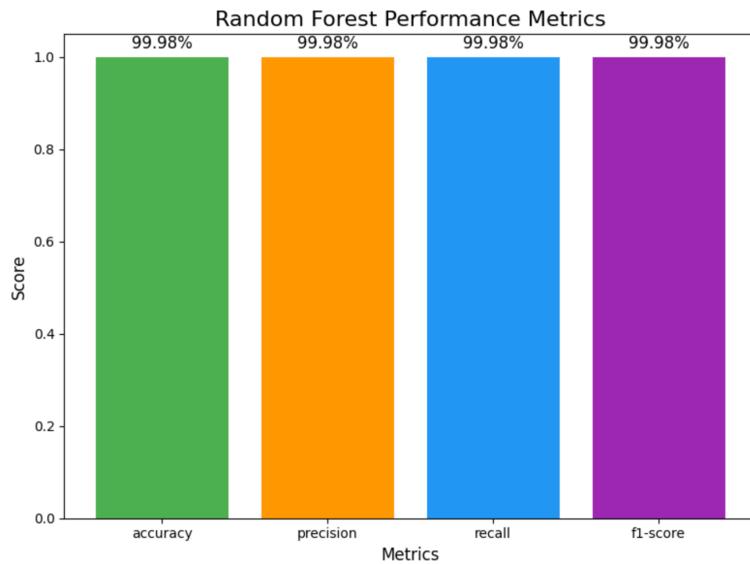
# Store the metrics for Random Forest
metrics_rf = {
    "accuracy": accuracy_rf,
    "precision": precision_rf,
    "recall": recall_rf,
    "f1-score": f1_rf
}

# Update the performance_metrics dictionary using a map
performance_metrics["random_forest"].update(metrics_rf)

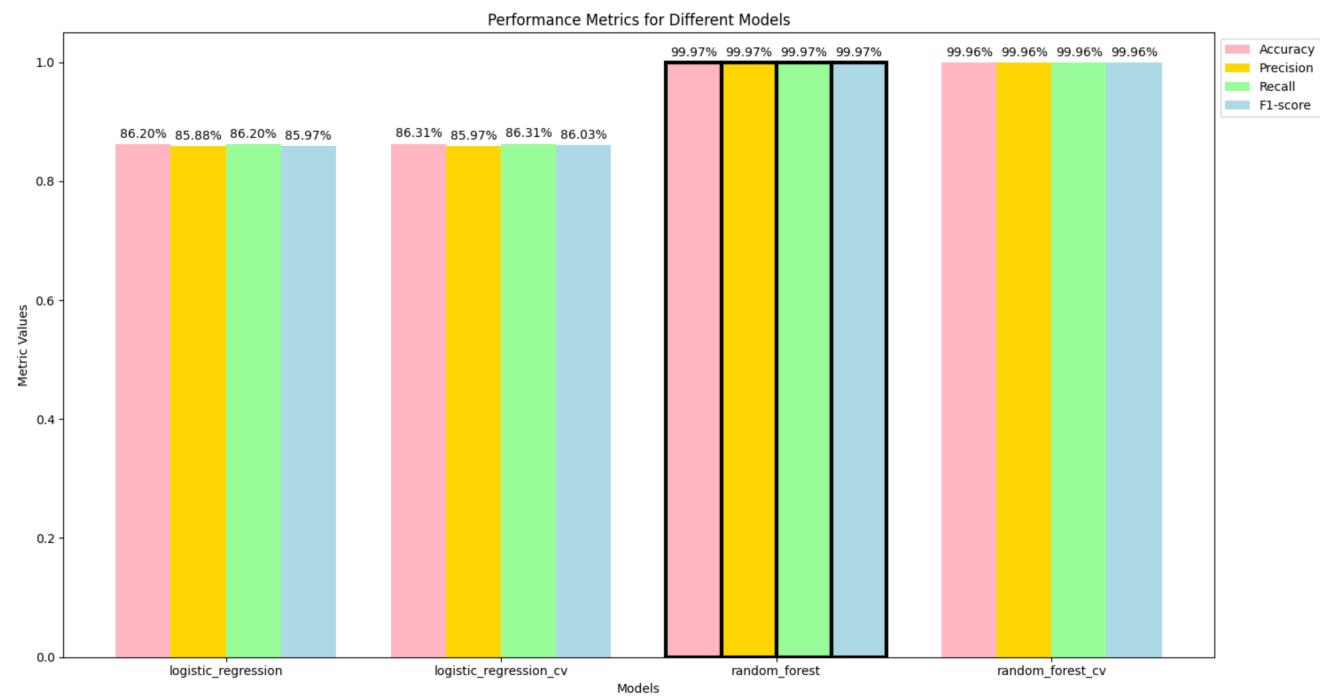
# Print out the performance metrics for Random Forest
print(f"Random Forest Accuracy: {accuracy_rf:.4f}")
print(f"Random Forest Precision: {precision_rf:.4f}")
print(f"Random Forest Recall: {recall_rf:.4f}")
print(f"Random Forest F1-Score: {f1_rf:.4f}")

