Predict The Genre of A Song From It's Lyrics

Siddanta K.C, Aayush Bhetuwal
Kiran Vutukuri, Jianlin Lin
Katz School of Science and Health
MS in Artificial Intelligence
Yeshiva University

Abstract

In this report, we explore the effectiveness of an ensemble of machine learning models for various Natural Language Processing (NLP) tasks. The models employed include transformers, Naive Bayes, decision trees, logistic regression, and Support Vector Machines (SVM). Leveraging the unique strengths of each model, we tackle diverse challenges in NLP, such as sentiment analysis, text classification, and language generation. The state-of-the-art capabilities of transformers in capturing contextual information make them suitable for language generation and sentiment analysis, while Naive Bayes and decision trees offer simplicity and interpretability for specific text classification tasks. Logistic regression serves as a baseline with its straightforward approach and interpretable probabilistic framework, and SVM excels in handling high-dimensional data and complex decision boundaries. Through comprehensive analysis and comparison, this ensemble approach provides valuable insights into the models' performance, guiding their applicability across different NLP tasks and contributing to the advancement of the field. In addition to that, we are exploring the pervasive theme of alcohol references in song lyrics spanning over three decades. Our project meticulously combs through song lyrics from 1991 to 2023, dissecting the words, phrases, and contexts that artists employ to weave alcohol-related narratives into their music.

1. Introduction

In today's data-driven landscape, the allure of music remains unwavering and ever-captivating. As a powerful medium of expression, music possesses the unique ability to encapsulate a wide spectrum of emotions, experiences, and cultural narratives. It's within this intriguing intersection of sound and sentiment that our journey embarks—an expedition aimed at unraveling the intricate and fascinating correlation between song lyrics and the diverse tapestry of musical genres[1].

Driven by an unwavering quest for knowledge, we employ cutting-edge statistical and machine-learning techniques to decipher the hidden code embedded within lyrical compositions. Our mission is nothing short of awe-inspiring: to predict song genres with an unparalleled precision, relying solely on the narrative threads interwoven into the lyrical fabric. This voyage encompasses not only the mechanics of algorithms but also the soulful resonance of words and melodies.

Our pursuit orbits around the creation of robust classifiers, meticulously honed to unlock the key lyrical motifs resonating across a myriad of contexts. This journey, however, transcends mere categorization. It unveils the profound interplay of language, emotion, and artistic ingenuity—a narrative that stretches beyond the realm of genre prediction. At the heart of our approach lies the intricate marriage of Natural Language Processing (NLP)[2] and machine learning, forging a harmonious symphony where words metamorphose into meaningful insights and patterns.

As you navigate these pages, you will encounter a collection of distilled insights, each a microcosm of our intensive study. These condensed fragments encapsulate intricate concepts, showcasing our ability to communicate complexity with elegant precision[3]. This process is a testament to our commitment to lucidity—our dedication to not just scratching the surface but excavating the depths of our collaborative exploration.

Far more than a mere report, this document embodies our shared voyage—a vessel brimming with the collective wisdom of shared insights, late-night contemplations, and united challenges. It stands as an enduring testament to our unwavering commitment to deciphering the intricate relationship between lyrical narratives and the ever-evolving landscape of musical genres[4]. With these words, we invite you to join us on this expedition, a harmonious blend of artistry and technology. This report marks not just an endpoint but a remarkable beginning—an exploration of the limitless possibilities woven at the intersection of music and data-driven revelation.

2. Related Work

Our expedition into the intricate realm of predicting song genres through lyrical analysis is firmly grounded in a rich tapestry of comprehensive research, delving deep into the multifaceted interplay between music and data-driven methodologies. Within this vibrant landscape, a multitude of studies have diligently pursued the unraveling of inherent patterns and subtle intricacies interwoven within musical compositions, skillfully employing advanced techniques to unearth concealed insights. As we embark on our journey, we find inspiration in these trailblazing efforts, charting a distinct trajectory that pivots around the lyrical dimension of music.

The application of machine learning algorithms to diverse realms of music analysis has witnessed remarkable advancements. Research endeavors have encompassed a spectrum of tasks, encompassing genre classification, mood detection, artist identification, and recommendation systems. Our exploration stands as an organic evolution of this trajectory, honing in on lyrics as a potent wellspring of information for genre prediction. By harnessing state-of-the-art statistical and machine learning techniques, our aim is to illuminate the intricate nexus between lyrical content and the diverse tapestry of musical genres.

