Predicting a Song's Popularity Period

FINAL PRESENTATION

Tarun Vallabhaneni
tvallabh@uchicago.edu
MPCS 53120
Applied Data Analysis

INTRODUCTION

Significance

Aid stakeholders in the music industry—such as radio stations, record labels, and streaming platforms—to optimize music selection and marketing strategies.

Objective

Utilize advanced machine learning techniques to predict the popularity period of songs.

Scope

Analyze datasets to understand the factors influencing a song's hit potential and temporal popularity dynamics using a variety of models.

Conflicting Literature:

1) Pachet and Roy concluded that the popularity of a song cannot be learnt by using machine learning (no feature selection)
2) Salganik, Dodds, and Watts say that quality of a song only partially influences popularity
3) Pham, Kyauk and Park found that while using EchoNest's popularity metric, Lasso regression (using shrinking) gave the lowest test error (with feature selection and 10-fold cross validation).

DATASET USED

Kaggle Dataset

Spotify API

Description: Contains detailed audio features, metadata, and genre information for over 600,000 songs spanning nearly a century.

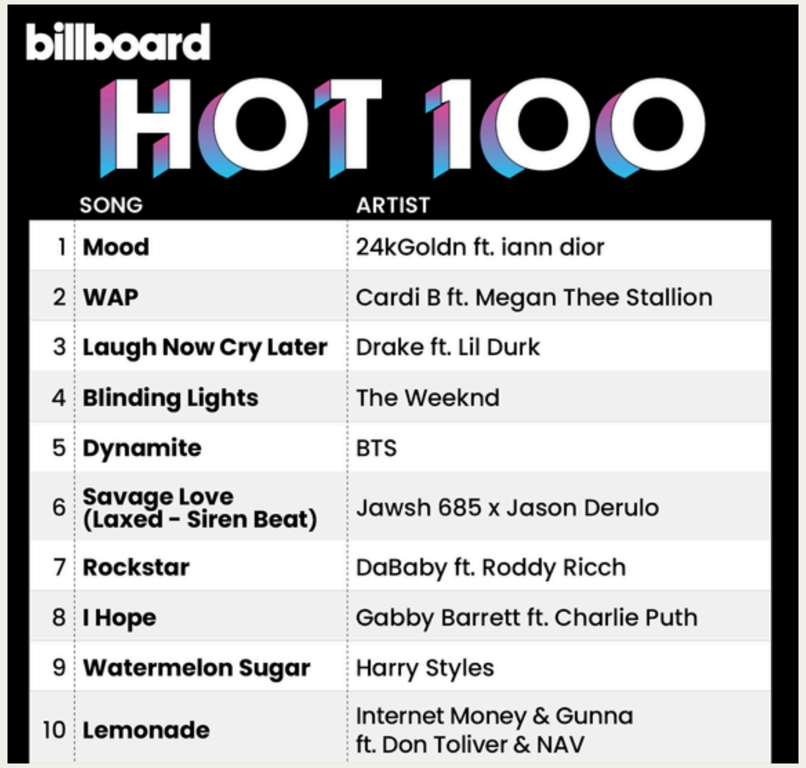
Key Features: Tempo, duration, loudness, energy, danceability, valence, and genre tags.

GitHub Dataset

Billboard Hot 100

Description: Historical popularity metrics for songs appearing on the Billboard Top 100, providing an objective benchmark for song popularity. Scraped using Python and BeautifulSoup.

Integration: Used as a benchmark to classify songs as hits.



billboard HOT 100

	SONG	ARTIST
1	Mood	24kGoldn ft. iann dior
2	WAP	Cardi B ft. Megan Thee Stallion
3	Laugh Now Cry Later	Drake ft. Lil Durk
4	Blinding Lights	The Weeknd
5	Dynamite	BTS
6	Savage Love (Laxed - Siren Beat)	Jawsh 685 x Jason Derulo
7	Rockstar	DaBaby ft. Roddy Ricch
8	I Hope	Gabby Barrett ft. Charlie Puth
9	Watermelon Sugar	Harry Styles
10	Lemonade	Internet Money & Gunna ft. Don Toliver & NAV

1) Data Cleaning

Removed duplicate songs based on name and artist, handled missing values by dropping incomplete rows, and removed songs without genre information.

2) Feature Engineering

Converted release dates to decade of release, categorized songs into Pre-2000s Hits, Post-2000s Hits, and Non-Hits. Also constructed collinearity matrix for the features.