❷ Random Forest Accuracy: 0.9998
Random Forest Precision: 0.9998
Random Forest Recall: 0.9998
Random Forest F1-Score: 0.9998
```

Random Forest...



Comparison of Results



Recommendation System

Reading/Validate the Input Data

```
▶ # Function to read valid input for each feature and check for null values
def get_valid_input(attribute_name):
    while True:
        # Read input for each attribute dynamically
        input_value = input(f"Please enter the value for {attribute_name}: ")

        # Check if input is not empty or null
        if input_value.strip():
            return input_value
        else:
            print(f"Input for {attribute_name} is invalid. Please enter a valid value.")

    # Define all the features
features = ['danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'lyrics']

    # Dictionary to store the valid inputs
feature_inputs = {}

    # Loop through each feature and get the valid input
for feature in features:
    feature_inputs[feature] = get_valid_input(feature)

    # Output the valid inputs
for feature, value in feature_inputs.items():
    print(f"Valid Input for {feature}: {value}")


```

☞ Please enter the value for danceability: 0.7
Please enter the value for energy: 0.7
Please enter the value for key: 7
Please enter the value for loudness: 11
Please enter the value for mode: 2
Please enter the value for speechiness: 0.05
Please enter the value for acousticness: 0.0001
Please enter the value for instrumentalness: 0.7
Please enter the value for liveness: 0.4
Please enter the value for valence: 0.4
Please enter the value for tempo: 100
Please enter the value for duration_ms: 53764
Please enter the value for lyrics: I woke up to the sound of the rain, Falling softly on my windowpane. A gentle breeze, the world feels new, I can't help but think

Recommended Songs

```
# Filter songs by emotion
filtered_songs = final_songs_dataframe[final_songs_dataframe['emotion_category'] == emotion_mapping[y_pred]]

# Function to convert the string representation of the glove_vector to a numpy array
def convert_glove_vector(glove_str):
    glove_str_cleaned = glove_str.replace('[', '').replace(']', '').replace('\n', ' ').strip() # Remove newlines, brackets, and extra spaces
    glove_vector = np.array(list(map(float, glove_str_cleaned.split()))) # Convert string to list of floats
    return glove_vector

# Convert the 'glove_vector' column strings to numpy arrays using .loc
filtered_songs.loc[:, 'glove_vector'] = filtered_songs['glove_vector'].apply(convert_glove_vector)

# Apply cosine similarity between the feature_inputs['glove_vector'] and each song's glove_vector
filtered_songs['similarity'] = filtered_songs['glove_vector'].apply(
    lambda x: cosine_similarity(feature_inputs['glove_vector'].reshape(1, -1), x.reshape(1, -1))[0][0]
)

<ipython-input-110-cccc41a0557d>:26: 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-versus-a-copy
filtered_songs['similarity'] = filtered_songs['glove_vector'].apply()

# Return top 5 songs
recommended_songs = filtered_songs.sort_values(by='similarity', ascending=False).head(5)

# Select only the 'lyrics' and 'similarity' columns
recommended_songs_output = recommended_songs[['lyrics', 'similarity']]

# Set display options for a clean and neat table output
pd.set_option('display.max_columns', None) # Display all columns
pd.set_option('display.width', 1000) # Set display width to fit more columns
pd.set_option('display.max_colwidth', None) # Prevent truncation of long text in columns