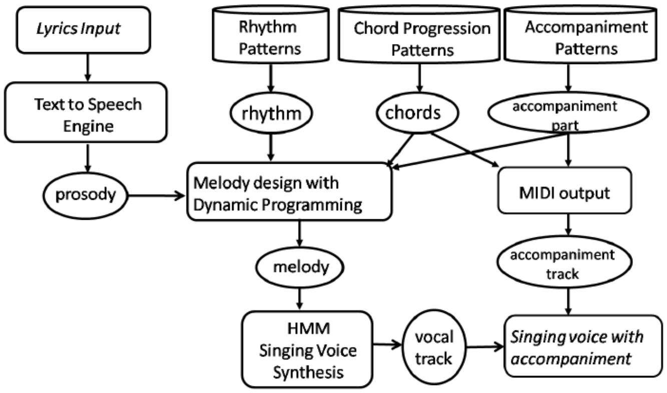


Figure 1. Songs with the lyrics input and the choices of patterns

A noteworthy fact in this journey is the integration of Natural Language Processing (NLP) within the domain of music analysis. Studies have harnessed NLP for lyric-based sentiment analysis, emotion detection, and thematic categorization. Our approach converges at the confluence of these streams, employing NLP to unravel the embedded meaning within song lyrics, thereby enriching our genre prediction framework. The symbiosis between NLP and machine learning, a recurring motif in this evolving terrain, underscores the potency of linguistic analysis in unveiling the latent dynamics of music.

The trajectory of our exploration is also shaped by the evolution of feature extraction and dimensionality reduction techniques in music analysis. Researchers have striven to distill the essence of musical compositions into succinct, informative features that underpin predictive models. Parallelly, we embark on a journey to discern pivotal lyrical motifs serving as indicators of the genre. Through meticulous feature engineering, our aspiration is to encapsulate the core essence of lyrical narratives, resonating at the very heart of musical genres.

Our endeavor harmonizes with a dynamic and continually evolving sphere of research, converging at the confluence of music and data-driven insights. As we navigate this intricate landscape, our contribution reverberates not only within the tapestry of music analysis but also in fostering a deeper comprehension of the intricate interplay between language, music, and predictive modeling. Our journey adds a lyrical cadence to the symphony of research, amplifying the melody of interdisciplinary exploration at the crossroads of art and technology.

3. Methods

3.1. Data Preprocessing

This dataset has been sourced from Kaggle and comprises information as recent as 2022, extracted from Genius. It extends upon the 5 Million Song Lyrics Dataset by employing models to ascertain the native language of each entry. It encompasses 142 languages, though we are exclusively gathering data in the English language. The resulting *filtered_data* dataset encompasses 3,399,993 rows and 11 columns, including attributes such as title, tag, artist, year, views, features, lyrics, and identifiers. The `′language_cld3′` attribute, which facilitated the initial filtering, attests to the English-language exclusivity achieved through preprocessing.

In essence, the data preprocessing endeavors have laid the groundwork for meaningful analyses and modeling, ensuring that subsequent investigations are built upon a foundation of quality and consistency. Through the fusion of language filtration, chunking optimization, and data aggregation, the *filtered_data* dataset is poised to unveil the intricate relationship between song lyrics and genre prediction. The commitment to thorough data preprocessing echoes the dedication to deriving accurate and profound insights from the musical landscape.

During our exploratory data analysis (EDA), which involved examining top contributors, most viewed songs, and conducting a year-wise evaluation, we encountered a significant amount of extraneous information. The dataset exhibited a pronounced skew towards Pop music (41%), while other genres like R&B and country only represented around 1–2% of the dataset. This imbalance in genre distribution introduces a bias in our model, making it more adept at predicting Pop compared to other genres.

To address this issue, we endeavored to construct a balanced dataset by equalizing the number of songs across genres. Following dataset filtering, we followed a series of common preprocessing steps, including the removal of special characters, tokenization, elimination of stop words, and lemmatization.

As shown in Figure 6, notable aspect was the extraction of the most common words associated with each genre. This intricate process involved a thorough examination of the complete lyrics corpus to discern the prevailing words that encapsulate the essence of individual music genres. This endeavor not only provided us with a valuable lens through which to view the thematic undercurrents and recurrent motifs inherent in various genres but also contributed to a profound understanding of the intricate tapestry that forms the musical landscape.