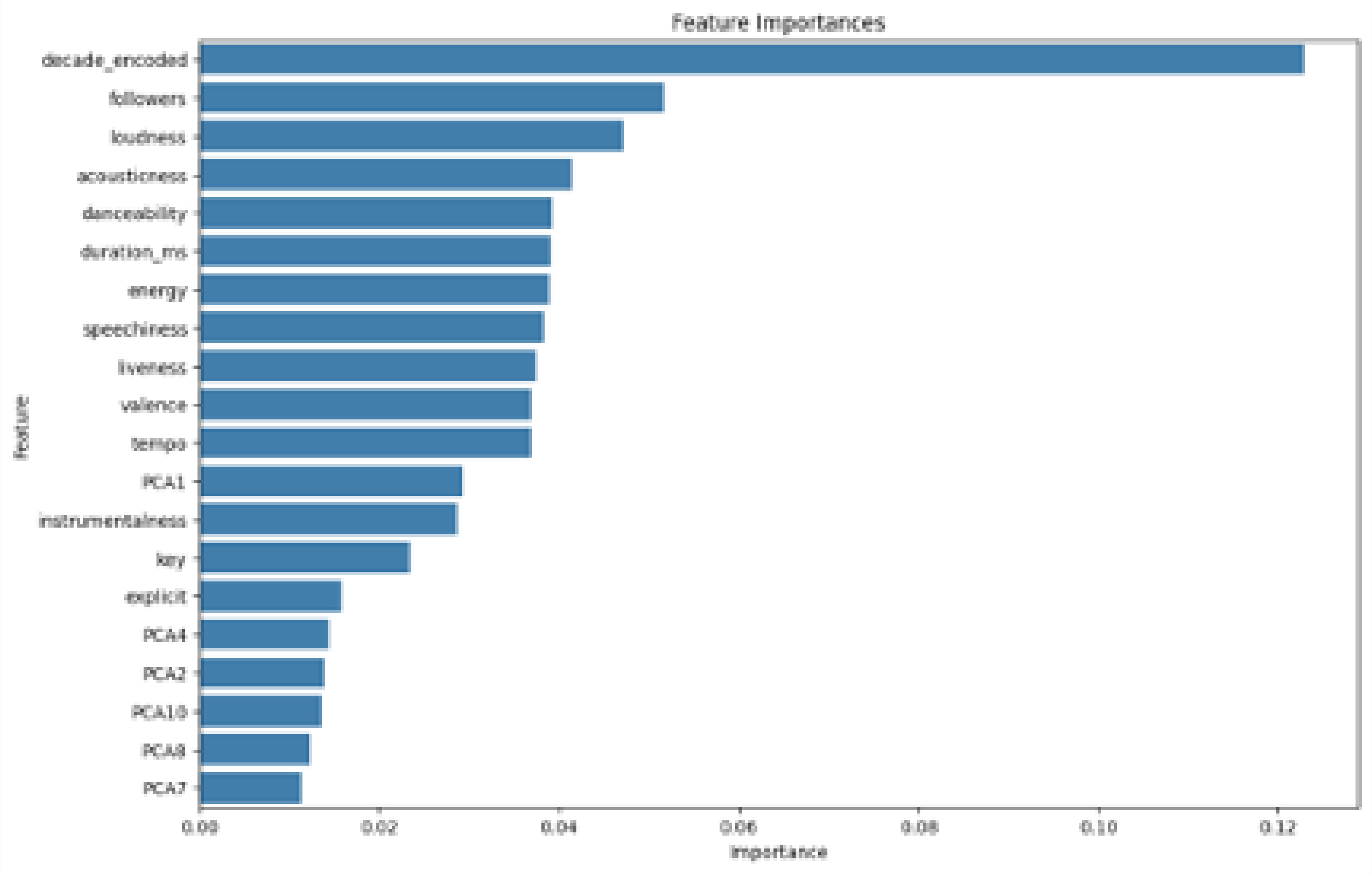
3) Genre Embeddings

Generated 768-dimensional embeddings for genre tags using Sentence Transformers and applied PCA for dimensionality reduction.