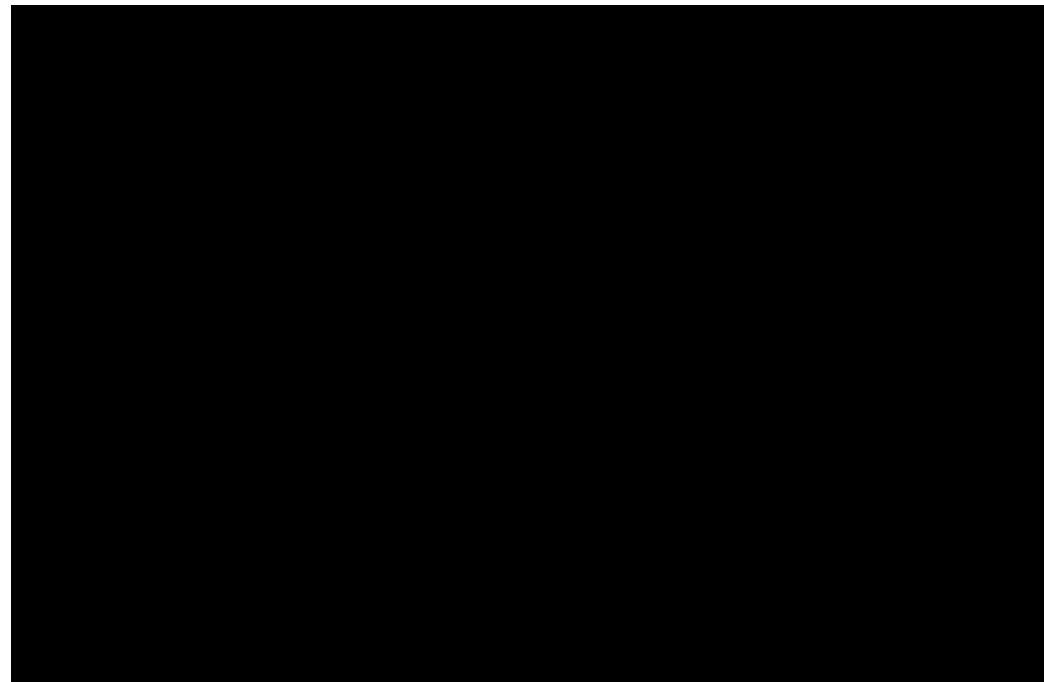
# Display the table (neatly formatted)
display(recommended_songs_output)
```

Recommended Songs...

		lyrics similarity
7354	wake half time, know five minutes think "What country? city? hotel in?" five minutes think "Is wild animal scratching around next bed? large rat raccoon squirrel broke in?" stumble around room, touching walls, feeling around Feeling around dark feel feel find light switch turn recognize cat there's beautiful woman laying bed you, head clear Wait I'm wrong, thought clear I'm alone There's guitar recording gear Okay, realize I'm hotel room San Francisco stayed girl tour San Francisco long That's really remember, hotel night long engineer's limping around messed knee mosh pit night He's blaming let Come watch McGregor fight apartment "I already girlfriend over," told "It's last night together Marc Hopkins sessions" carpet gold, walls yellow, curtains pink carpet gold, walls go bed 7 o'clock morning, wake 4 PM dark New Orleans 5:30 walk across Coliseum Square I'll get iced tea Mojo Magazine sit bit come back window, see Pontchartrain Hotel sign lit like always is, neon green finished book John Connolly, got choked end Like end book, Book Lost Things end Stan Laurel hears Pacific Ocean waves crashing mind, takes past Oliver Hardy audience clapping I'm listening Nels Cline's Lovers I'm watching news mudslides snowstorms east coast bother like everybody fingerpointing Saying somebody something wrong like news anymore pick remote click I'm falling asleep, I'm dream state I'm back San Francisco, walking around apartment earthquake I'm feeling? guitars falling place floors rising ceiling getting closer ceiling, heart beating faster faster run kitchen, grab food bottled water exit apartment fast look fruit bowl there's unbearably loud hum Coming cable car tracks oranges yellow, lemons green, cable car goes vroom-vroom dream, walked SPCA somebody asked "Mark, pet?" said, "Yes, do" says, "Well, kind pets?" said, "Well, take guess" guesses guesses said "Well wrong, unique pets" says, "How different?" said, "Well, let show video them" said, "Okay, let watch video them" show video pets' eyes bulged frightened Like Mia Farrow's eyes Rosemary's Baby fucked Satan kitty cat barks, duck goes quack, puppy dog goes meow kitty cat barks, duck goes quack, puppy dog goes meow kitty cat barks, duck goes quack, puppy dog goes meow friend said, "Mark, know leaving tonight child look after?" said, "Yeah, he's smart kid wants grow actor movie director Steve Railsback favorite actor world, watch movies together son's favorite movies Devil's Rejects, Ed Gein, Helter Skelter" friend said "Well, who's babysitter?" said, "Don't worry it, that's business, he's neighbor" said, "Look, gotta go, called taxi, son's gonna fine Talk later, alligator" got babysitter coming, live 666 Post, neighbor is... zoom got babysitter coming, live 666 Post, neighbor is... zoom got babysitter coming, live 666 Post, neighbor is... zoom got babysitter coming, live 666 Post, neighbor is... zoom	0.986262
31781	Shit got heavy like eyelids highway end got quiet like island left headed sea toothless shark actor playing part can't pretend care like unknown second home we're upside down, love Gravity change heart fell hell fell I've incisive intuition curse anything Always first know everything ends Bullets June nice night knife fight two boats drifting sun shines you, love well-lit like work art I've exiled like enemy Forever dark	0.982193
