Problem Statement

In today's fast-paced world, music listeners often desire a quick and engaging way to discover new songs. *[Music Streaming Company]* aims to address this need by introducing a new product feature that plays reels of song choruses, enabling rapid and intentful music discovery. To make this feature successful, an accurate and efficient automated chorus detection model is crucial. The development of such a model will not only enhance user experience but also provide valuable insights into the structure and characteristics of popular music.

Data Wrangling

The dataset used for this project consists of 332 manually labeled songs, predominantly from electronic music genres. The annotation process followed a comprehensive guide that outlined the definition of a chorus, criteria for chorus labeling, and the annotation process itself. The data wrangling steps included:

1. Audio Preprocessing:
    a. Songs were uniformly formatted (e.g., '.mp3') and processed at a consistent sampling rate (48000 Hz). Leading and trailing silences were trimmed, and relevant metadata was extracted using Spotify's API.
2. Manual Chorus Labeling:
    a. Chorus sections were labeled using Serato DJ software, which provided playback with spectrogram visualizations and beat/bar quantization. Start and end labels for each chorus were documented in the format (hh:mm:ss.ms), with milliseconds rounded to one decimal place.
    b. Tracks with ambiguous musical structures or thematic elements were noted and skipped to maintain annotation quality. Songs with more than three choruses were also skipped and their chorus segments re-evaluated.

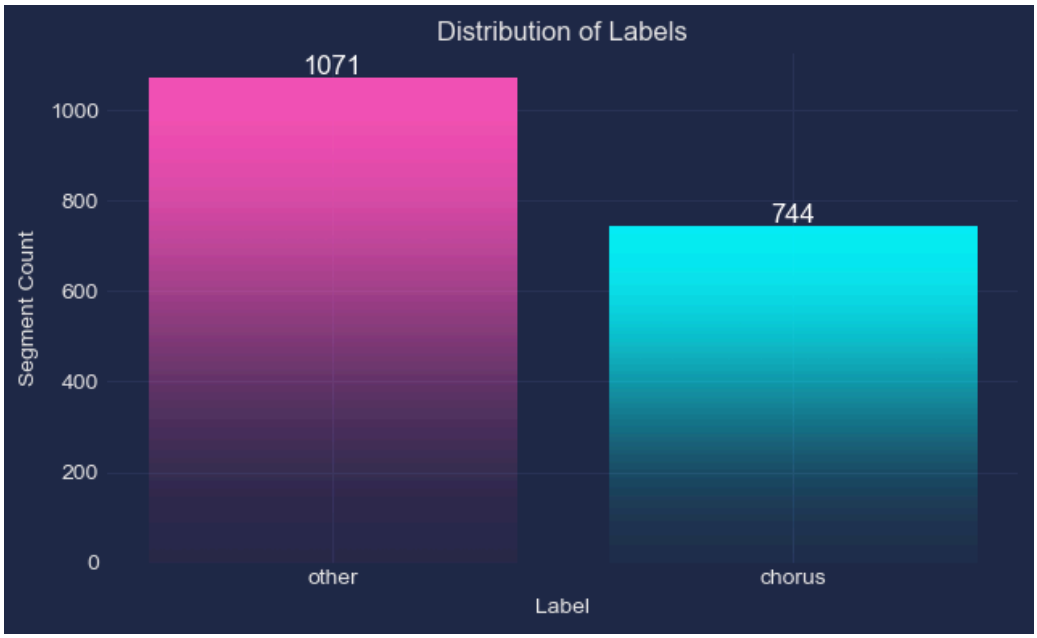
Exploratory Data Analysis

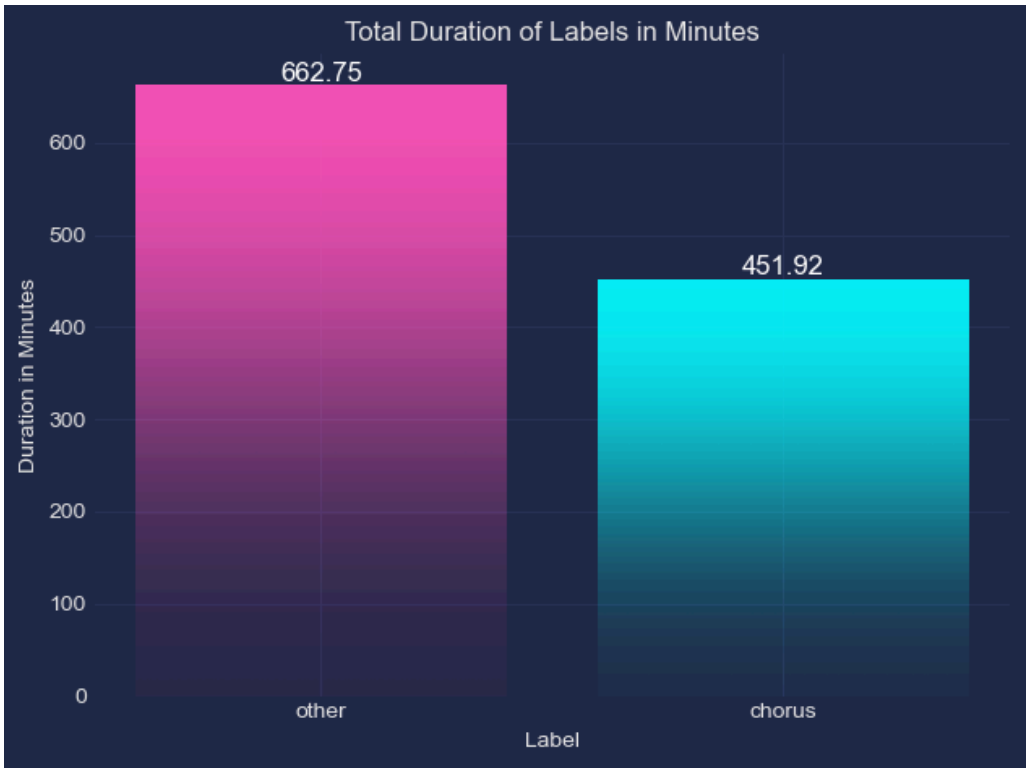
The exploratory data analysis aimed to uncover insights and patterns within the dataset to inform the development of an accurate and efficient automated chorus detection model. The analysis encompassed the following key areas:

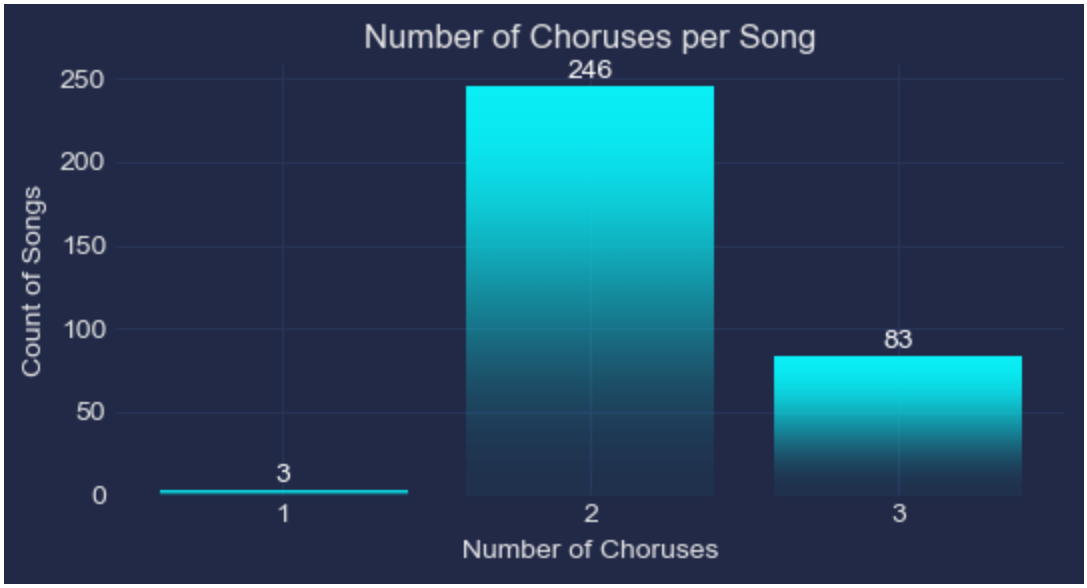
- Data Profile
- Label Validation
- Audio Feature Visualization and Analysis

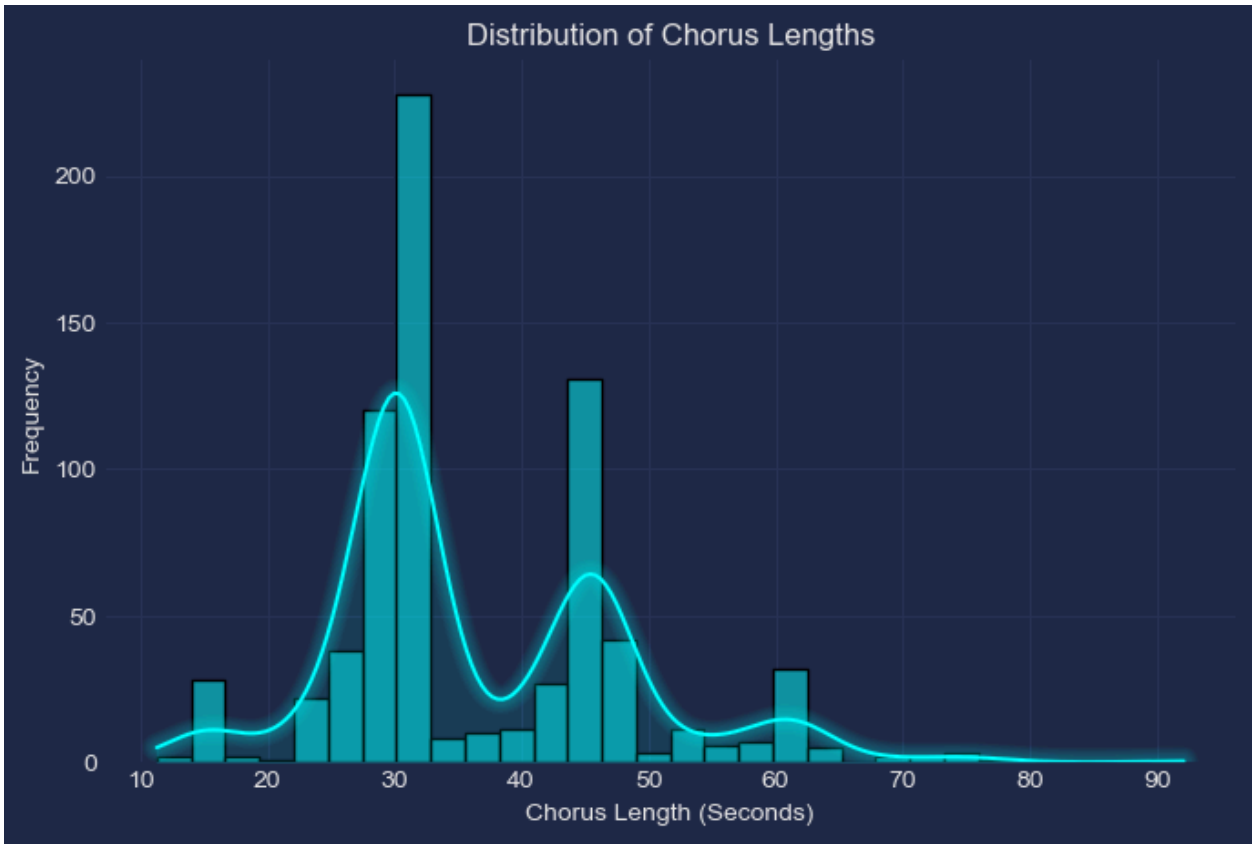
Data Profile

What does the distribution of the labeled choruses look like?