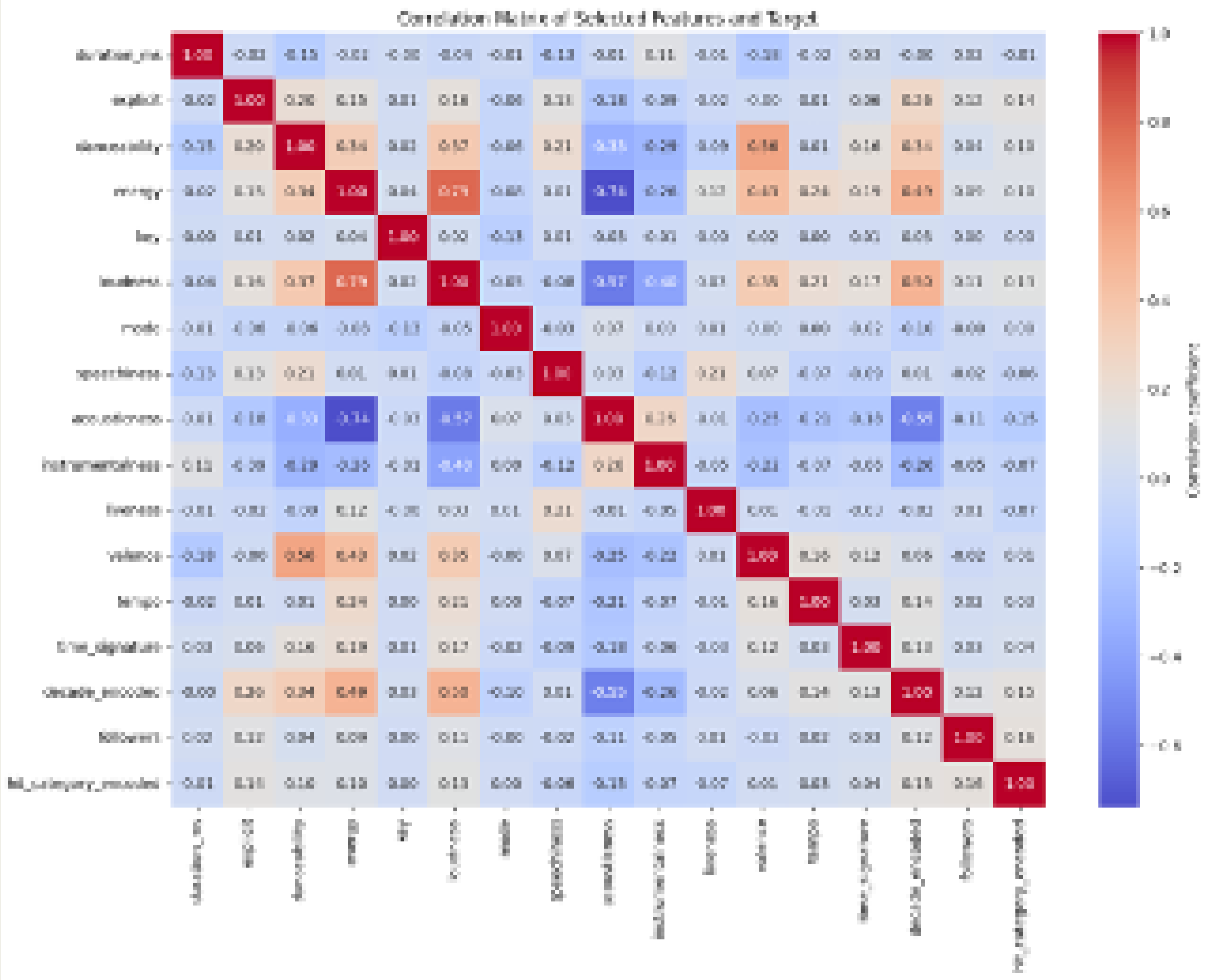
4) Class Imbalances

Improved my popularity metric by integrating Spotify-provided popularity. Applied SMOTE to address class imbalances, ensuring a balanced representation of hits and non-hits.

FEATURE INSIGHTS



Twenty Most Important Features (post-PCA)



Correlation Matrix (excluding genre embeddings)

INITIAL (FAILED) ATTEMPTS

Random Forest Model Without Genres

- Trained an initial Random Forest model without genre classification, with poor performance, especially for pre-2000 hits (F1 score of 0.10).

Interpretation of Results

- Hypothesized that incorporating genre information could improve performance.

Initial One-Hot Encoding Approach

- Initially classified songs by decade using one-hot encoding, resulting in poor F1 scores for older decades.

Changing Classification Approach

- Revised to categorize songs into Pre-2000s hits, Post-2000s hits, and Non-hits.

Preliminary Analysis of Genres

- Over 4,672 unique genres made a dictionary approach infeasible.
- Used Natural Language Processing (NLP) techniques for genre tags.

Using Sentence Transformers

- Generated 768-dimensional embeddings for genre tags.

Dimensionality Reduction with PCA

- Applied PCA to reduce genre embeddings to 50 features.
- Prevented potential overfitting and maintained model accuracy.

Random Forest Model With Genres

- Included genre embeddings, significantly improving model performance.
- Demonstrated the effectiveness of incorporating genre information.
- Potential for lyrical information to be added.