64402	Damm go walking find horses man-made lake trees Came back room covered sweat Swiss Waldhaus Hotel Filled application work visa Japan Australia weeks since I've left home feel place element work 7 night 5 AD says "Wrap" runner named Fabio flashlights back hotel sun comes get bed talk girl phone birds chirping hell end playing Italian film Set ski town Switzerland? Damm go walk yard, alone [?] felt like Jimmy Page walking mountains behind Aleister Crowley's house dark, got cold turned back around Came back room, read Graham Nash's Wild Tales Til fell asleep sound sound birds Films Yeah I've asked around nobody knows names 'em birds birds Films Yeah I've asked around nobody knows names 'em Damn go walking road girl named Veronica stopped said Milan recognized film today birthday talked little bit barrier went one way went walked along dandelions market bought flowers way back hotel left lobby hers, note "Veronica, happy birthday - Mark" saw set said "Grazie", could tell gesture touched heart Damm go dinner last night Paul throat sore could see feeling ill spends time set cold last two days, playing Hitler could see grappling felt bad, gave words support talked John Hughes movies, home ownership cost living San Francisco New York damn go later set dresser something like Name said "Brianna" talked four hours bar street music terrible yeah, liked her, kinda someone four five years kinda figured anyhow, told "Well, I" made life easier we walked drunk ass back room like gentleman, try went room looked waterfront balcony felt surrealness surroundings got bed Looked baby blue ceilings thought home girl ached love Damm ended fly New Orleans saw kitty cats sleeping porches drank real iced tea first time six eight weeks nice walk awkward path yell [?] eating pasta pomodoro 38th time month price [?] 60 Swiss fuckin' francs Damn go walking next afternoon Orelha Castle ate catfish lunch Cafe Reconcile side macaroni cheese cornbread collard greens Saw advertised channel 99 public access channel walked across street gym watched two fighters spar talked break sipped Snapple thought, life fight? test grace would match throwing bombs like Mike Tyson some, like Pernell, slippery [?] fearless like Gatti like Henry Akinwande buckle stall going gets tough, much due respect break cry Like Oliver "The Atomic Bomb" McCall Life's chess game us Hit, hit, jab hook feint bob weave fighters got back ring thought fight life time leaving damn go airport fly Cleveland, Ohio dinner Sylvester's North Canton girlfriend friends first time surrounded genuine smiles (beautiful smiles) table them, felt content grounded rooted dropped face hardships single mom happens one closest dearest friends Fell asleep spare room sound crop dusters cars highway Back roots unconditional love Rules everything could longer hear birds Films	0.982003
	Soon. Soon. Listen, I've drinking house lies ruin. know I'm Alone dark park pier. Watching ships disappear rain. world's bones. world black stones dressed rain place go home- like Nance. night like this, why, pro-stars, pro-sky. lit sick fighting Beneath diseased lighting discotheque night. mean thing. never means thing. mean thing. never means thing. got swing. I've seen all. I've seen all. Magnolia's girl. heart's made wood. apocalypses go, that's pretty good. Sha-la-la, say? Please remove spurs. Come think it, remove antlers. seen ages. still fly rages mention name, Christine	

Fetch and Highlight Lyrics of Song

Read Audio File and Display the Lyrics



Results



Language Detection: Implemented language detection for a dataset of 100,000 song records, using the **NLTK Corpus-Similarity model**. This approach processed the data **in under a minute** with high accuracy.



Text Preprocessing: Applied **Tokenization, Lemmatization, Stemming, and Vectorization** techniques, contributing to the enhancement of the Recommendation System.



