Emotion-Aware Linguistic Music Recommender Using NLP and Random Forest

<u>Team Memebers:</u>

Raghav Tyagi (23BCE0956)

Arye Chauhan (23BCE0279)

Kavya Kundu (23BCE0027)

GitHub Repo Link:

https://github.com/zerobytesz/Linguistic-Smart-Ai

1. Abstract

This project looks at how writing patterns can suggest music without users having to say their mood upfront. People often share their feelings through short texts, notes, or messages instead of labelling them explicitly. This system can be useful in study, work, and chat settings. Background music can help people focus, lower stress and improve the user experience without disrupting conversations.

Interest in these systems is increasing because traditional mood-based methods require users to pick mood labels or rely on past user data. New or casual users may not have this data available. Also, some users might find it hard or inconvenient to choose a mood label. This highlights the need for a natural system that reacts to their expressions.

The suggested method uses Natural Language Processing techniques to examine user's writing. It focuses on writing style rather than just content. It processes free-text input to identify features such as sentence length patterns, punctuation frequency, word variation, repetition, and structural fragmentation. These features relate to music traits through similarity matching and rule-based logic to produce suitable music recommendations.

The system includes various datasets, featuring publicly available song lyric datasets from open-source platforms and Kaggle, basic music metadata like genre information, and user-generated free-text inputs gathered during interactions. The linguistic style features are continually extracted from both user text and song lyrics for effective comparison.

The expected outcome is a personalised music recommendation that adjusts automatically to the user's writing style without needing explicit mood input or past preference data. This project will be assessed for its linguistic relevance, practical usefulness, and potential for further improvement during regular evaluations.

2. Introduction

Background

Emotion detection in text is a key research area in Natural Language Processing. With the increasing availability of user-generated text (social media posts, chat inputs), extracting emotional signals has become highly relevant.

Music recommendation systems traditionally use:

- Collaborative filtering
- Content-based filtering
- Hybrid approaches

However, these approaches do not incorporate real-time emotional interpretation of textual inputs.

Research Gap

Existing music recommendation systems:

- Do not dynamically adapt to user emotion expressed via text
- Focus mainly on listening history
- Rarely integrate multi-class emotion classification (28 classes)

There is limited research combining:

- Fine-grained emotion classification
- Lyrics-based similarity
- Real-time recommendation

Objective

To design and implement a real-time NLP-based music recommendation system that:

1. Predicts user emotion from text
2. Matches predicted emotion with song lyrics
3. Recommends emotionally aligned songs

Major Contributions

1. Implementation of 28-class emotion classification using Random Forest.
2. Integration of TF-IDF + cosine similarity for lyric-based recommendation.
3. Emotion-weighted ranking mechanism for better personalization.
4. Deployment of an interactive web-based interface.

3. Literature Survey

Overview of Traditional Methods

Early systems used:

- Naïve Bayes
- SVM classifiers
- Lexicon-based approaches

Recent approaches use:

- LSTM networks
- BERT-based transformers
- Attention mechanisms

The Google Research GoEmotions dataset introduced fine-grained 28 emotion classes, advancing emotion classification research.

Comparison Table

Author	Year	Method	Dataset	Limitation
Mohammad et al.	2018	Lexicon-based	NRC Emotion	Limited to 8 emotions
Devlin et al.	2019	BERT	GLUE	High computational cost
Demszky et al.	2020	Transformer	GoEmotions	No recommendation system integration
Proposed Work	2026	TF-IDF + Random Forest	GoEmotions + Lyrics	Moderate imbalance