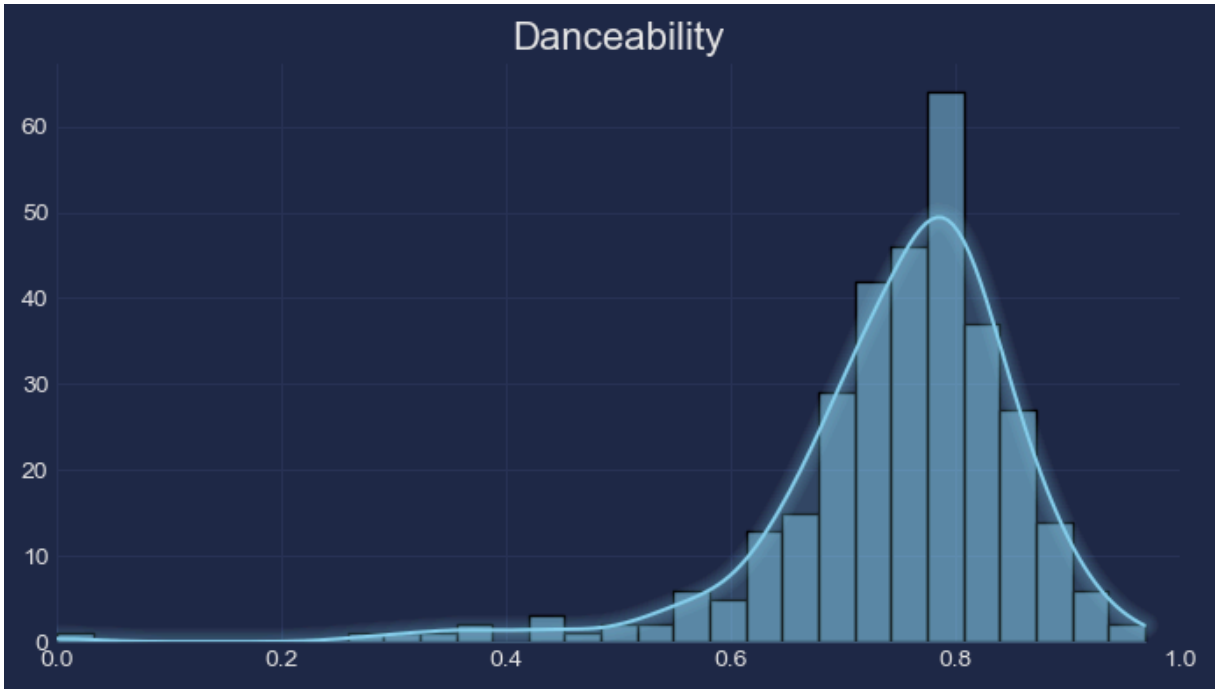


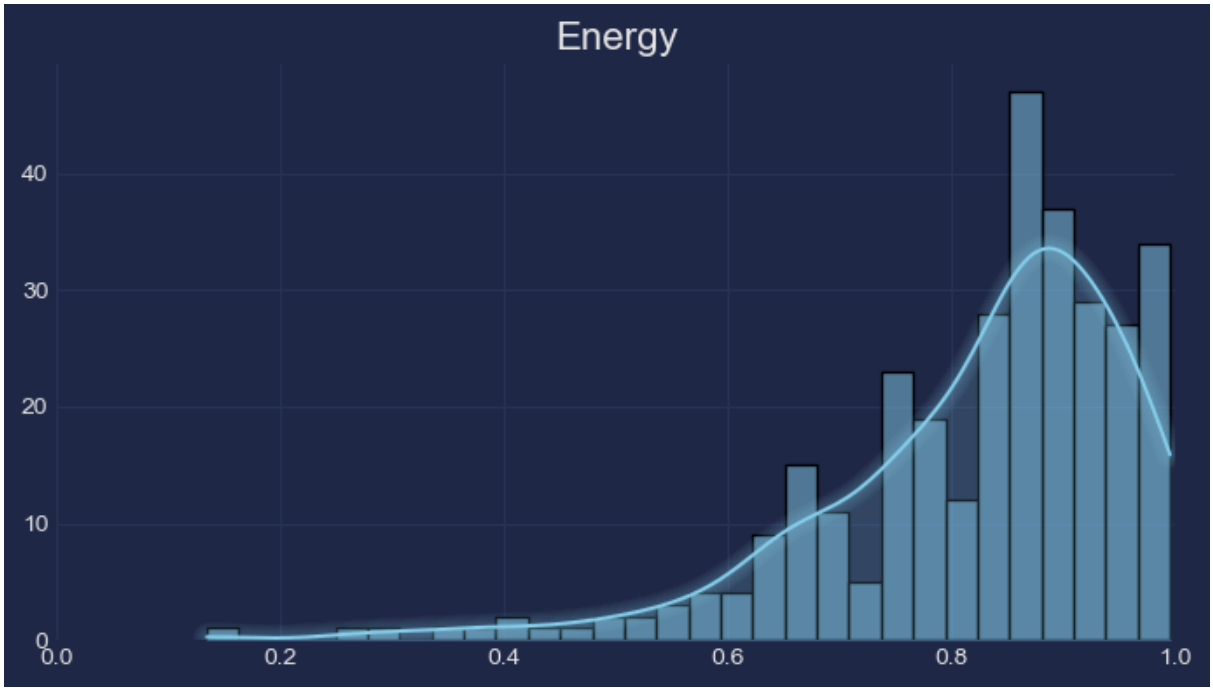


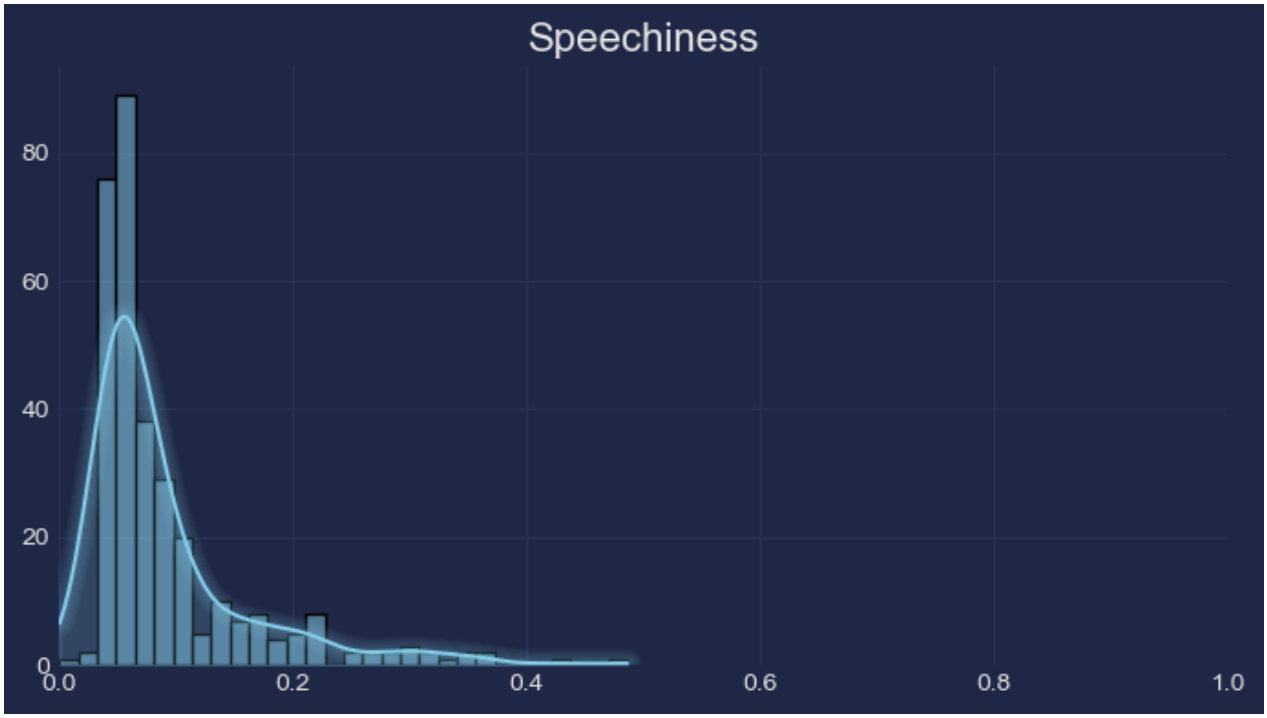

Number of Choruses per Song
Count of Songs
250
200
150
100
50
0
246
83
3
1
2
3
Number of Choruses

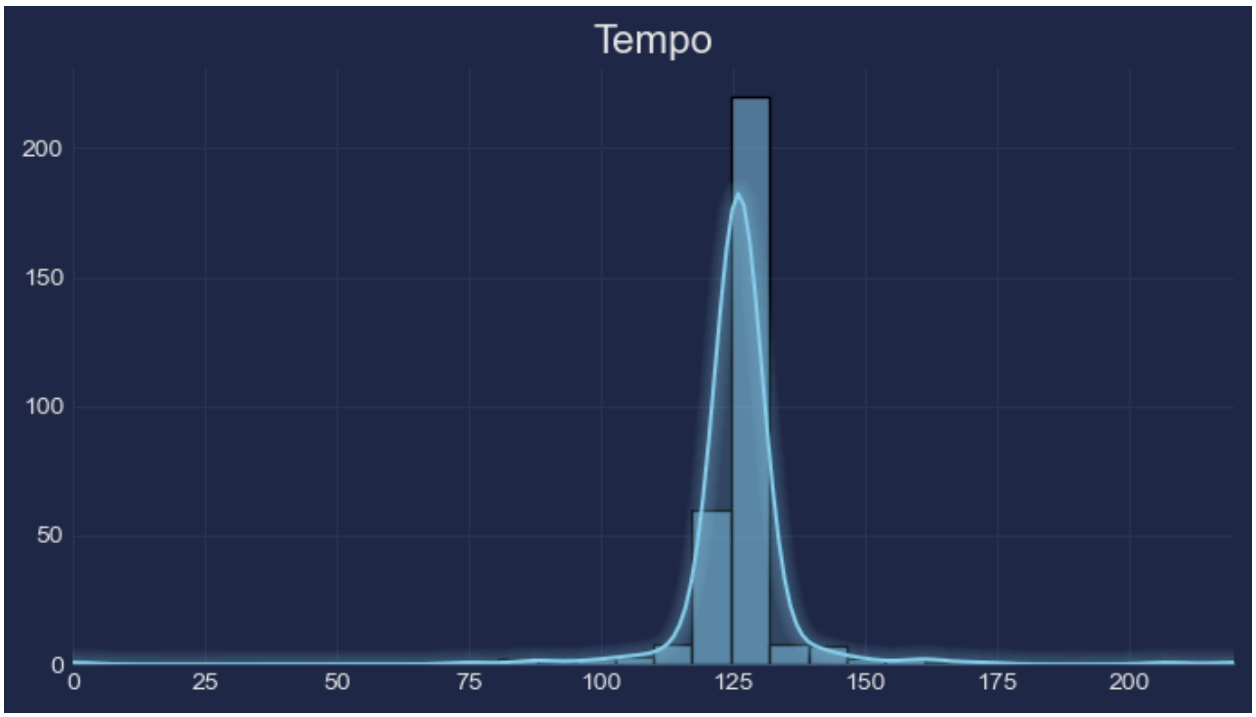

Distribution of Chorus Lengths
Frequency
200
150
100
50
0
10
20
30
40
50
60
70
80
90
Chorus Length (Seconds)

What do the songs in the dataset look like?