Emotion Detection: Utilized attributes such as Danceability, Valence, Energy, and Liveliness from the Songs Dataset to analyze and **identify the emotional tone of each song**.



Model Training & Accuracy: Trained the dataset using multiple machine learning models and found Random Forest to be the most effective, achieving a consistent **99% accuracy**, even with cross-validation. This model is used to predict the emotion of a given song.



Song Recommendations: Leveraged the Cosine Similarity library to generate song recommendations, achieving a **similarity score of 0.9**, indicating highly relevant results.



Song Lyrics Translation: Implemented song translation and synchronized the display of lyrics while the song plays in the background, **yielding a 60% accuracy rate in results**.

Challenges Faced

- **Data Size Limitation:** The original dataset contained 1 million records, which made it challenging to implement on Google Colab due to resource constraints.
- **Language Detection:** The language detection process was time-consuming, particularly when using libraries and deep learning techniques to analyze the data.
- **Lack of References:** We faced difficulties finding reference code or existing solutions on platforms like Kaggle or GitHub, which slowed down the development process.
- **Vectorization Performance:** The TF-IDF model was very slow in processing the data, leading to significant delays. After thorough analysis, we switched to the GloVe model, which improved performance.
- **Speech Recognition:** During audio-to-lyrics translation, we encountered challenges with background noise interference, which made it difficult to accurately perform part-of-speech recognition and clean the data.

Conclusion



SUCCESSFULLY MERGED AND PREPROCESSED DATASETS, FILTERING FOR ENGLISH SONGS THROUGH EFFICIENT LANGUAGE DETECTION METHODS.



CREATED AN EMOTION DETECTION PIPELINE USING GLOVE EMBEDDINGS AND MACHINE LEARNING, WITH RANDOM FOREST DELIVERING THE BEST RESULTS.



BUILT A RECOMMENDATION SYSTEM THAT SUGGESTS SIMILAR SONGS BASED ON USER-INPUTTED LYRICS AND ATTRIBUTES, LEVERAGING COSINE SIMILARITY.



DEVELOPED A REAL-TIME CAPTION GENERATION TOOL, ALIGNING LYRICS WITH AUDIO TO ENHANCE USER EXPERIENCE.



DEMONSTRATED THE VALUE OF INTEGRATING NLP, ML, AND AUDIO PROCESSING FOR ADVANCED MUSIC ANALYTICS AND PERSONALIZED RECOMMENDATIONS.



FUTURE IMPROVEMENTS: REFINE EMOTION DETECTION, EXPAND LANGUAGE SUPPORT, AND ENHANCE CAPTIONING ACCURACY.

Project Management

Naga Venkata Kanakalakshmi (Contribution – 50%)	<ul style="list-style-type: none">• Analyzing and Gathering the Data• Loading and Merging the Datasets• Data Preprocessing & EDA• Sentiment Analysis• Recommendation System• Identifying the Lyrics of the song
Naga Sai Sivani (Contribution – 50%)	<ul style="list-style-type: none">• Analyzing and Gathering the Data• Language Detection Preprocessing• Language Detection• Sentiment Analysis• Recommendation System

References

- **Spotify Songs with Attributes and Lyrics:** Primary dataset containing song attributes and lyrics, essential for analysis in this project. Available on Kaggle:
<https://www.kaggle.com/datasets/bwandowando/spotify-songs-with-attributes-and-lyrics>
- **Langdetect:** A Python library for language detection. Available at <https://pypi.org/project/langdetect/>
- **Langid.py:** An off-the-shelf language identification tool. GitHub repository:
<https://github.com/saffsd/langid.py>
- **XLM-RoBERTa:** A transformer-based model for multilingual tasks. Documentation available at
https://huggingface.co/docs/transformers/model_doc/xlm-roberta
- **Langdetect:** A Python library for language detection. Available at <https://pypi.org/project/langdetect/>