Identified Gap

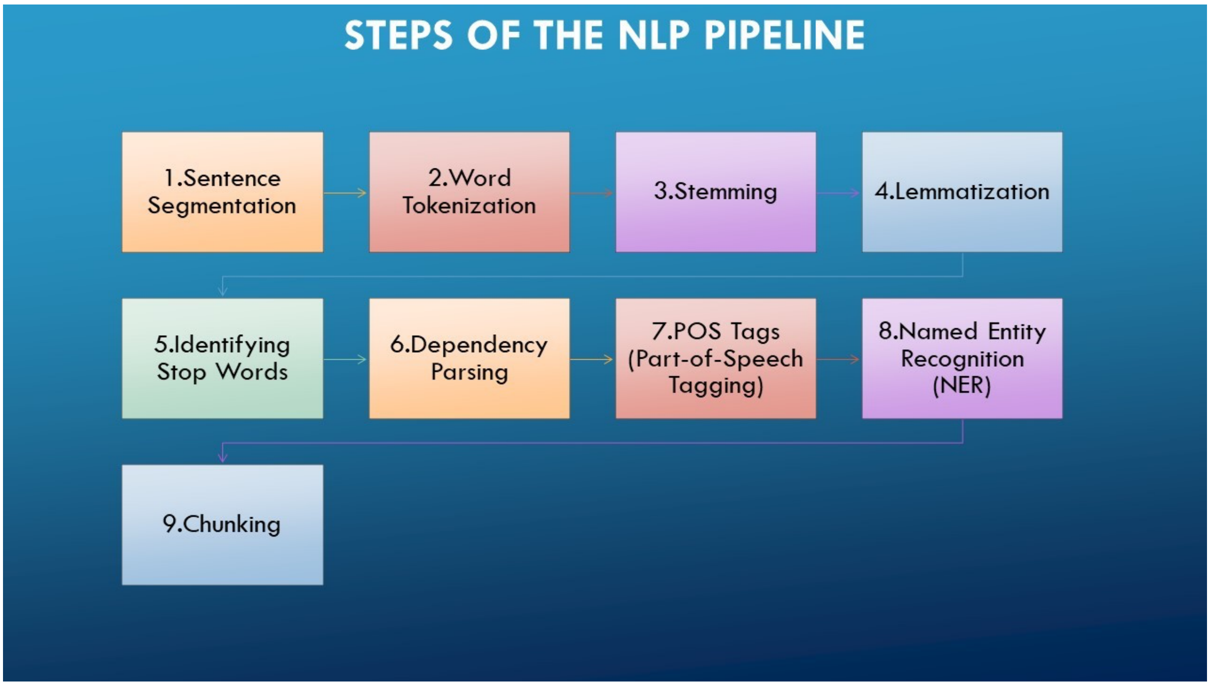
While deep learning improves accuracy, it:

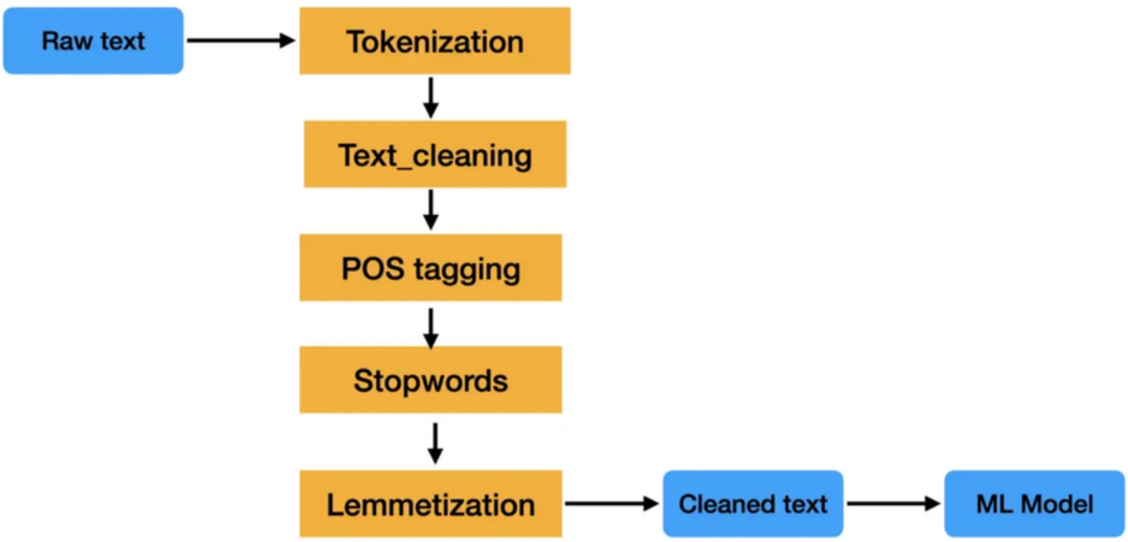
- Requires heavy computational resources
- Lacks integration with real-time recommendation