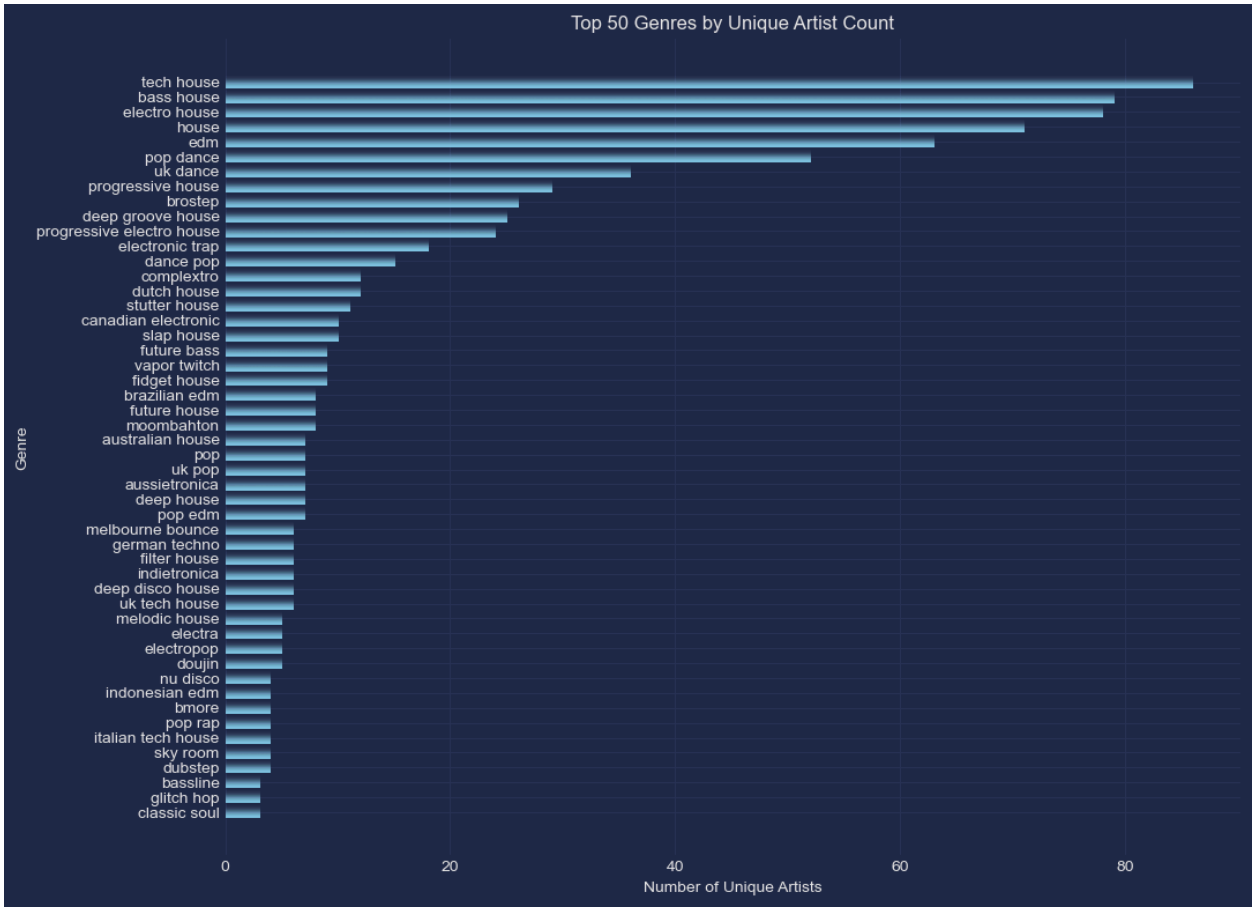





Speechiness
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0


Tempo
200
150
100
50
0
0
25
50
75
100
125
150
175
200

What genres of music are found in the dataset?



Label Validation

One of the aims of the EDA was to evaluate the quality and consistency of the manually labeled chorus segments, as well as the accuracy of the tempo estimation and beat-grid generation algorithms. Accurate tempo estimation and beat-grid generation are crucial, as these audio features will be used to partition the songs into meaningful units for modeling.

To assess the reliability of the manual chorus labeling, we rely on the hypothesis that a strong alignment between the chorus labels and an independently generated beat-grid and tempo estimation would indicate proximity to the "true" musical structure of the songs. This alignment-based approach to validation assumes that the manual labeling process accurately captures the essential features of the music, in accordance with the operational definition that choruses should start on the first downbeat of a bar and end after the last downbeat of a bar.

Methodology

1. Assessing Label Validity:
Choruses were manually labeled such that the start aligns with the first downbeat of the initial bar, while the end follows the last downbeat of the final bar. The chorus endpoints can be used to derive a beat grid, whose quality we can assess through its proximity to chorus start/end markers. Close alignment between labels and meter grid indices would suggest consistent labeling per the operational definition.

2. Tempo Estimation:
Accurate tempo estimation is crucial, as it will be used to generate the beat and bar grids, which can then be used to partition songs into meaningful units for modeling. We tested various tempo estimation methods, including Librosa's beat tracking approach, and compared the results to the tempo values provided by Spotify.

3. Beat and Meter Grid Generation:
Using the estimated tempos and an assumed 4/4 time signature, we generated beat and meter grids for each song, covering the full duration of the track. The beat grid represents the locations of individual beats, while the meter grid marks the beginning of each measure (bar).

4. Alignment Analysis:
We compared the manually labeled chorus start and end times to the generated beat and meter grids. This involved calculating the distance between the chorus boundaries and the nearest grid points, as well as checking if the chorus is fully contained within two adjacent grid indices. Songs with poor alignment between labels and grids are flagged for further investigation.

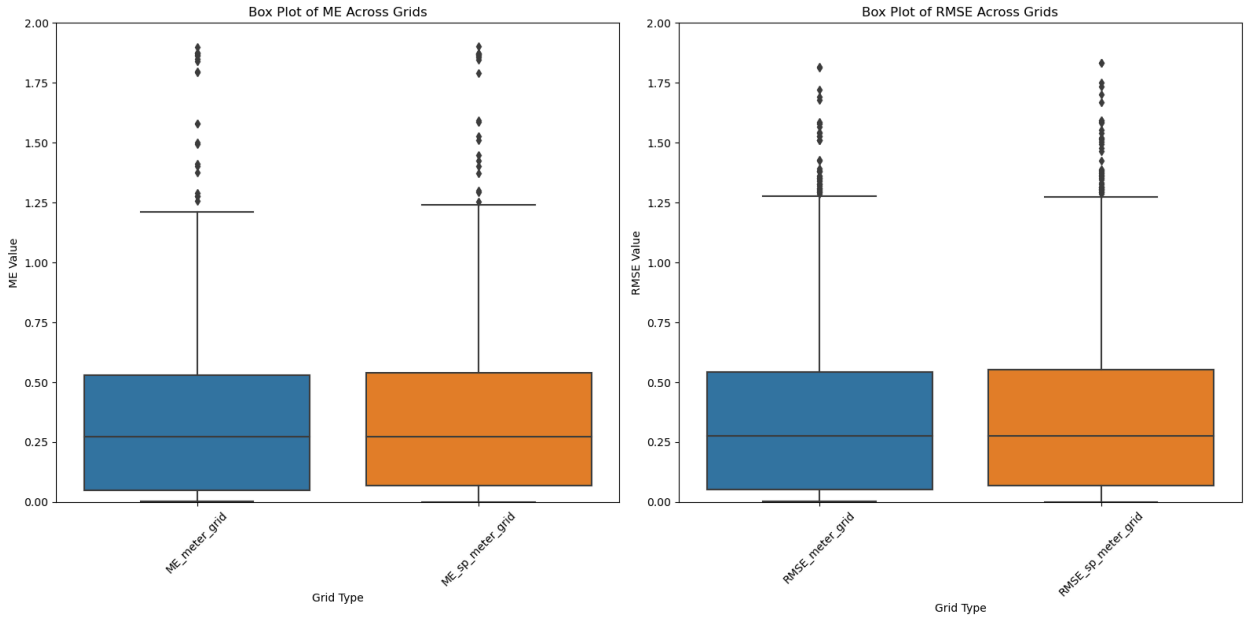
5. Outlier Analysis:
For songs where our tempo estimation differed significantly from Spotify's, we analyzed the alignment between labels and grids to determine whether to include or exclude those songs

from further analysis. Outliers may indicate issues with the tempo estimation or labeling process that require additional refinement.

Findings

The median ME and RMSE values for both grid types are around 0.27 seconds, indicating that **the chorus start and end times are, on average**, **within 0.27 seconds of the nearest meter.** This suggests a reasonably good alignment between the manual labels, labeling rules, and the generated grids.

The ME and RMSE values are very similar between beat grids generated using Spotify's tempo estimation and our own, suggesting that both tempo estimation methods are producing comparable results in terms of aligning with the manual chorus labels. There were no statistically significant differences between Librosa's and Spotify's tempo estimations.



Audio Feature Visualization

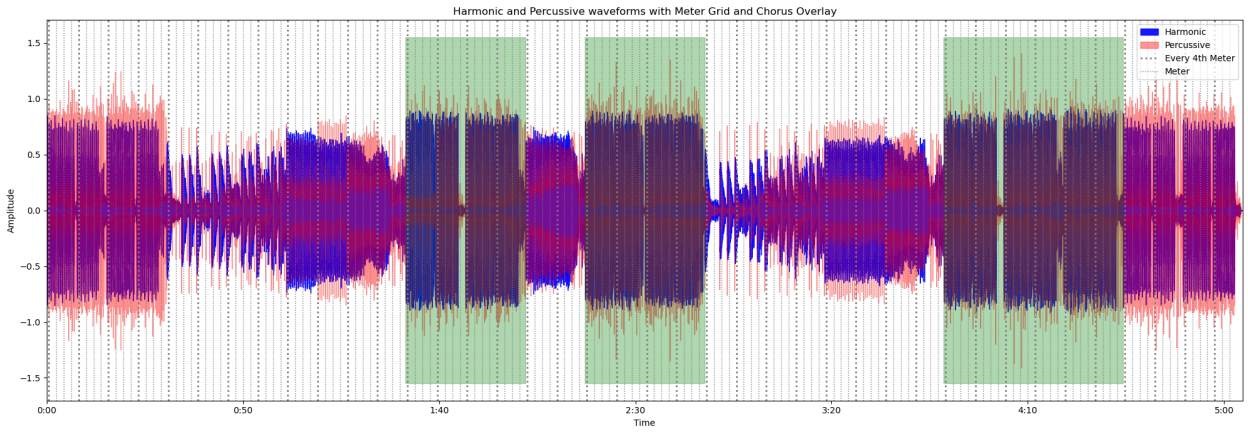
The last goal of the EDA was to visualize various audio features to gain insights into the characteristics of the music data and identify which features might be pertinent for the task of audio segmentation and chorus identification.

By visualizing features of a song like its spectrogram, tempogram, or chromagram, we can observe patterns or anomalies that might not be apparent from numerical representations alone. Visualization allows us to assess the potential usefulness of each feature for the task of chorus identification. Features that exhibit distinct patterns or characteristics within the chorus regions could be more informative for building accurate models.