3.2. Implementation

The primary hurdle in this project revolves around managing extensive datasets, encompassing nearly 3.3 million songs post data preprocessing, spanning the period from 1991 to 2023. This undertaking is computationally demanding and considerably time-intensive. Consequently, we've opted to partition our dataset into distinct timelines. However, this segmentation could potentially introduce biased outcomes due to the inherent differences across these timeframes.

We have employed a variety of models, including SVM, logistic regression, and transformer models, while considering different timelines, yielding diverse outcomes. In one instance, we focused on datasets exclusively post the year 2020, training models on sample sizes ranging from 10% to 100% of the available data. Likewise, we repeated this process for the time frame spanning from 2010 to 2017, conducting separate implementations for each period. This approach allowed us to explore distinct timelines in conjunction with the various models at hand.

4. Model Building

4.1. Model-1

The Decision tree model, a widely used machine learning algorithm, is known for its interpretability and effectiveness in handling both classification and regression tasks. Its underlying structure comprises internal nodes, each representing a decision based on a specific feature, and leaf nodes, which signify the final predictions or outcomes. The model is constructed using a top-down, greedy approach, selecting the most informative feature at each internal node to maximize information gain or minimize impurity, resulting in accurate predictions[5]. Due to its versatility, decision trees can handle both categorical and numerical data, and they are often visualized to gain valuable insights into the data's underlying patterns.

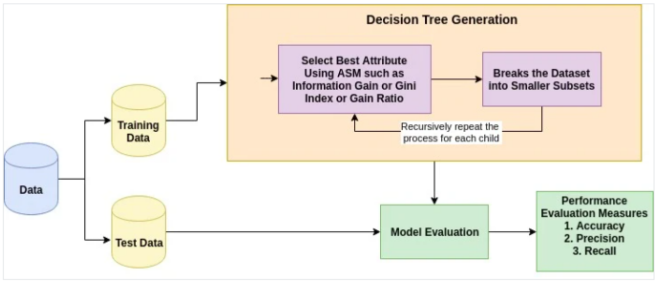

Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Training Data
Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Figure 2. decision tree model

While Decision trees offer interpretability and ease of visualization, they can be prone to overfitting, particularly when the tree becomes too deep. Fortunately, techniques such as pruning and ensembling methods like Random Forests or Gradient Boosting can help mitigate overfitting issues and improve model performance. The decision tree model finds application across various domains, ranging from finance and healthcare to natural language processing and image recognition. Its simplicity and transparency make it an attractive choice for scenarios where understanding the decision-making process is of paramount importance, providing stakeholders with confidence in the model's predictions.

4.2. Model-2

Logistic Regression is a fundamental and widely used machine learning model for binary classification tasks in Natural Language Processing (NLP)[6]. It is particularly well-suited for problems where the goal is to classify text data into two distinct classes, such as sentiment analysis (positive/negative), spam detection (spam/ham), or topic categorization (relevant/irrelevant). In the context of NLP, text data is often transformed into numerical features using techniques like Bag-of-Words, TF-IDF, or word embeddings before being fed into the Logistic Regression model.

The model's core principle is to estimate the probability of a given text belonging to a particular class using the logistic function, also known as the sigmoid function. The sigmoid function maps any real-valued number to a range between 0 and 1, representing the probability of belonging to one class. Logistic Regression utilizes a linear combination of the features, which is then passed through the sigmoid function to obtain the probability score. By choosing a suitable threshold (usually 0.5), the model can make binary predictions. Logistic Regression is computationally efficient and interpretable, making it a popular choice in many NLP applications, especially when interpretability of the model's decision is crucial.

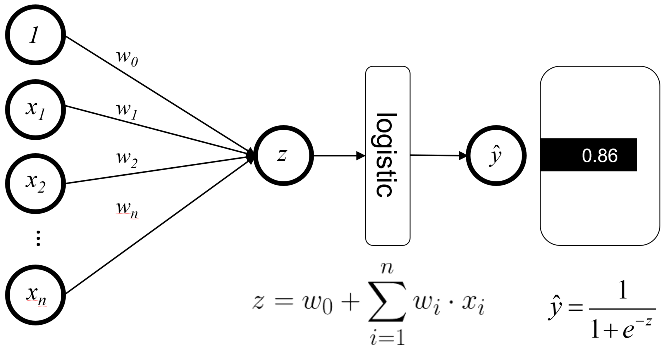

1
x_1
x_2
x_n
w_0
w_1
w_2
w_n
z
logistic
$\hat{y}$
0.86
$z = w_0 + \sum_{i=1}^{n} w_i \cdot x_i$
$\hat{y} = \frac{1}{1+e^{-z}}$