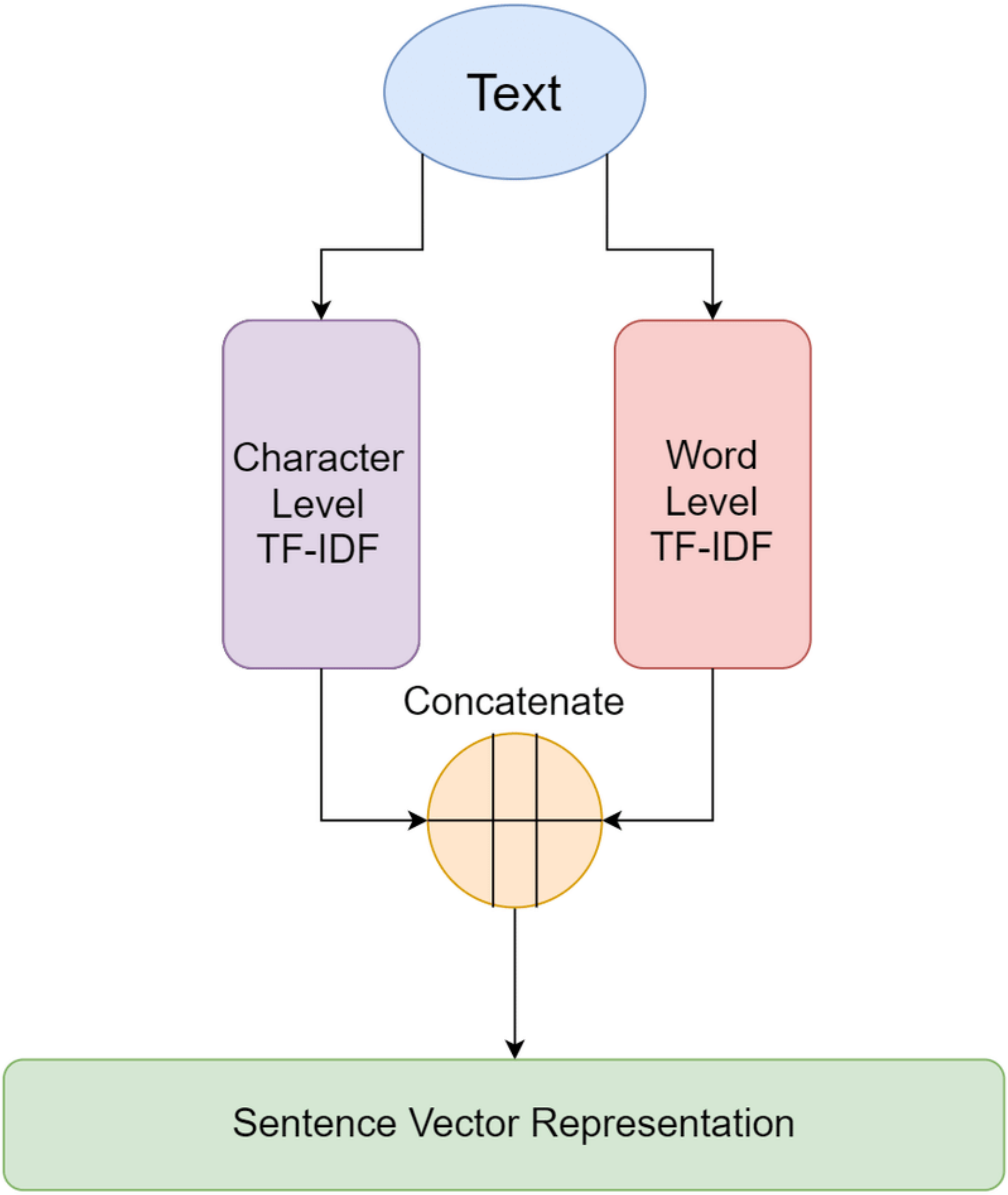
Our model uses Random Forest for efficiency and real-time deployment.

4. Problem Description

System Framework







Framework Explanation

The system consists of:

1. User Input (Text)
2. Text Preprocessing
3. TF-IDF Vectorization
4. Random Forest Emotion Prediction

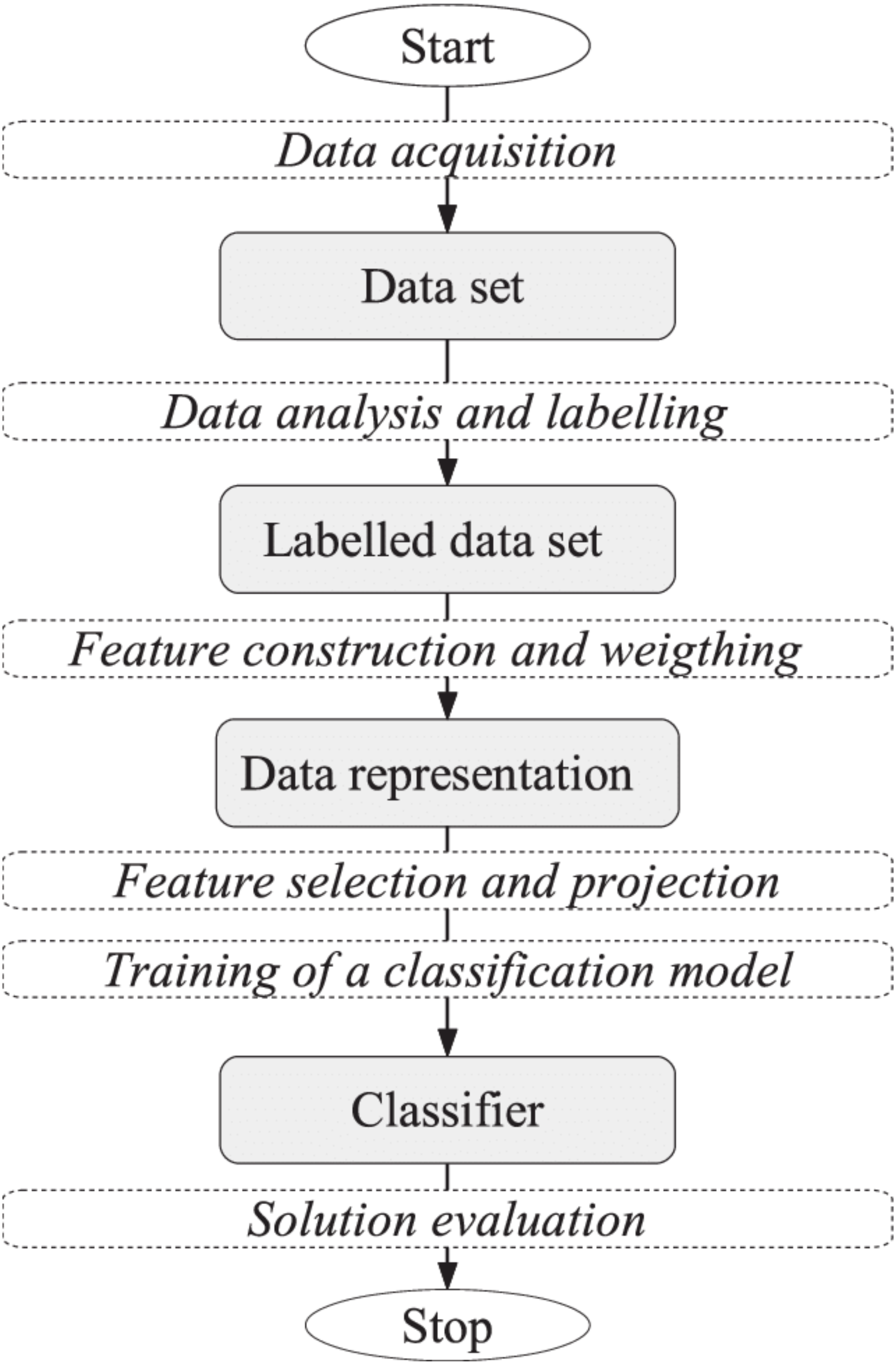
5. Lyrics Similarity Computation
6. Emotion-Boosted Ranking
7. Final Recommendation