In all of our visualizations, we overlay a meter grid, making it easier to spot segmentation and patterns that occur on a per-meter basis or over a multiple of 4 meters, which is common in many musical forms. In addition, we highlighted the chorus sections of each song in green blocks to see how certain audio features illuminate them.

Harmonic/Percussive Source Separation

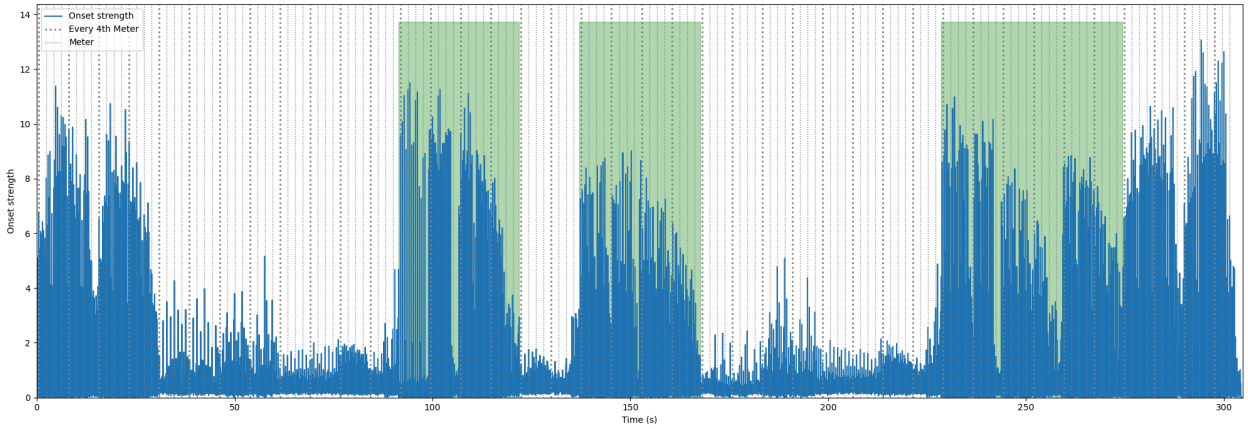
Harmonic-Percussive Source Separation (HPSS) is a technique used to **decompose an audio signal into its harmonic and percussive components**. Visualizing these components can provide valuable insights for chorus detection, as choruses often exhibit distinct characteristics in terms of their **melodic/tonal** *(i.e., harmonic)* and **rhythmic** *(i.e., percussive)* content. Certain audio features can be enhanced through extracting them from either the harmonic or percussive source.



Visualizing both the harmonic and percussive components together offers a comprehensive view of the interplay between the melodic and rhythmic elements. In the example above, we can observe how certain sections of the song are dominated by harmonic content while other sections are predominantly percussive. Additionally, the overlaid meter grid partitions the song into meaningful segments, with every fourth meter emphasized, as motifs and patterns in pop songs often unfold in multiples of four measures.

Onset Envelope

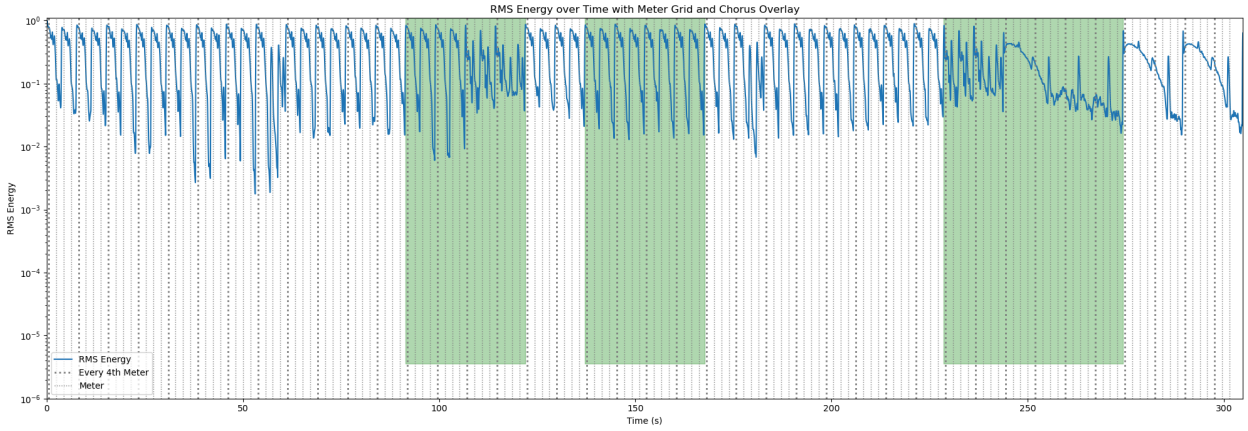
The onset envelope is another valuable audio feature that represents the **changes in the audio signal over time**. It can be particularly useful for detecting rhythmic patterns and identifying potential segment boundaries within a song.



In the onset envelope visualization, **the peaks correspond to the onset of new musical events**, such as the attack of a drum hit or the beginning of a new note or chord. By analyzing the onset envelope, we can observe the rhythmic structure and intensity variations throughout the song. Choruses often exhibit distinct rhythmic patterns or accentuations, which may be reflected in the onset envelope as a series of prominent peaks or a sustained high intensity region.

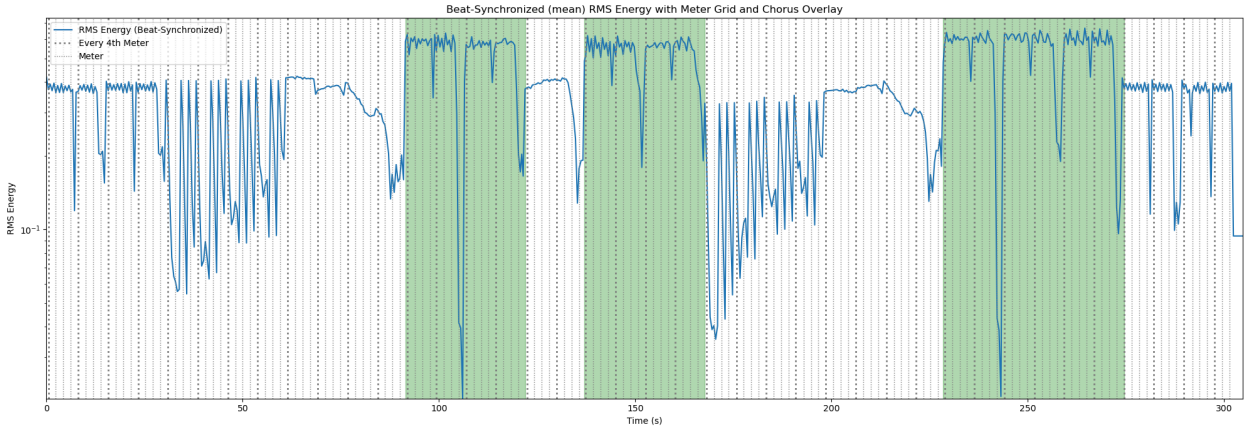
RMS Energy

RMS energy is a measure of the **overall energy** or **loudness of an audio signal over time.** It is calculated by taking the root mean square of the signal's amplitude values within a given window or frame. The RMS energy can be a useful feature for chorus detection because chorus sections often exhibit distinct energy patterns compared to other parts of a song.



Beat-Synced RMS Energy

We also visualize our audio features after synchronizing them with the beat information, averaging the feature values over each beat interval. By taking the average at each beat, you are essentially computing a running average or a low-pass filtered version of the RMS energy, where the averaging window size is determined by the beat period. This averaging process effectively smooths out the high-frequency variations and emphasizes the lower-frequency energy patterns that are more closely tied to the musical structure and chorus locations. Choruses, which are typically characterized by higher energy and intensity, become more apparent because the beat-synchronized RMS Energy captures the elevated energy levels sustained over the length of these sections.

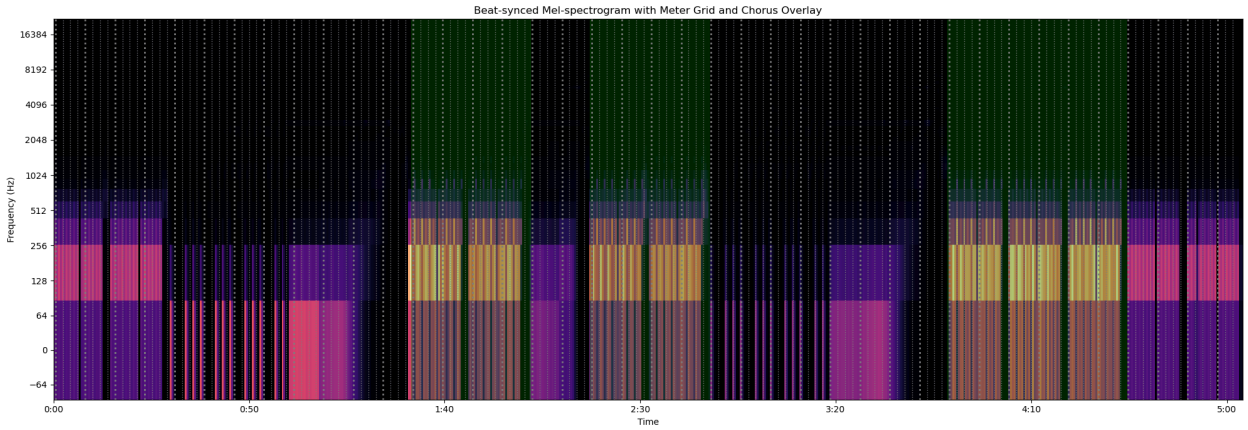


The clear pattern and segmentation that become apparent after beat-synchronization suggest that the energy dynamics of choruses in this song are more aligned with the song's beat structure than with instantaneous energy changes. **This observation underscores the importance of considering the temporal and structural context of music when analyzing audio features for tasks like chorus detection.**

Mel-Spectrogram

A mel spectrogram is a visual representation of the spectral content of an audio signal, where the frequency axis is warped to the mel scale, which is based on the human auditory perception of pitch. The mel scale is approximately linear at lower frequencies and logarithmic at higher frequencies, reflecting the non-linear sensitivity of the human ear to different frequency ranges.

Mel spectrograms can be valuable for chorus identification because of their effectiveness in capturing and visualizing harmonic/melodic patterns and the presence of human speech/vocals.