Figure 3. logistic regression architecture

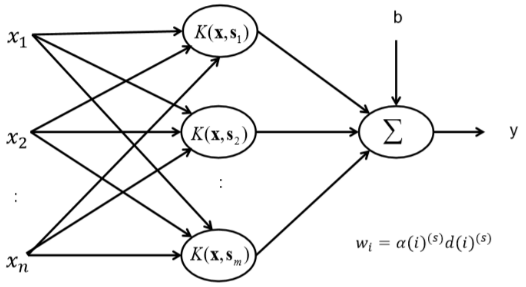
Although Logistic Regression is suitable for binary classification tasks, it can also be extended to handle multi-class classification problems using techniques like One-vs-Rest or softmax regression. In NLP, multi-class sentiment analysis, topic classification into multiple categories, or language identification tasks often benefit from these extensions. However, for highly complex and non-linear NLP problems, more sophisticated models like neural networks or transformer-based architectures may outperform Logistic Regression, as they can capture intricate patterns and dependencies present in the text data.

4.3. Model-3

Support Vector Machines (SVM) is a versatile and widely-used supervised machine learning algorithm. It is primarily employed for classification tasks, where its main objective is to find the optimal hyperplane that best separates data points into distinct classes while maximizing the margin between them[7]. SVM is particularly effective in scenarios with complex decision boundaries and high-dimensional data. It achieves this by identifying the critical data points, known as support vectors, which significantly influence the position of the hyperplane. SVM's ability to handle non-linear data is enabled by the kernel trick, implicitly transforming data into higher-dimensional spaces, allowing for more flexible decision boundaries.

The core principle of SVM is to maximize the margin between the decision boundary and the nearest data points from each class, promoting better generalization and mitigating overfitting. The regularization parameter, C, plays a crucial role in SVM by controlling the trade-off between maximizing the margin and minimizing classification errors on the training data. A smaller C value leads to a larger margin, potentially allowing for some misclassifications in the training data, while a larger C value emphasizes better classification accuracy, which might result in a smaller margin. SVM is widely applied in various domains, including image recognition, text classification, and bioinformatics, making it a popular choice for many real-world machine-learning problems.

Architecture of a support vector machine


x_1
x_2
:
x_n
$K(\mathbf{x},\mathbf{s}_1)$
$K(\mathbf{x},\mathbf{s}_2)$
:
$K(\mathbf{x},\mathbf{s}_m)$
b
Σ
y
$w_i = \alpha(i)^{(s)} d(i)^{(s)}$

$\mathbf{s}_i$ are the support vectors

Figure 4. svm Architecture

4.4. Model-4

The Naive Bayes model is a popular and simple machine learning algorithm used for classification tasks, particularly in natural language processing and text classification. It is based on Bayes' theorem, which leverages probabilistic reasoning to make predictions. Despite its simplicity, Naive Bayes has proven to be effective in various real-world applications due to its efficiency and ability to handle high-dimensional data. The algorithm assumes that all features are independent of each other, given the class label, which is why it is called "naive." This assumption allows the model to estimate the probability of a class label for a given input by multiplying the probabilities of individual features. Naive Bayes is particularly useful when dealing with large datasets or high-dimensional feature spaces, as it requires a relatively small amount of training data compared to more complex models.

One of the key advantages of the Naive Bayes model lies in its interpretability and ease of implementation. The algorithm is computationally efficient and requires minimal hyperparameter tuning, making it a suitable choice for projects with limited computational resources or time constraints. Additionally, Naive Bayes performs well in scenarios where the independence assumption is reasonably satisfied, such as text classification tasks, spam filtering, sentiment analysis, and document categorization. However, it is worth noting that the independence assumption may not hold true for all datasets, which can impact the model's performance in certain cases. Despite this limitation, the Naive Bayes model remains a valuable tool in the machine learning toolkit, offering a practical and efficient solution for a variety of classification problems.