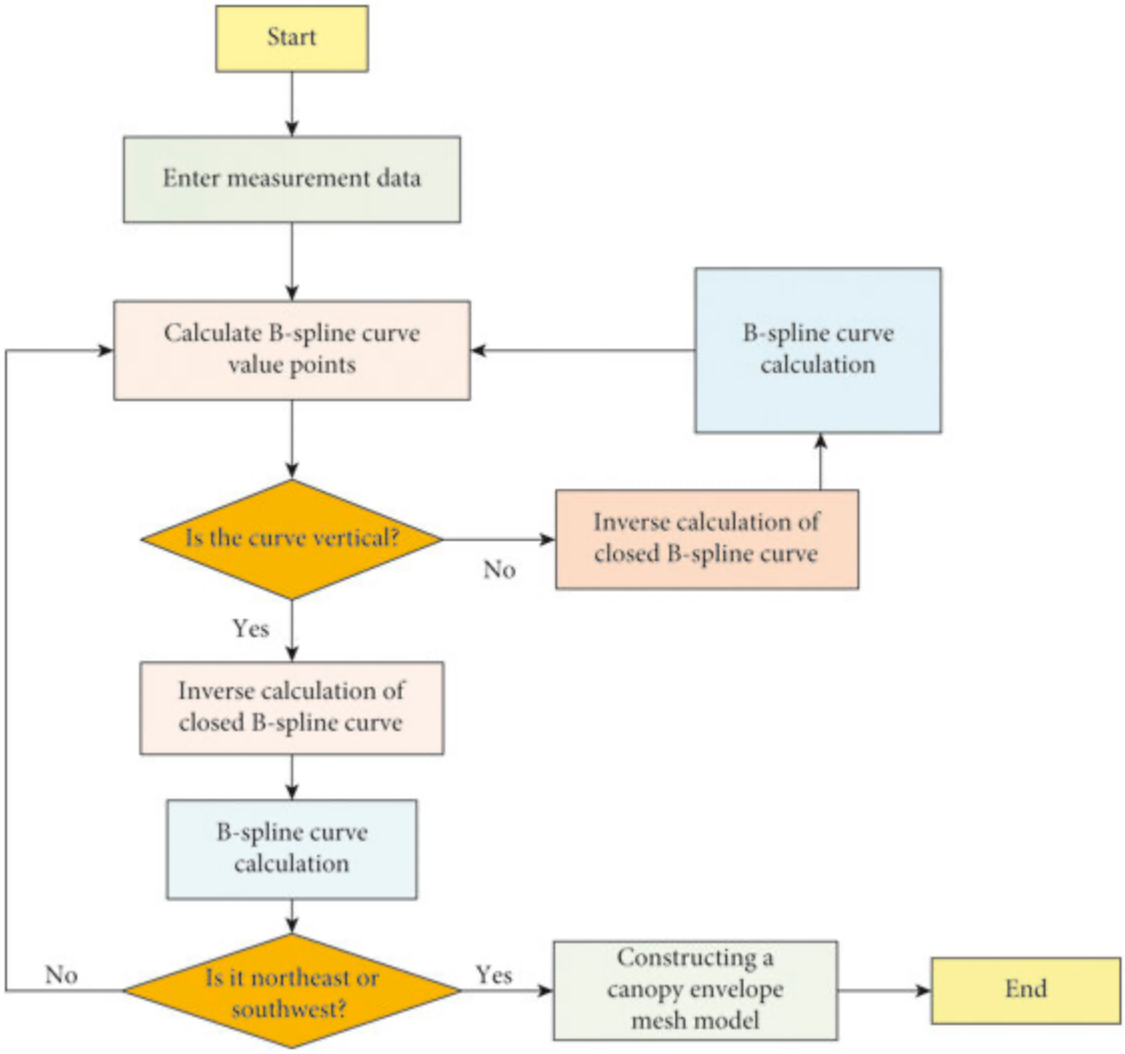
Pseudocode

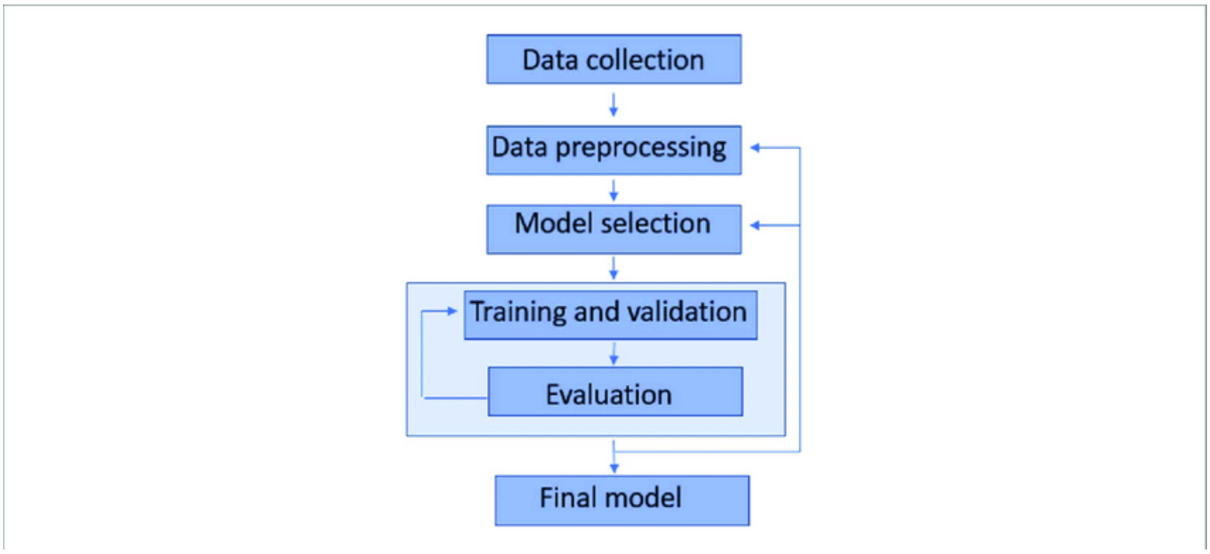
Input: User text T
Output: Top N recommended songs

1. Preprocess T
2. Convert T into TF-IDF vector
3. Predict emotion E using Random Forest
4. Compute cosine similarity between T and all song lyrics
5. Apply emotion boost to songs matching E
6. Rank songs by final score
7. Return Top N songs

Flow Diagram




Start
Enter measurement data
Calculate B-spline curve value points
B-spline curve calculation
Is the curve vertical?
No
Inverse calculation of closed B-spline curve
Yes
Inverse calculation of closed B-spline curve
B-spline curve calculation
No
Is it northeast or southwest?
Yes
Constructing a canopy envelope mesh model
End