Models and Results

"We think we’ve figured out how the brain works regarding music taste

- Mike McCready, Founder of Polyphonic HMI

RANDOM FOREST

85.8%
Accuracy

0.68
Macro F1-Score

Class 2 (Pre-2000s Hits)
Poorest Class Performance

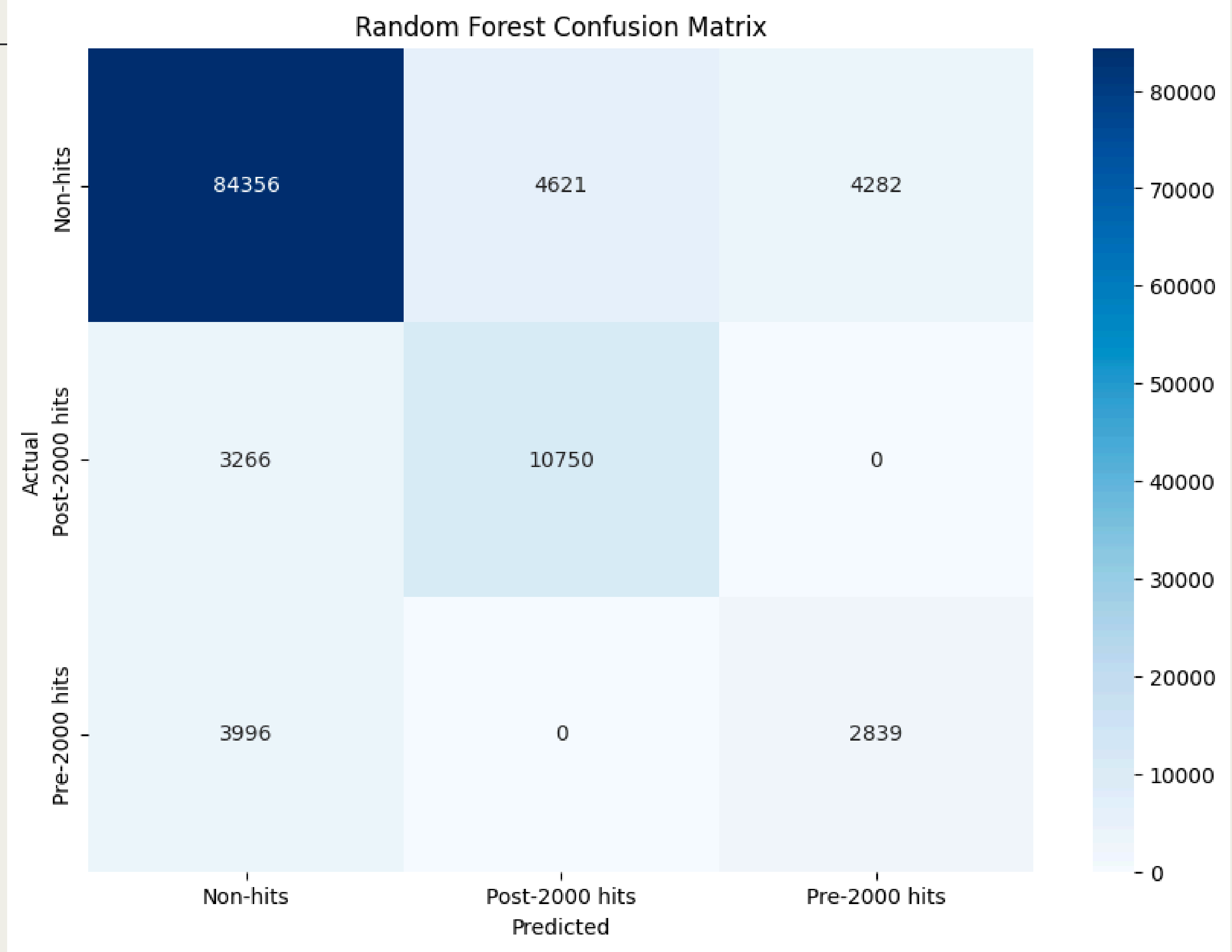
SMOTE
Better Performance

Genre Embeddings
F1-Score increases for hit classes

100 trees
Hypertuning

No Cross-Classification
Model Limitation

Metric	Class 0 (Non-hits)	Class 1 (Post-2000 hits)	Class 2 (Pre-2000 hits)	Macro Average	Weighted Average
Precision	0.92	0.70	0.40	0.67	0.86
Recall	0.90	0.77	0.42	0.70	0.86
F1-score	0.91	0.73	0.41	0.68	0.86