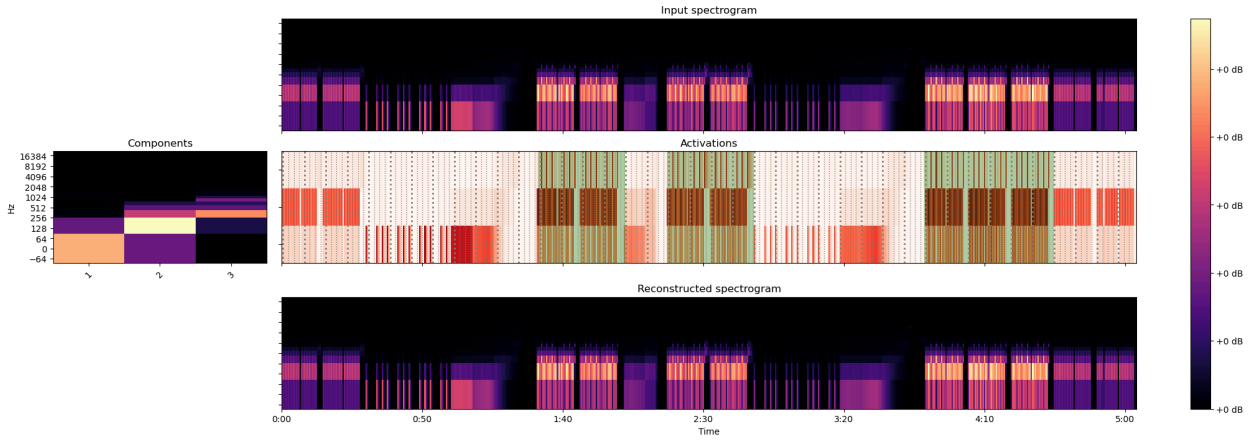
Decomposed Mel-Spectrogram Activations

Non-negative Matrix Factorization (NMF) is a dimensionality reduction technique that decomposes a non-negative matrix (in this case, the mel spectrogram) into the product of two lower-rank non-negative matrices. Specifically, given a mel spectrogram matrix ‘V’, NMF finds two non-negative matrices W and H such that:

$$\mathbf{V} \approx \mathbf{W} \times \mathbf{H}$$

Where:
V is the mel spectrogram matrix
W is the basis matrix, representing the basis vectors or spectral patterns
H is the activation matrix, representing the activations or temporal weights of the basis vectors

The key idea behind using NMF for chorus identification is to leverage the activation matrix **H** as a feature representation for the audio signal. The activation matrix captures the temporal evolution and contribution of the spectral patterns (basis vectors) in the mel spectrogram, which can be useful for identifying repetitive or recurring patterns associated with chorus sections.

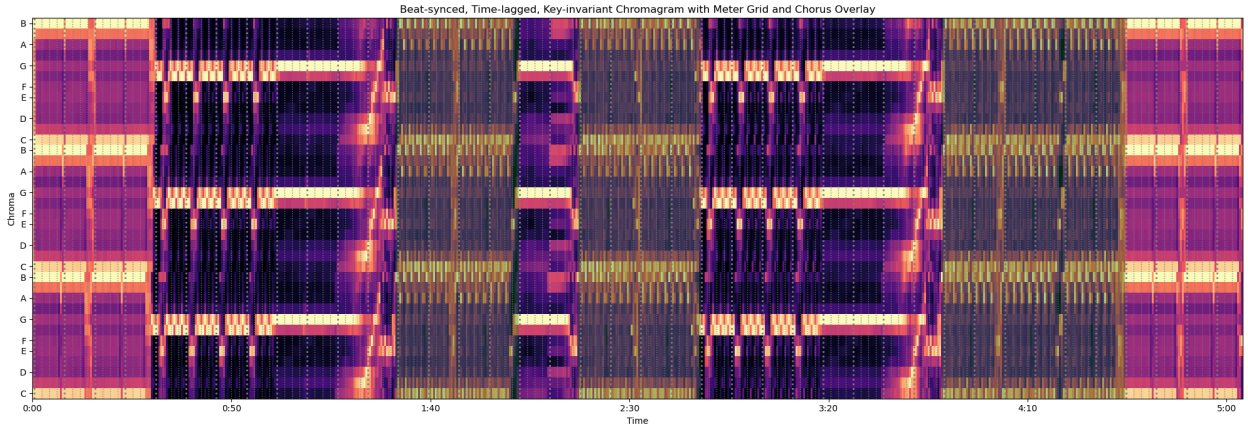


NMF decomposes the high-dimensional mel spectrogram into a lower-rank representation, effectively reducing the dimensionality of the feature space. In our example, we decompose a mel spectrogram with 128 mel features into just 3 activations. This is **computationally more efficient** and can potentially improve the performance of machine learning models by mitigating the curse of dimensionality.

Chromagram

A chromagram is a time-frequency representation of an audio signal that captures the distribution of energy across the 12 pitch classes (semitones) in the chromatic scale. We visualize using beat-synced, key-invariant chromagrams for the benefits of beat-synchronization mentioned earlier, and because key-invariant chromagrams are normalized to a specific key, making them useful for identifying chorus sections that may occur in different keys, as their harmonic patterns will still be recognizable in the key-invariant chromagram.

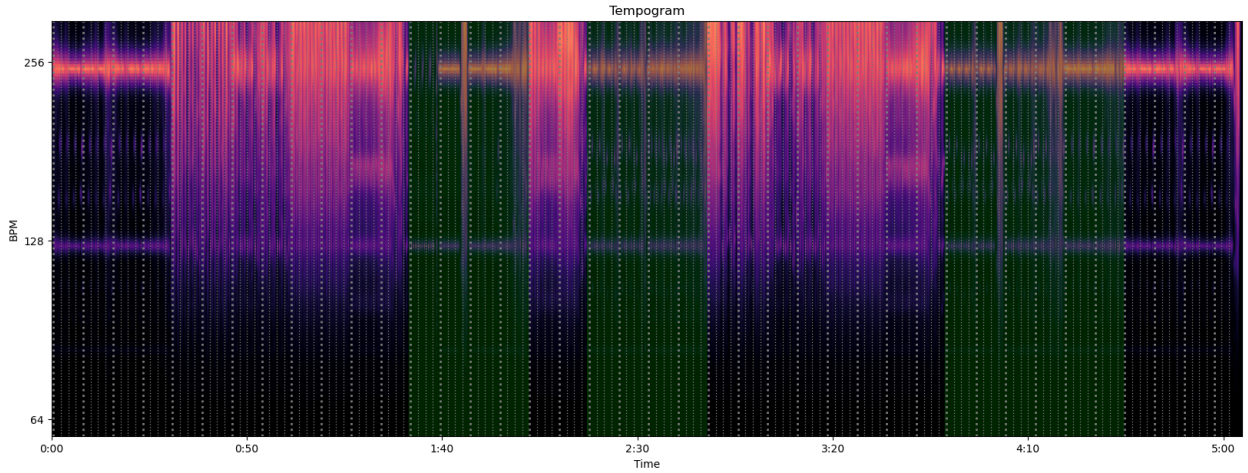
Time-lagged, Stacked Chromagram



By time-lagging and stacking multiple instances of the chromagram, we create a visual representation that highlights repeating patterns more prominently. This technique can make it easier to visually identify chorus sections, as their harmonic patterns will appear as vertically aligned structures in the stacked chromagram.

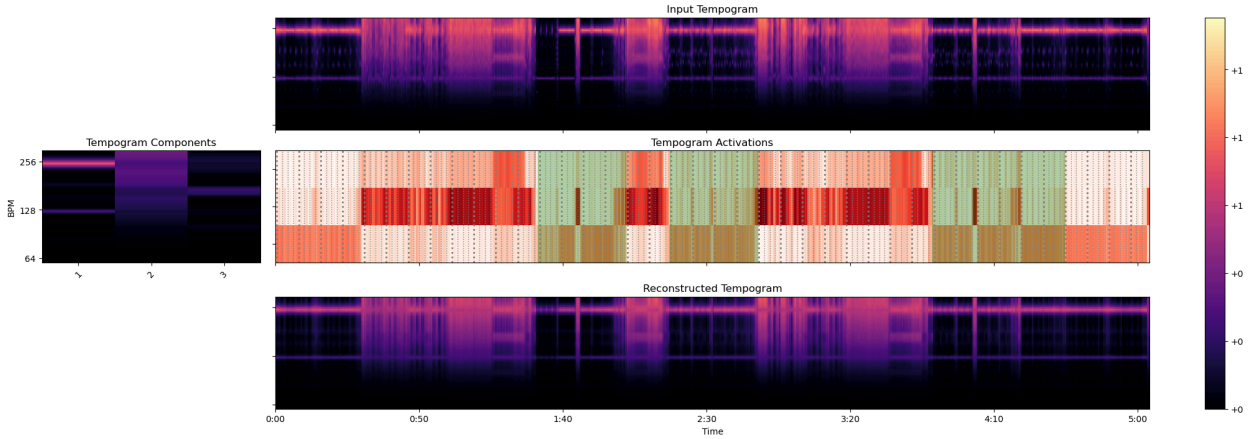
Tempogram

A tempogram is a time-frequency representation that captures the rhythmic and tempo information in an audio signal. It essentially represents the energy or strength of different tempi (beats per minute) over time. Tempograms can be useful for chorus identification because chorus sections often exhibit distinct rhythmic patterns and tempi compared to other sections of a song.



Decomposed Tempogram

Additionally, as mentioned earlier, applying Non-negative Matrix Factorization (NMF) to decompose the tempogram reduces the dimensionality of the feature. In our example, we reduce 384 dimensions (tempi values) into 3 components.



Modeling

For the task of identifying chorus sections in songs, we chose a Convolutional Recurrent Neural Network (CRNN) architecture. This type of model is well-suited for analyzing the complex and hierarchical structure of audio data because it combines the strengths of two powerful neural network components: convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

Feature Selection

Before building the model, we carefully selected and preprocessed input features based on exploratory data analysis and visualization of various audio characteristics. The features we extracted and used include:

1. **RMS Energy**: The root mean square energy of the audio signal.
2. **Mel-Spectrogram Activations**: The mel-spectrogram computed and decomposed into 4 components using Non-negative Matrix Factorization (NMF).
3. **Key-Invariant Chromagram and Chromagram Activations**: The chromagram computed from the harmonic audio component, made key-invariant, and decomposed into 3 components using NMF.
4. Tempogram Activations: The tempogram computed from the onset envelope and decomposed into 3 components using NMF.

5. **Mel-Frequency Cepstral Coefficients (MFCCs)**: Widely used features in audio processing, computed directly from the audio signal.

These features were standardized, weighted, and concatenated into a single combined 36-dimensional representation per audio frame.

Feature Partitioning

We **segmented the audio data into musical meters (bars)** to align the feature representations with the underlying musical structure. This segmentation can help the model better capture relevant patterns, allow separate modeling of temporal dependencies within each measure, and make processing more efficient.

Hierarchical Positional Encoding

Additionally, we introduced a novel hierarchical positional encoding scheme to enrich the model's input with temporal and structural context. The hierarchical positional encoding operates at two levels:

1. **Meter-level Encoding:** At the macro level, this encoding embeds the position of each musical meter within a song, reflecting the overarching structure of verses, choruses, bridges, and other sections. By encoding this sequential arrangement, the model gains insights into the progression and dynamics of a composition, which aligns with the human intuition that chorus detection requires an understanding of how different sections are organized.

2. **Frame-level Encoding:** At a more granular level, the encoding embeds the position of each frame within its respective meter. This fine-grained temporal information allows the model to discern subtle rhythmic and melodic variations within each measure, potentially enabling it to learn patterns that distinguish choruses from other sections.

The hierarchical nature of these encodings mirrors the inherent hierarchical structure of music itself, spanning from the broad arrangement of sections down to the timing of individual notes. While the predetermined kernel size may limit the model's ability to learn certain spatial-temporal patterns, the hierarchical positional encoding and the **inductive bias introduced by the meter-level segmentation** provide valuable contextual information that the Convolutional Recurrent Neural Network (CRNN) can leverage.

Model Selection

Music is inherently rich with patterns that span different timescales - from the fine-grained rhythms and melodies within each measure, to the broader arrangement of sections like verses

and choruses. To effectively capture these multi-level patterns for accurate chorus detection, we needed a model architecture that could analyze both local and global contexts.

The Convolutional Recurrent Neural Network (CRNN) architecture emerged as an ideal choice for this task. It combines the strengths of two powerful components: convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

CNNs excel at extracting local patterns from audio data, such as specific rhythms or timbral qualities within short time frames. Meanwhile, RNNs are adept at modeling sequences and capturing long-range dependencies, making them well-suited for understanding the progression and relationships across extended musical segments.