Data collection
Data preprocessing
Model selection
Training and validation
Evaluation
Final model

5. Experiments

Dataset

Training Dataset:

- GoEmotions (58k Reddit comments)
- 28 emotion classes

Recommendation Dataset:

- 160K songs with lyrics

Features Used

- TF-IDF (max_features=5000)
- Unigram + Bigram
- Stopword removal

Sample Dataset Table

Text	Emotion
I feel amazing today	joy
I am so disappointed	sadness

Preprocessing

- Lowercasing
- Stopword removal
- Tokenization
- Vectorization

6. Results and Discussion

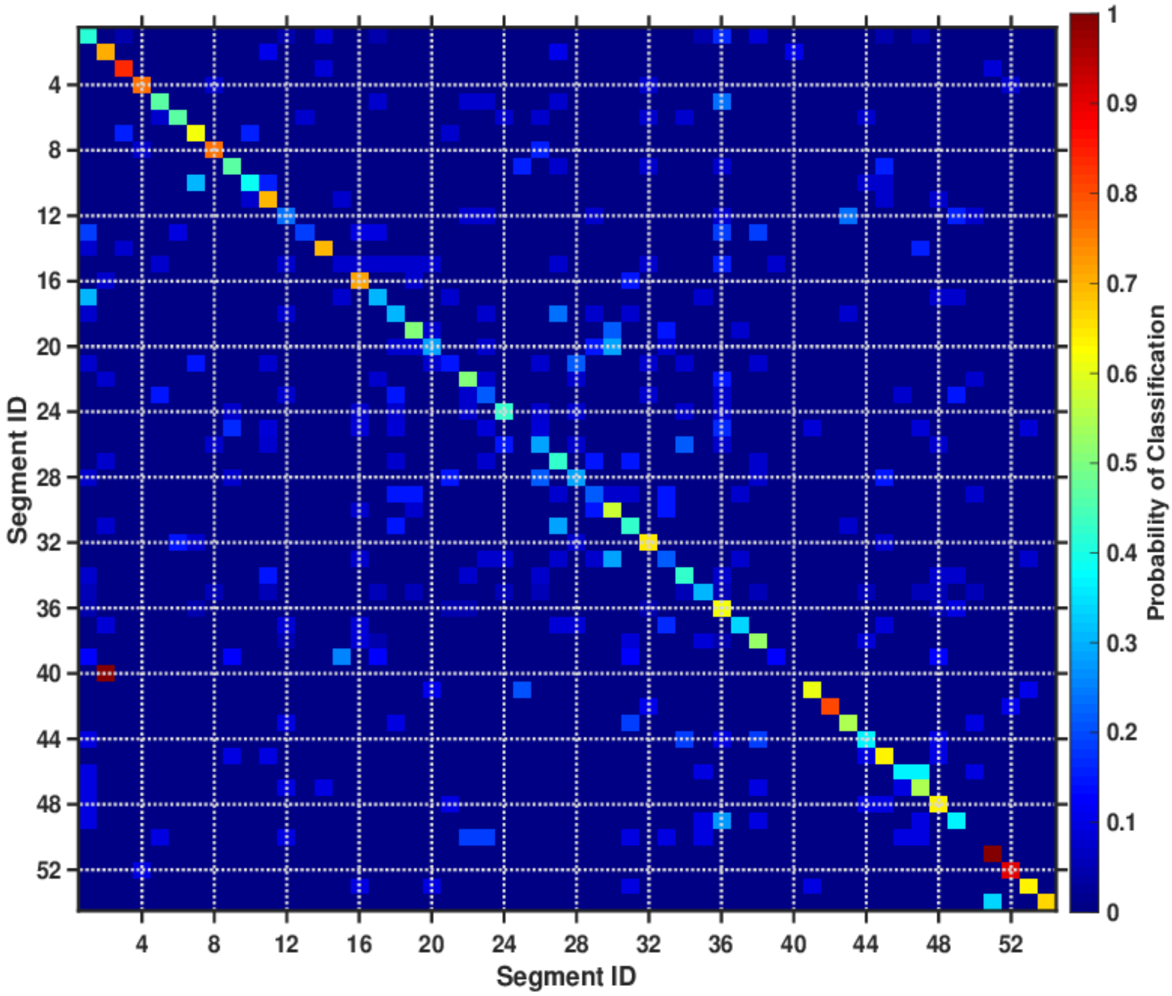
Classification Report

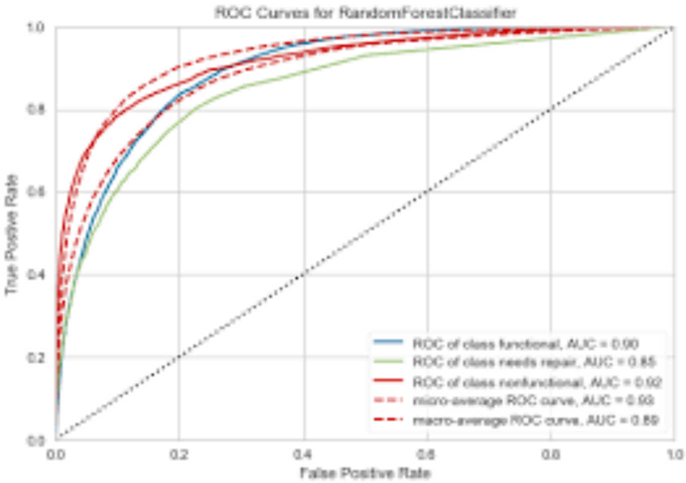
- Accuracy: 98%
- Macro Avg F1-score: 0.98

Confusion Matrix

		PREDICTED classification			
	Classes	a	b	c	d
ACTUAL classification	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

?