XGBOOST

85.1%
Accuracy

Class 2 (Pre-2000s Hits)
Slight Differences

Random Forest Similarities
Across All Classes and Scores

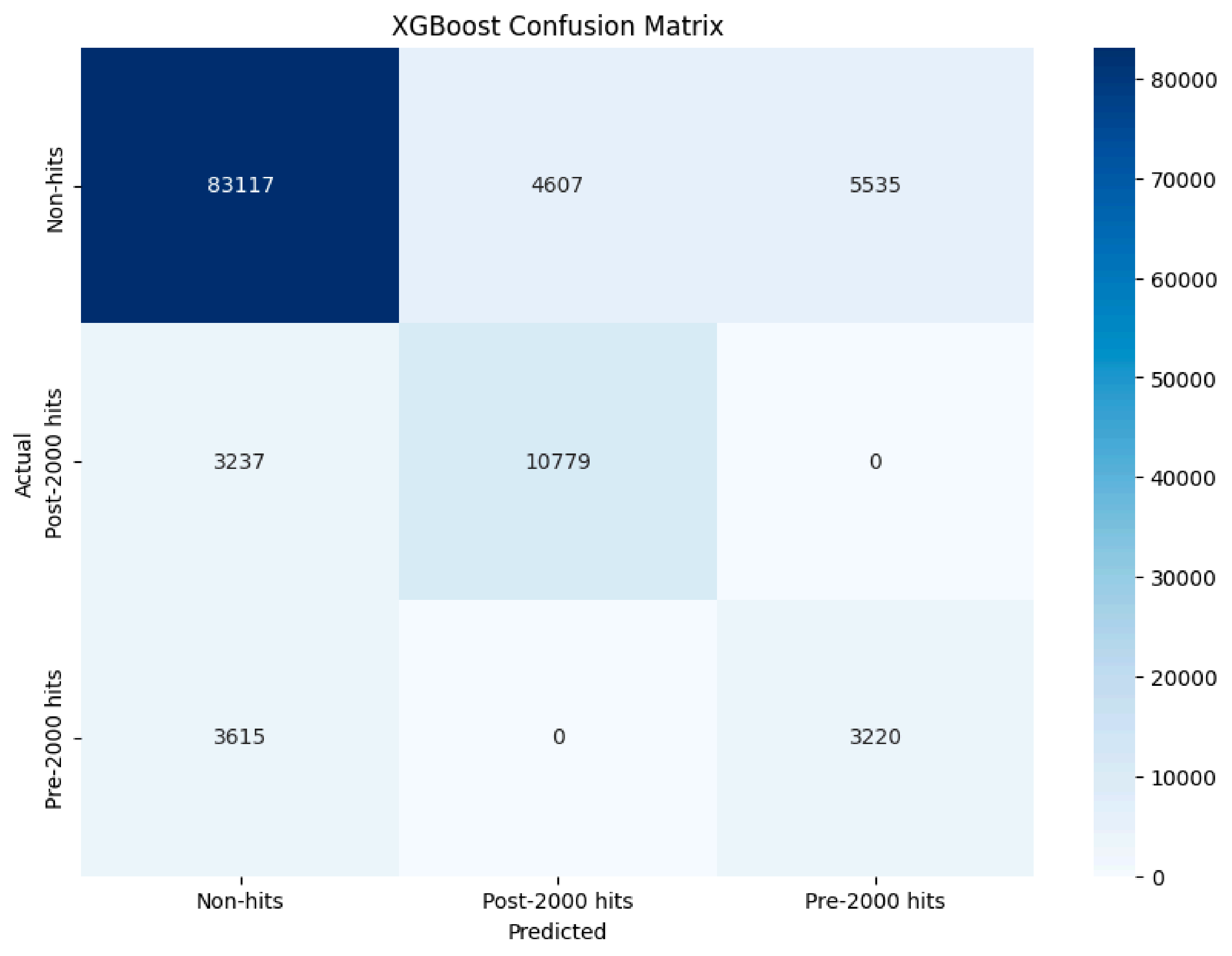
0.68
Macro F1-Score

SMOTE
Better Performance

500 trees
Hypertuning

No Cross-Classification
Model Limitation

Metric	Class 0 (Non-hits)	Class 1 (Post-2000 hits)	Class 2 (Pre-2000 hits)	Macro Average	Weighted Average
Precision	0.92	0.70	0.37	0.66	0.86
Recall	0.89	0.77	0.47	0.71	0.85
F1-score	0.91	0.73	0.41	0.68	0.86