By integrating these complementary capabilities, the CRNN can simultaneously learn the intricate rhythmic and melodic nuances that distinguish choruses, while also grasping the overarching structure and flow of a song's sections. This multi-level analysis aligns with how humans intuitively perceive and identify chorus sections.

Model Architecture

The CRNN model consists of two main components: a CNN component and an RNN component. The CNN component processes the input features at the frame level, using three 1D convolutional layers with ReLU activation and max-pooling layers. This component is responsible for extracting local patterns, such as specific frequencies, rhythmic patterns, or timbral characteristics, which are important for distinguishing different sections of a song.

The output of the CNN component is then fed into the RNN component, which consists of a Bidirectional Long Short-Term Memory (LSTM) layer. This component is designed to model the temporal dependencies and long-range patterns across musical measures, enabling the model to understand the broader context and progression of a song.

To handle variable-length input sequences and efficiently process the data, we incorporated a masking layer and wrapped the CNN component in a TimeDistributed layer. This approach allows the model to process each measure independently while still capturing the overall sequential structure of the input.

The final output of the model is produced by a TimeDistributed Dense layer with a sigmoid activation, which **generates predictions for each meter**, indicating the likelihood of it belonging to a chorus section.

Model Training

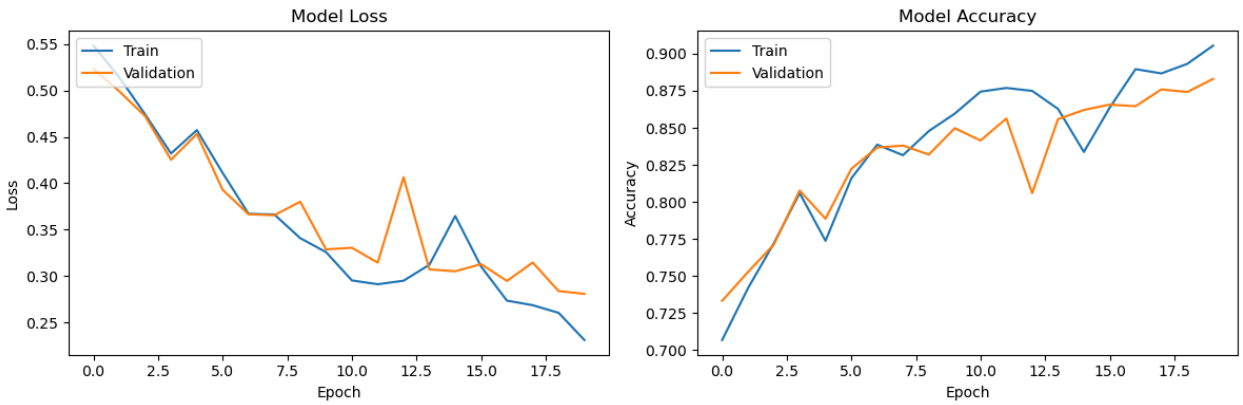
We defined a custom binary cross-entropy loss function and a custom accuracy metric to handle the presence of -1 labels, which are used for padding and should be ignored during loss calculation and accuracy evaluation.

The training process implemented a set of callbacks to ensure optimal performance and prevent overfitting:

- ModelCheckpoint: Saves the best model based on minimizing the validation loss.
- EarlyStopping: Stops training if the validation loss doesn't improve for a specified number of epochs.
- ReduceLROnPlateau: Reduces the learning rate if the validation loss doesn't improve for a specified number of epochs.

The model is trained using the fit method, passing the training dataset, number of epochs (20), validation dataset, and the defined callbacks. This approach ensures that the model is continuously monitored and optimized during the training process.

Below are plots of the model loss and accuracy over the 20 training epochs.



Model Evaluation

After training our Convolutional Recurrent Neural Network (CRNN), we tested its ability to accurately identify chorus sections in songs. To do this, we used a separate test set of 50 songs that the model had never seen before. We evaluated the model's performance on this test set using several key metrics. We summarize the results in the table below:

Metric	**Value**
Loss	0.2343
Accuracy	0.8997
Precision	0.8843
Recall	0.8689
F1 Score	0.8766

Loss
This measures how often the model made incorrect predictions. In our case, the loss value of **0.2343** tells us how often, on average, the model is making incorrect predictions about whether a particular section of a song is a chorus or not. This loss value is calculated across the entire set of test songs that the model has never seen before. A loss of 0.2343 is relatively low, which suggests that the model is making accurate predictions most of the time, with a relatively small number of mistakes or incorrect identifications of chorus sections.

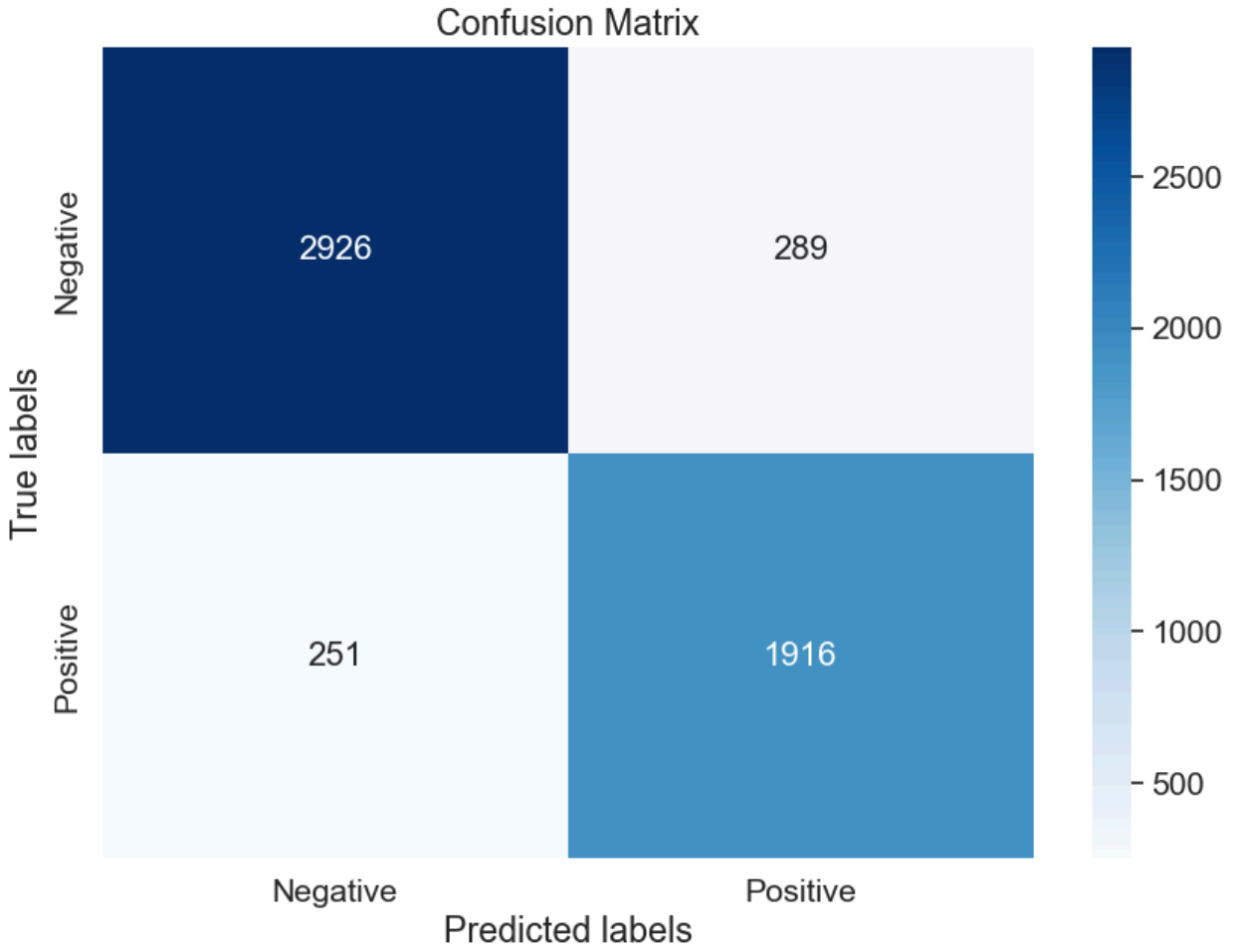
Accuracy
This simply tells us what percentage of predictions were completely correct. Our model achieved an accuracy of **89.97%,** which means it correctly identified whether a section was a chorus or not in nearly 9 out of 10 cases.

Precision
This metric measures how reliable the model's predictions of "this is a chorus" were. Our precision of 88.43% indicates that **when the model predicted a section to be a chorus, it was correct 88.43% of the time.**

Recall
This metric tells us the proportion of true choruses that the model was able to correctly detect and label as such. Our model achieved a recall of 86.90%, which means that **out of all the actual chorus sections in the test set, the model correctly identified 86.90% of them as choruses.** This high recall value indicates that the model is effectively capturing the vast majority of true choruses, minimizing the number of instances where it misses or fails to detect a chorus section.

F1 Score

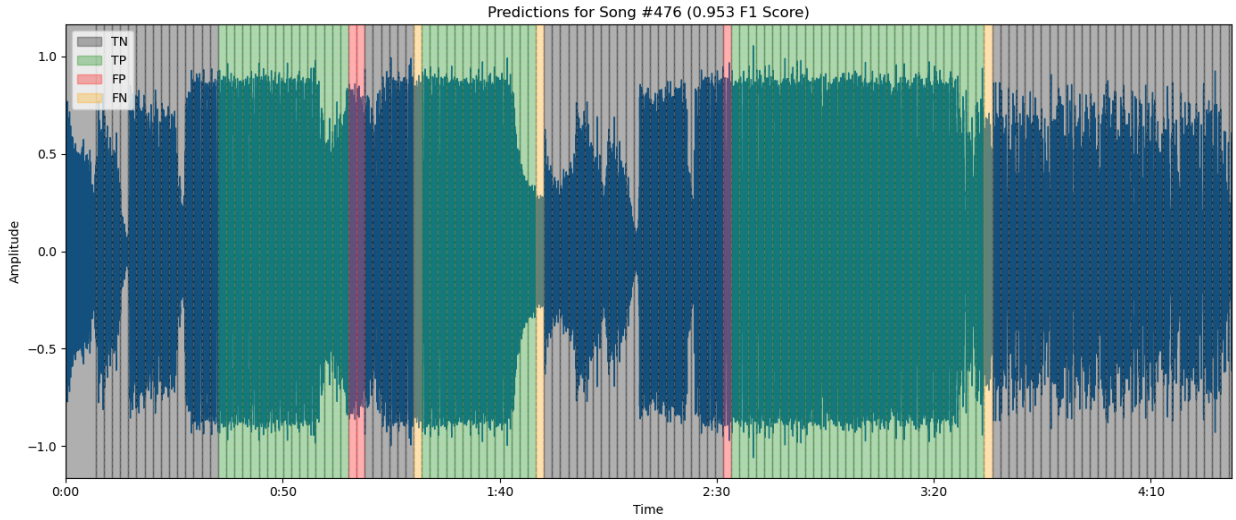
This is an overall performance metric that combines accuracy and precision. Our model's F1 score of 0.876 (on a scale of 0 to 1, where 1 is perfect) suggests it performed well in accurately identifying choruses while minimizing incorrect predictions.