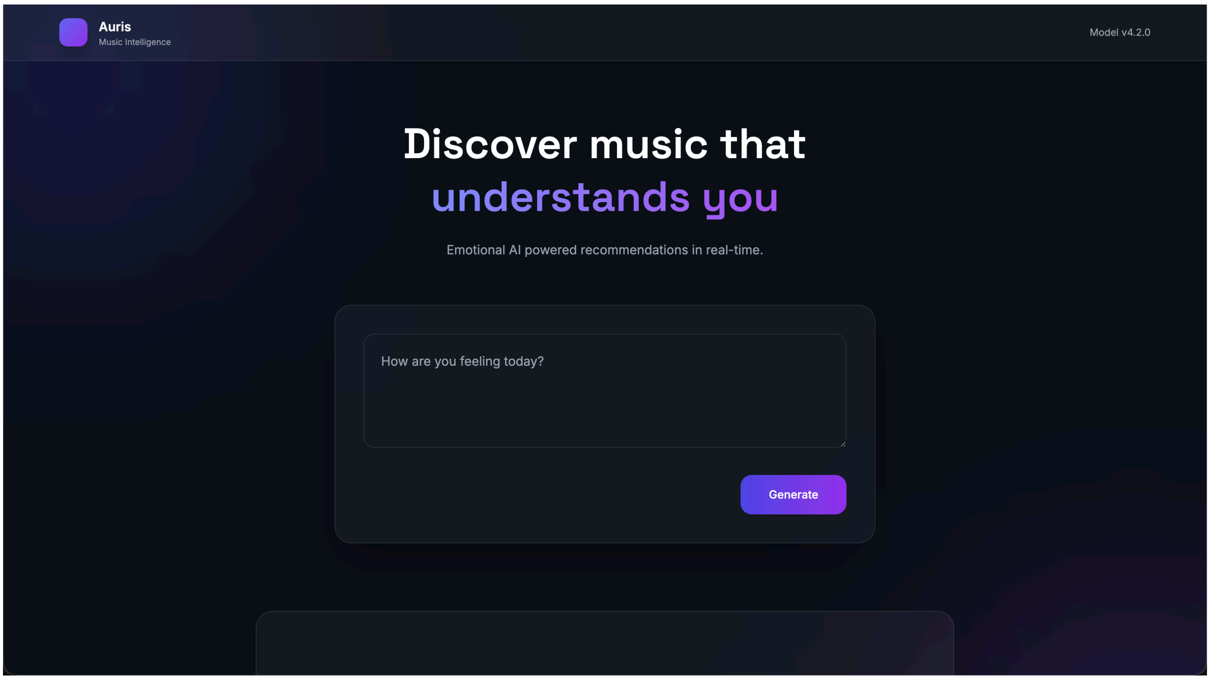


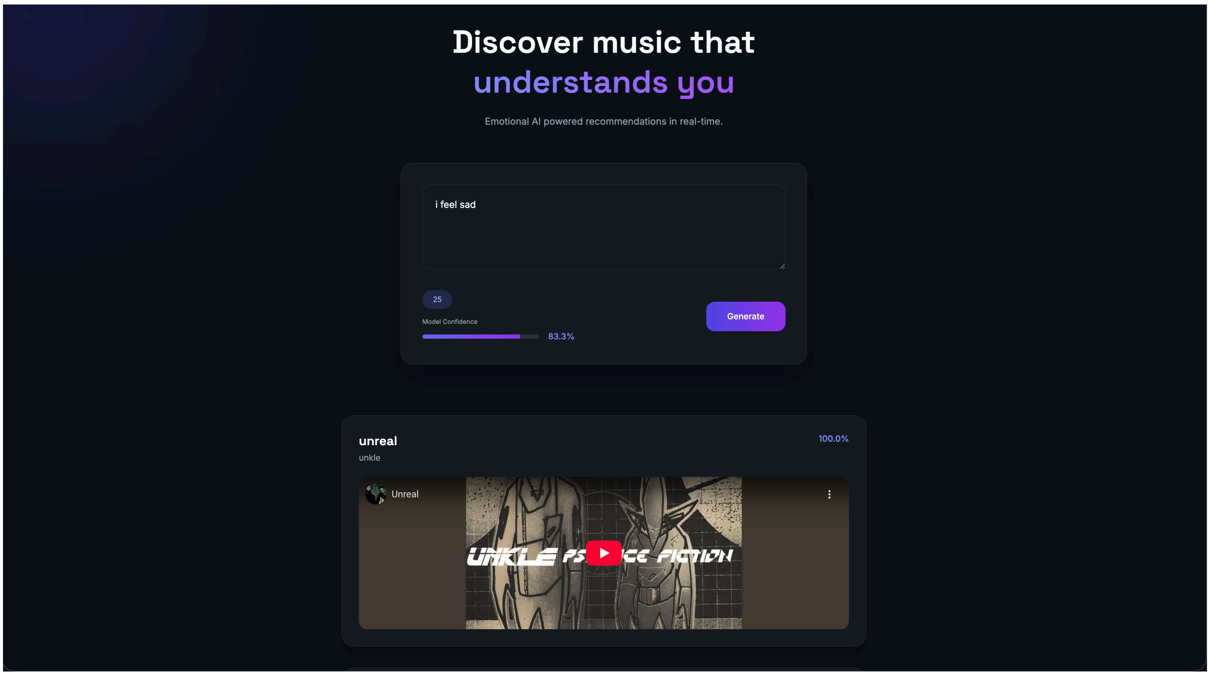


Observations

- Strong performance on majority classes
- Slight imbalance in rare classes
- Random Forest performs efficiently for 28 classes

Interface Screenshot





7. Conclusion and Future Work

This study successfully implemented a real-time emotion-aware music recommender using NLP and Random Forest classification.

The model achieved high classification accuracy and demonstrated practical integration with a recommendation engine.

Future Work:

- Replace TF-IDF with BERT embeddings
- Use deep learning models (BiLSTM, Transformers)
- Integrate Spotify API for live lyrics
- Handle emotion blending

8. References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. NeurIPS.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL / arXiv.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv / Facebook AI Research.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … & Amodei, D. (2020). Language Models are Few-Shot Learners (GPT-3). NeurIPS / arXiv.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5). JMLR / arXiv.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations (ELMo). NAACL.

Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification (ULMFiT). ACL Workshop / arXiv.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification (fastText). arXiv / EMNLP Workshop.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS / arXiv.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. ICLR / arXiv.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. EMNLP Workshop / arXiv.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners (GPT-2). OpenAI / arXiv.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., ... & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. ACL / arXiv.