KNN

76.6%
Accuracy

Poor Performance
Across All Classes

Reduced Features
With High Correlation

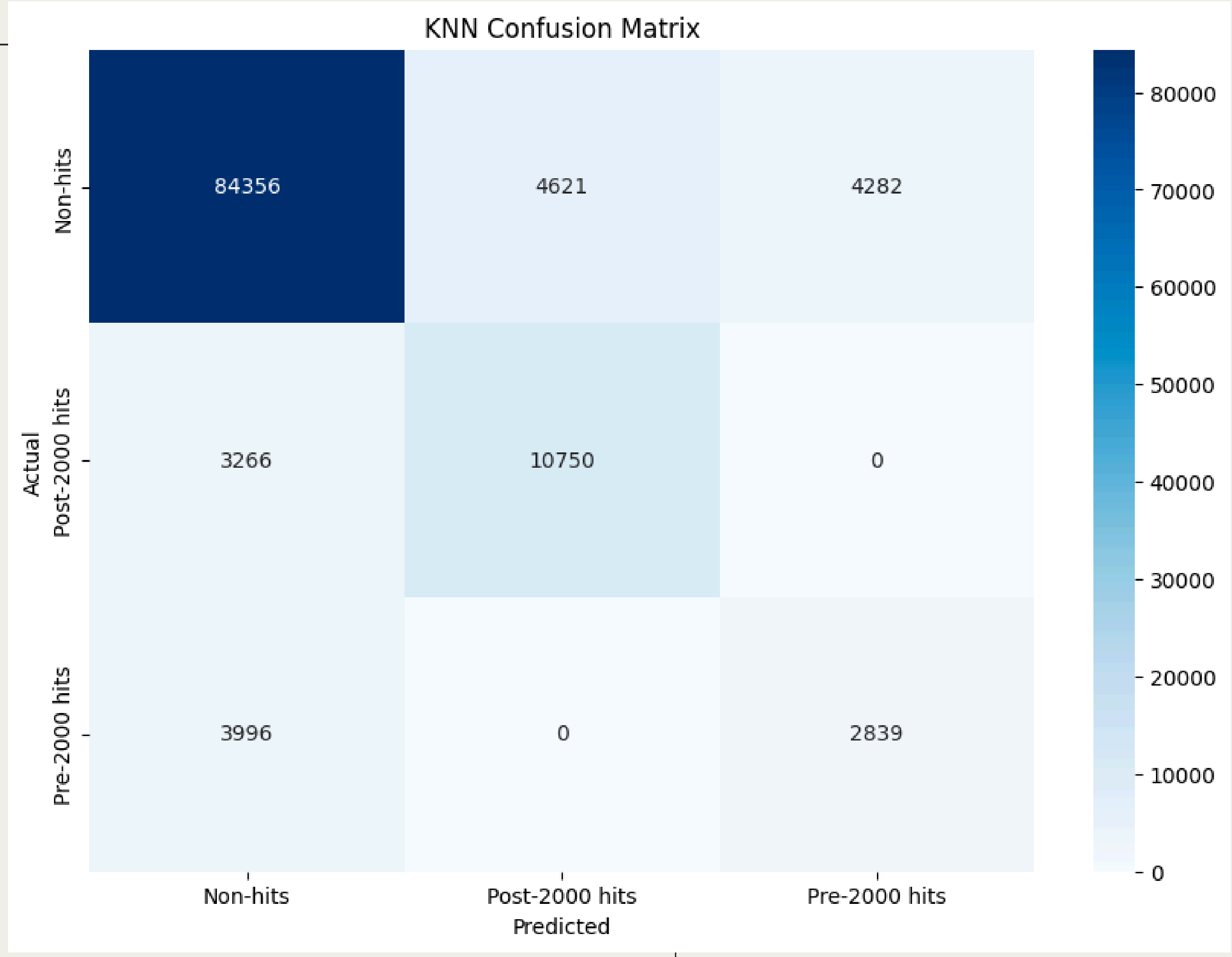
0.61
Macro F1-Score

Scaled Data
Distance-Based Algorithm

3 neighbours
Hypertuning

No Cross-Classification
Model Limitation

Metric	Class 0 (Non-hits)	Class 1 (Post-2000 hits)	Class 2 (Pre-2000 hits)	Macro Average	Weighted Average
Precision	0.93	0.53	0.25	0.57	0.84
Recall	0.78	0.78	0.57	0.71	0.77
F1-score	0.85	0.63	0.34	0.61	0.79