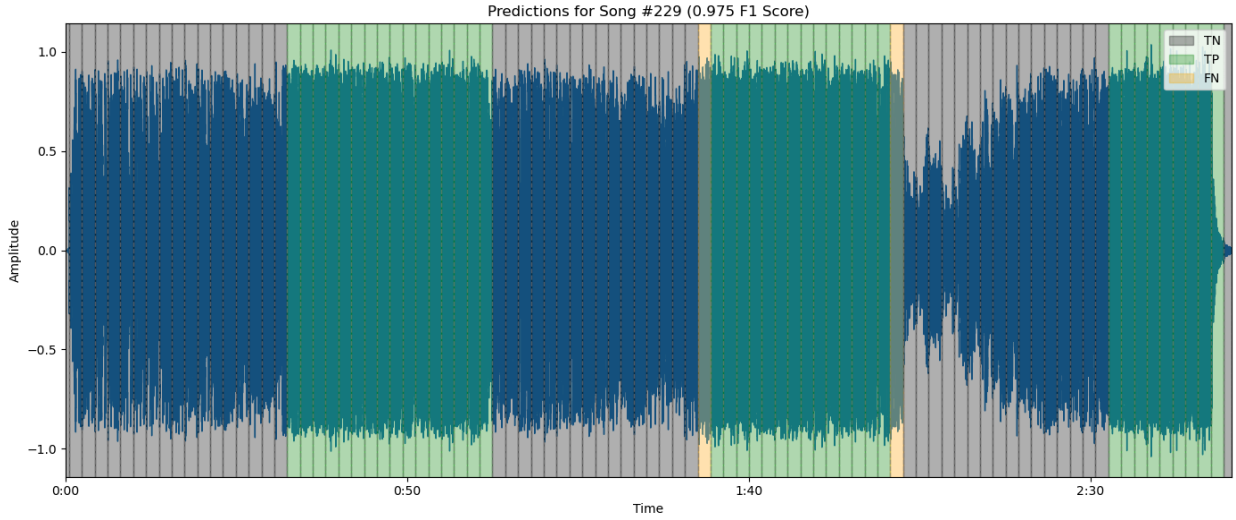


While no machine learning model is perfect, these evaluation results demonstrate that our CRNN model is highly effective at recognizing chorus sections in songs. It gets the vast majority of predictions right, with a low error rate and reliable identification of true choruses.

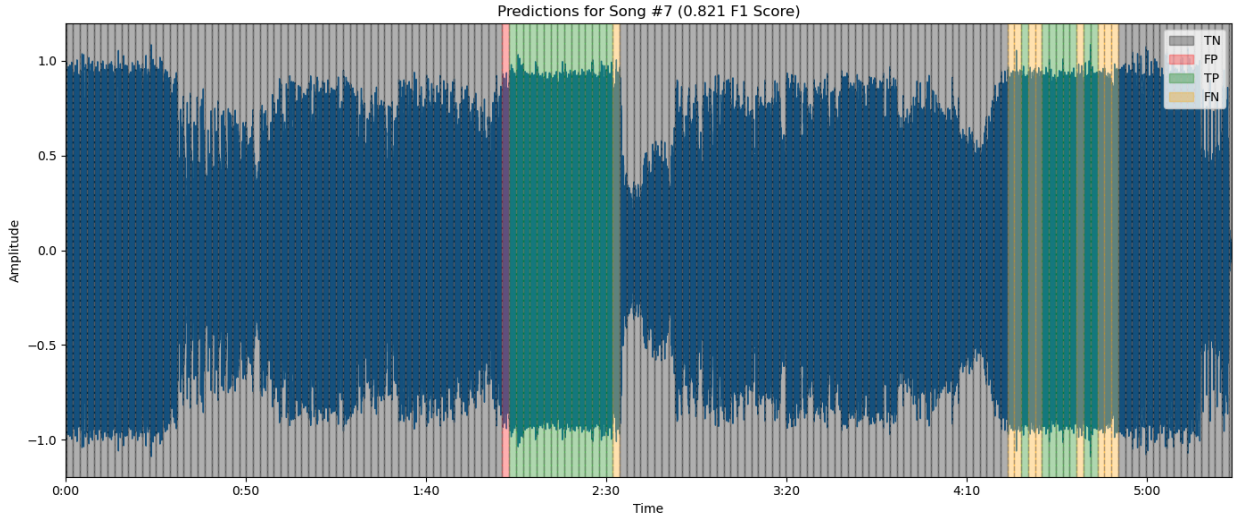
Prediction Visualizations

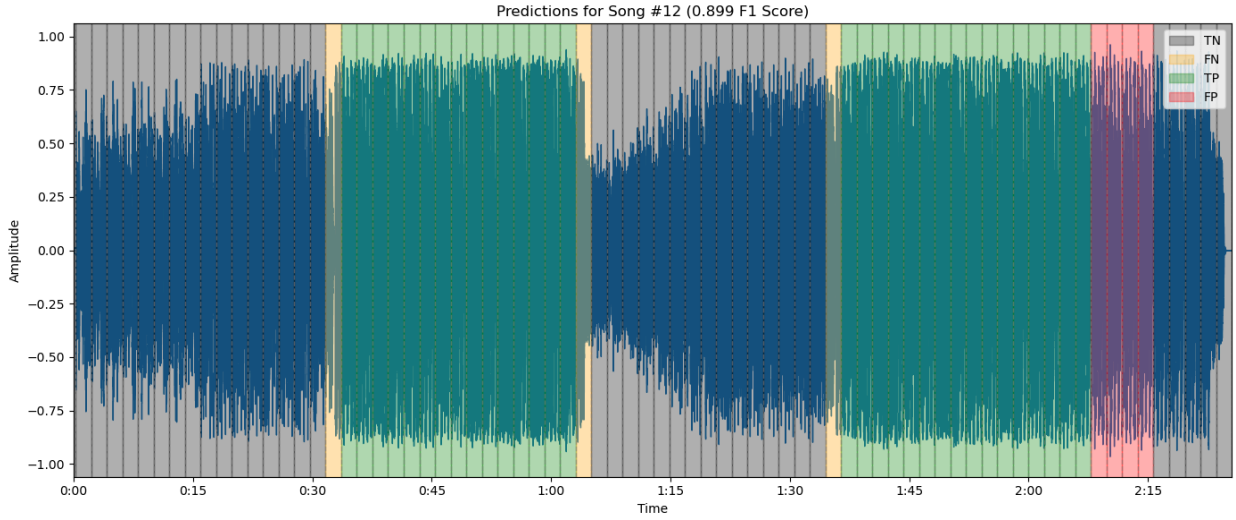
We also conducted a qualitative analysis by visually inspecting the model's predictions on the test songs. This allowed us to identify specific scenarios where the model excelled or struggled, providing insights for potential future improvements. Below are a selection of test predictions:

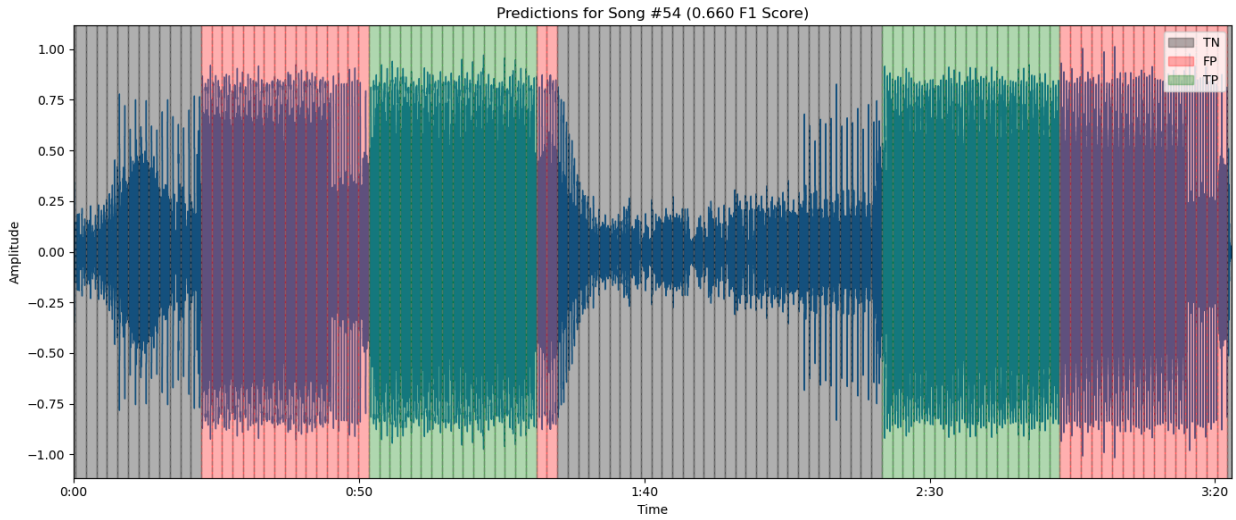


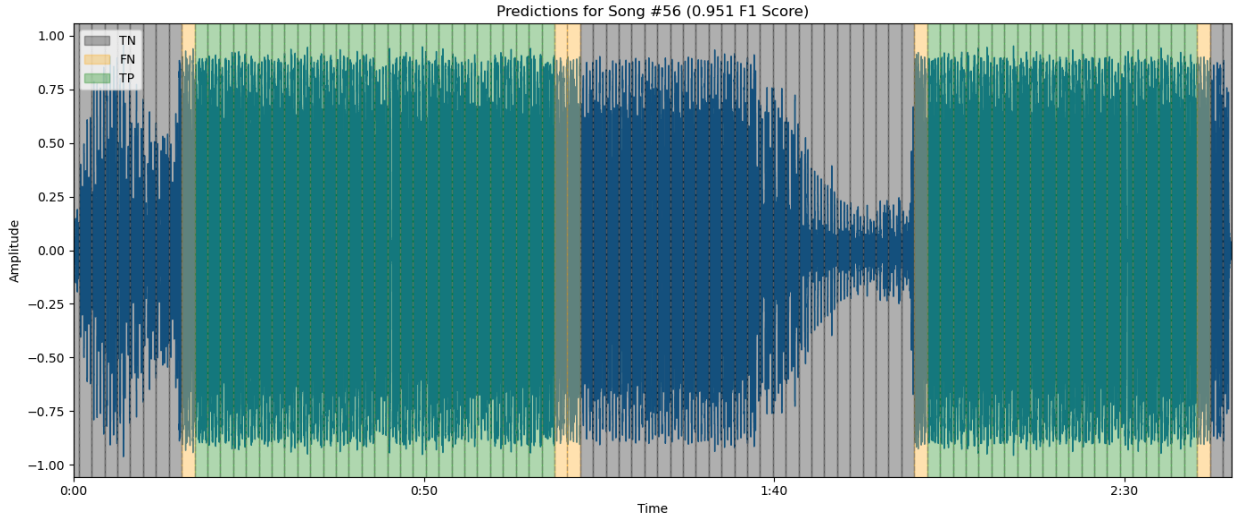


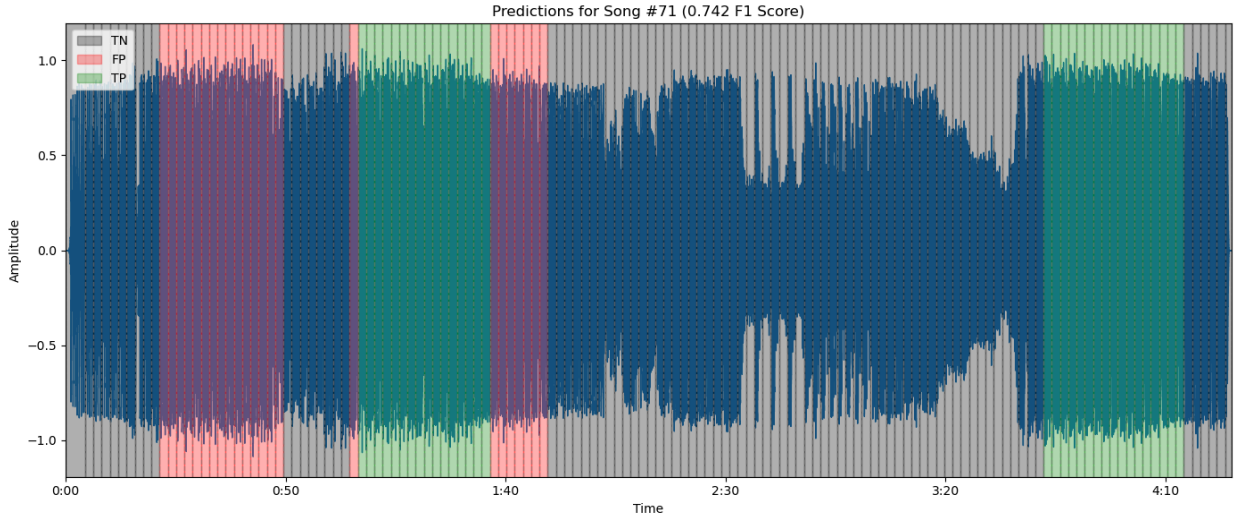

Predictions for Song #2 (0.900 F1 Score)
TN
TP
FN
FP
Amplitude
1.0
0.5
0.0
−0.5
−1.0
0:00
0:50
1:40
2:30
3:20
4:10
Time