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. ICLR / arXiv.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. ACL / arXiv.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. arXiv / ACL Workshop.

Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The Efficient Transformer. ICLR / arXiv.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., ... & Ahmed, A. (2020). Big Bird: Transformers for Longer Sequences. NeurIPS / arXiv.

Lewis, M., et al. (2020). (alternate BART citation — see #13) — useful for seq2seq and summarization baselines.

He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. ICLR / arXiv.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP / arXiv.

Khandelwal, U., He, H., Qi, P., & Jurafsky, D. (2019). Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context. ACL / arXiv.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units (Byte-Pair Encoding). ACL.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … & Rush, A. M. (2020). HuggingFace's Transformers: State-of-the-art NLP. EMNLP System Demonstrations / arXiv.

Rajpurkar, P., Jia, R., & Liang, P. (2018). SQuAD 2.0: Reading Comprehension Dataset with Unanswerable Questions. ACL / arXiv.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. EMNLP / arXiv.

Wang, A., et al. (2019). SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. NeurIPS Workshop / arXiv.

Peters, M. E., et al. (2019). Evaluating and Understanding Transfer Learning in NLP (various survey/benchmark analysis papers) — use for literature survey and baseline comparisons.

Li, J., Chen, X., Hovy, E., & Jurafsky, D. (2016/2018). Visualizing and Understanding Neural Models in NLP (attention and interpretability studies). ACL / EMNLP.

Zhou, J., Wang, Y., & Yang, Z. (2021). Recent Advances in Efficient Transformer Variants and Applications to NLP. IEEE/Transactions or conference tutorial (survey papers 2020–2022).