NEURAL NETWORK

81.1%
Accuracy

0.68
Macro F1-Score

Class 2 (Pre-2000s Hits)
Best Model Performance

128-64-32
Architecture

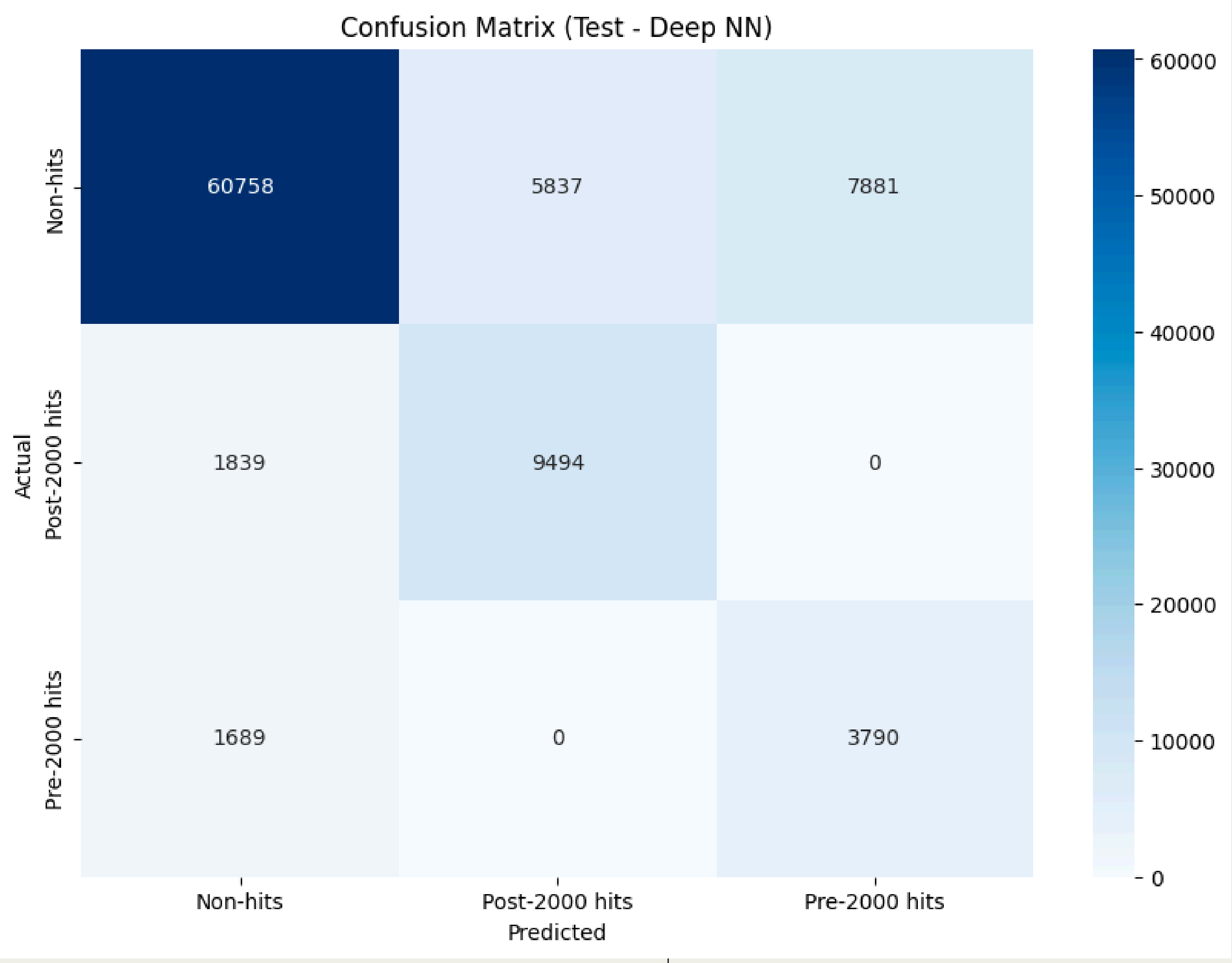
Feed-Forward Network
Sequential Class in Keras

0.001
Learning Rate

Cross-Entropy
Loss Function

30
Epochs

Metric	Class 0 (Non-hits)	Class 1 (Post-2000 hits)	Class 2 (Pre-2000 hits)	Macro Average	Weighted Average
Precision	0.95	0.62	0.32	0.63	0.87
Recall	0.82	0.84	0.69	0.78	0.81
F1-score	0.88	0.71	0.44	0.68	0.83



INSIGHTS

Easier Prediction
of Non-hits

Challenges
with Pre-2000s Hits

No Cross Categorization
Between Hit Classes

Decent Performance
of Post-2000s Hits

Neural Networks
Potential, Highest Pre 2000s Hits F1 Score

Unbalanced Dataset
SMOTE

Surprsing Model Performance
With Limited Features

Random Forest, XGBoost
Best Performing Models

Genre Embeddings
Greatly Improved Performance

Predictability in New Music
With Post-2000s Hits

LIMITATIONS

1) Dataset Constraints

Issue: Missing detailed audio features due to the inability to use the Million Song Dataset.
Impact: The current low-dimensional fields may not capture the full complexity of songs, affecting model accuracy.

2) Data Inaccuracy

Issue: Instances of inaccurate data, such as low "speechiness" in rap songs.
Impact: These inaccuracies could skew model learning and predictions.

3) Lack of Lyrical Information

Issue: Rate limits on the Genius API prevented incorporating lyrical content.
Impact: Lyrics could provide additional context and improve prediction accuracy.