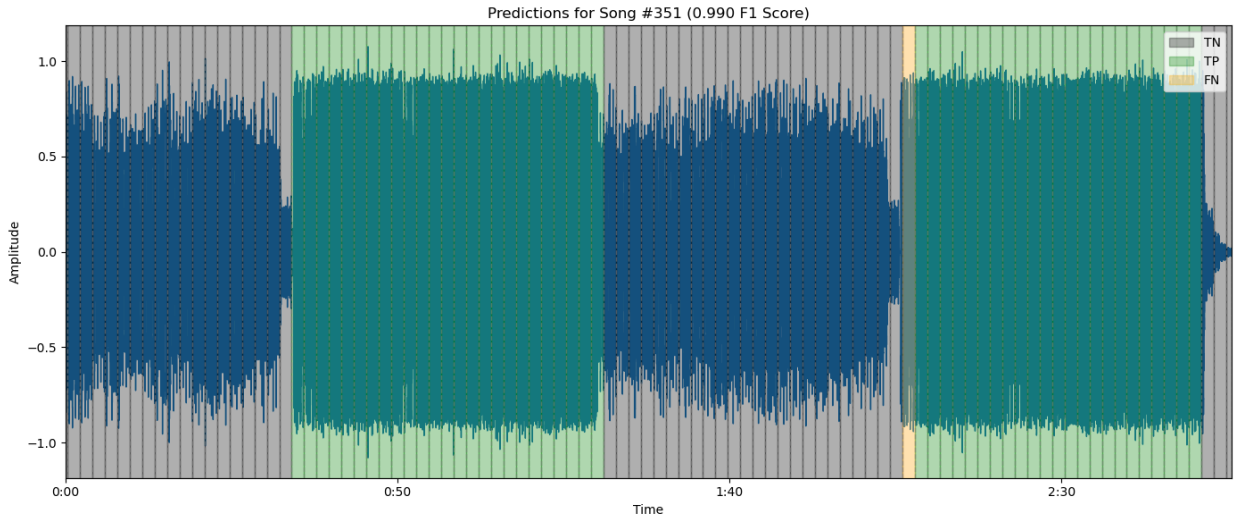

Predictions for Song #7 (0.821 F1 Score)
TN
FP
TP
FN
Amplitude
1.0
0.5
0.0
−0.5
−1.0
0:00
0:50
1:40
2:30
3:20
4:10
5:00
Time

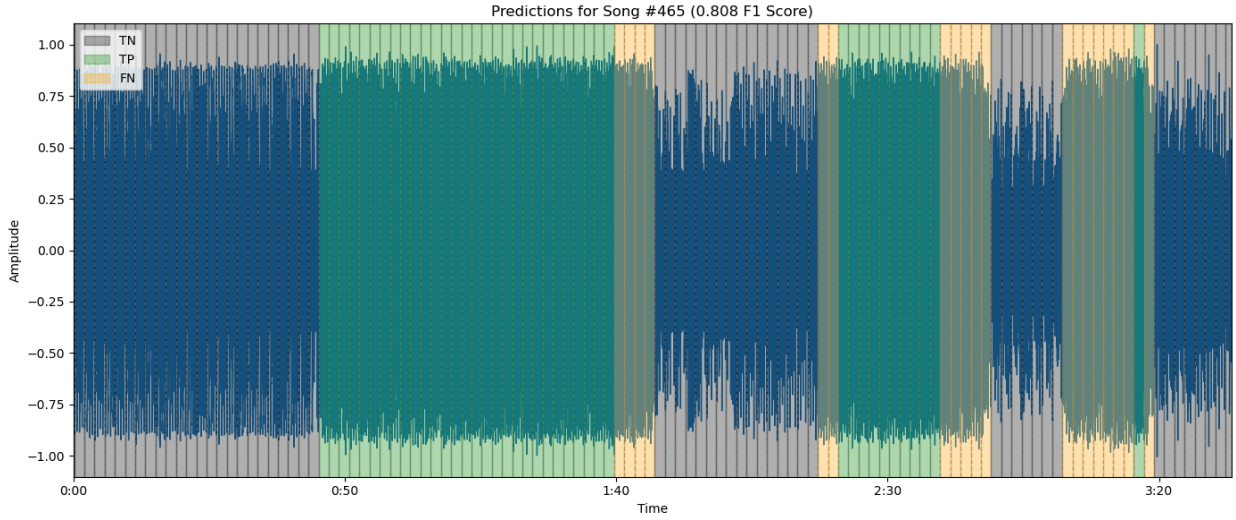

Predictions for Song #12 (0.899 F1 Score)
TN
FN
TP
FP
Amplitude
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0:00
0:15
0:30
0:45
1:00
1:15
1:30
1:45
2:00
2:15
Time

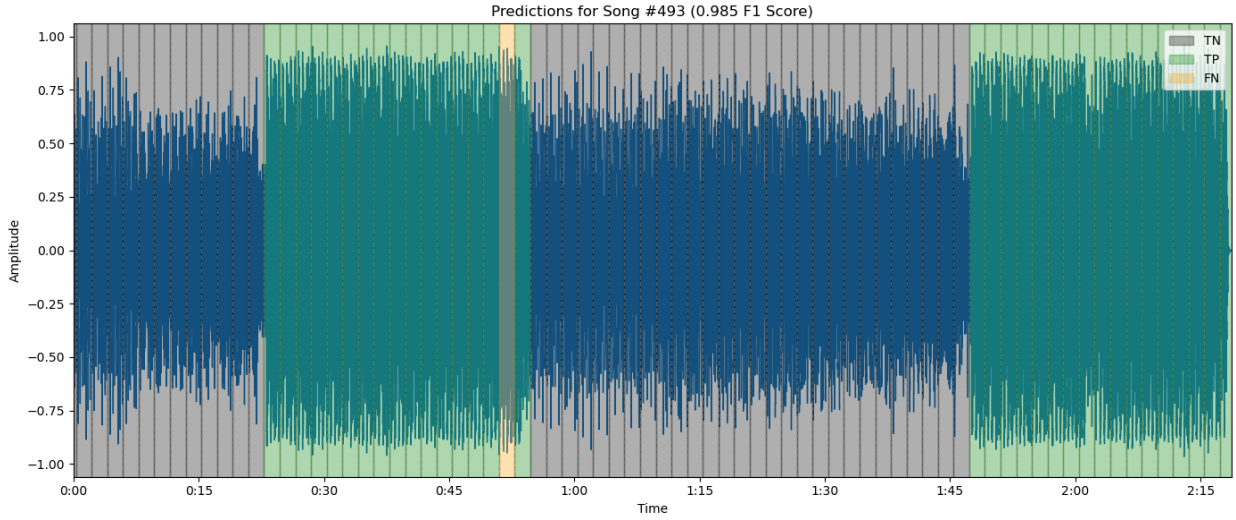

Predictions for Song #54 (0.660 F1 Score)
TN
FP
TP
Amplitude
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0:00
0:50
1:40
2:30
3:20
Time


Predictions for Song #56 (0.951 F1 Score)
TN
FN
TP
Amplitude
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0:00
0:50
1:40
2:30
Time


Predictions for Song #71 (0.742 F1 Score)
TN
FP
TP
Amplitude
1.0
0.5
0.0
−0.5
−1.0
0:00
0:50
1:40
2:30
3:20
4:10
Time


Predictions for Song #351 (0.990 F1 Score)
TN
TP
FN
Amplitude
1.0
0.5
0.0
−0.5
−1.0
0:00
0:50
1:40
2:30
Time


Predictions for Song #465 (0.808 F1 Score)
TN
TP
FN
Amplitude
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0:00
0:50
1:40
2:30
3:20
Time


Predictions for Song #493 (0.985 F1 Score)
TN
TP
FN
Amplitude
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0:00
0:15
0:30
0:45
1:00
1:15
1:30
1:45
2:00
2:15
Time

Conclusion

In this project, we successfully developed an accurate and efficient automated chorus detection model using a Convolutional Recurrent Neural Network (CRNN) architecture. Through rigorous exploratory data analysis, careful feature engineering, and novel architectural design choices, our model achieved impressive performance metrics on a held-out test set, with an F1 score of 0.876.

Key Innovations

The key innovations that contributed to the model's success include:

1. Extracting and combining diverse audio features like RMS energy, mel-spectrogram activations, chromagram activations, tempogram activations, and MFCCs, which captured complementary aspects of the audio signal.

2. Segmenting the audio data into musical meters (bars) to align the feature representations with the underlying musical structure, allowing the model to learn patterns within individual measures and across the broader song arrangement.

3. Introducing a novel hierarchical positional encoding scheme that enriched the model's input with both fine-grained temporal information (frame-level) and coarse-grained structural context (meter-level), mimicking the inherent hierarchical nature of music itself.

4. Employing a Convolutional Recurrent Neural Network (CRNN) architecture that effectively integrated the strengths of CNNs for local pattern extraction and RNNs for modeling long-range dependencies, enabling multi-level analysis of the rich patterns present in audio data.

Through rigorous evaluation on a separate test set, our model demonstrated its effectiveness in accurately identifying chorus sections, with a low error rate, reliable precision, and high recall. Qualitative analysis of the model's predictions on sample songs further validated its performance and provided insights into potential areas for future improvement.

Key Takeaways

1. Combining diverse audio features and leveraging their complementary strengths can significantly enhance the model's ability to capture complex patterns in music data.

2. Aligning feature representations with the underlying musical structure, through techniques like meter-level segmentation, can provide valuable contextual information and improve model performance.

3. Incorporating hierarchical positional encodings can help mirror the inherent hierarchical structure of music, enabling the model to learn patterns at different timescales more effectively.

4. The CRNN architecture, with its combination of CNNs and RNNs, is well-suited for tasks involving audio data, as it can analyze both local and global patterns, critical for understanding the rich and multi-level structure of music.

5. Rigorous evaluation, both quantitative and qualitative, is essential for assessing model performance, identifying strengths and weaknesses, and informing future improvements.

Overall, this project demonstrates the potential of deep learning techniques and carefully engineered architectures to tackle complex audio analysis tasks, paving the way for enhanced music experiences and a deeper understanding of the intricate patterns that define our favorite songs.