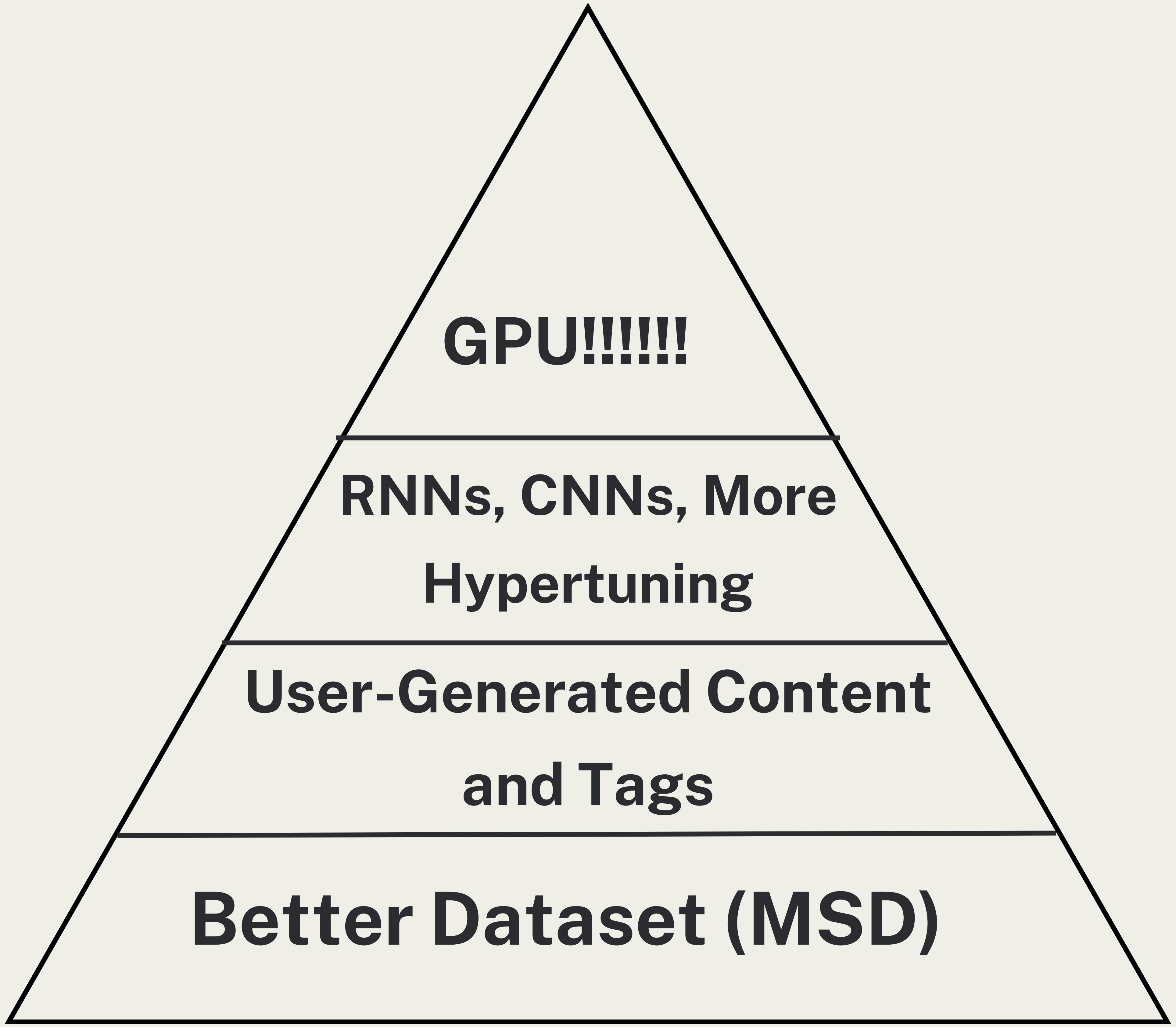
4) Genre Embeddings Dimensionality

Issue: High-dimensional genre embeddings had to be reduced to avoid a large dataframe.
Impact: Limited the richness of genre information available for model training.

5) Computational Limitations

Issue: Running SVMs and more complex neural networks on the large dataset was infeasible with only a CPU.
Impact: Exclusion of SVM results prevented additional insights from being included in the analysis.

FUTURE WORK



Possible Directions

Temporal Dynamics: How do factors influencing song popularity change over time, and can comprehensive datasets predict these changes more accurately?

Cultural Influence: Impact of cultural and regional differences on song popularity and integrating these factors into models.

Interactivity and User Preferences: Incorporating real-time user interactions for dynamic and personalized music recommendations.

Cross-Classification: Can songs be categorized into actual decades and predict if current songs could have been past hits or which past hits could blow up in popularity today?

Thank you!

FINAL PRESENTATION

Tarun Vallabhaneni
tvallabh@uchicago.edu
MPCS 53120
Applied Data Analysis