4.5. Model-5

Transformers have revolutionized the field of Natural Language Processing (NLP) and become a cornerstone in modern language modeling. Introduced by Vaswani et al. in 2017, transformers are deep learning architectures designed to handle sequential data efficiently, with particular success in NLP tasks. The key innovation in transformers lies in their attention mechanism, which allows them to capture long-range dependencies in texts. Unlike traditional recurrent neural networks (RNNs) that process inputs sequentially, transformers can process the entire input in parallel, making them highly scalable and significantly reducing training time. With their ability to handle large datasets and vast contextual information, transformers have powered state-of-the-art models for various NLP tasks, including machine translation, sentiment analysis, question answering, and language generation. The transformer architecture has become the foundation for various pre-trained models like BERT, GPT, RoBERTa, and others. These pre-trained models can be fine-tuned on specific NLP tasks, enabling transfer learning and making NLP more accessible to researchers and practitioners. Transformers have also played a pivotal role in advancing multilingual NLP[8], allowing models to handle multiple languages effectively. Despite their remarkable success, researchers are continually exploring ways to enhance the efficiency and effectiveness of transformers to tackle increasingly complex NLP challenges, ensuring that they remain at the forefront of NLP research and applications.

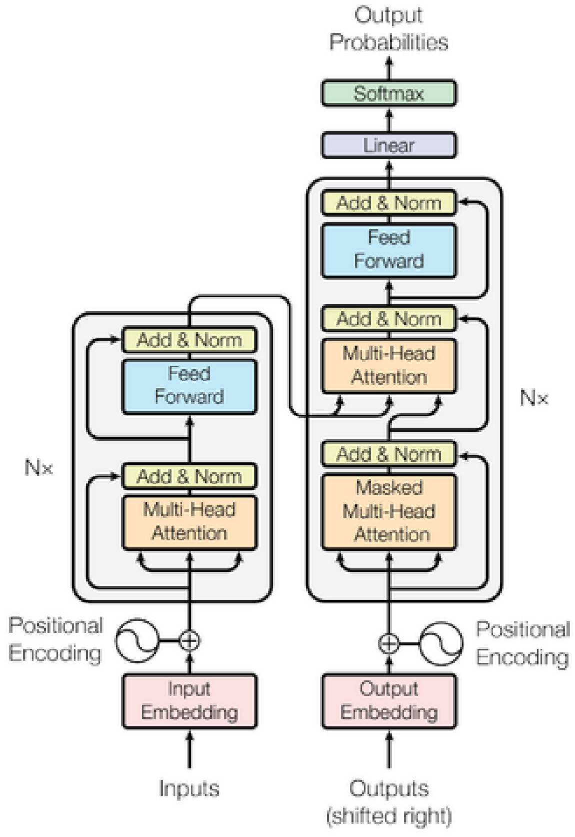


Figure 5. Transformers Architecture

5. Training

The aforementioned pretrained models serve training purposes. Among the initial four models (Decision Tree, Logistic Regression, Support Vector Machine, and Naive Bayes), they were trained across various timeframes: post-2020, 2018 to 2020, and 2010 to 2017. Within each training phase, the dataset was divided into training and testing subsets, followed by respective model training and testing procedures.

For the transfomers, it is implemented for two timelines only. This report outlines the key components for training a machine learning model using PyTorch. A batch size of 6 is chosen for efficient updates, while a learning rate of 5e-7 balances stability and exploration. The scheduler adjusts the learning rate based on validation loss changes. The Adam optimizer adapts parameters with its individualized learning rates. The CrossEntropyLoss measures training disparities. These elements combine in a loop for iterative updates, validation assessment, and learning rate adjustment, ensuring effective training for optimal performance.

6. Results

While implementing the four classical models, variations in accuracy across different time periods were observed. Logistic regression consistently demonstrated higher accuracy compared to the other models, followed closely by SVM. Notably, the decision tree and Naive Bayes exhibited similar test accuracy. Interestingly, in datasets from 2021 to the present, as the sample size increased, accuracy also improved. The peak test accuracy of 63.3% was achieved by logistic regression, while SVM achieved 58.9

Within the dataset spanning from 2017 to 2020, the accuracy ranking remained consistent with the above timeline. Logistic regression achieved an accuracy of 58.69%, while SVM achieved 54.2%. This pattern was similarly observed for the individual dataset covering the years 2010 to 2017.

Table 1. Comparison of Model Performance

Models	**2018-2020**	**2021-now**
Decision Tree	0.450176	0.512060
Logistic Regression	0.586967	0.633276
Naive Bayes	0.450176	0.512060
SVM	0.542700	0.589220

or BERT transformers, two distinct timelines were employed. In the timeline encompassing 2021 to the present, the testing accuracy stood at 64.89%. Conversely, for the 2020 timeline, the testing accuracy reached 62.32%.

Table 2. Comparison of Model Performance

Models	2020	2021-now
BERT	64.89%	62.32%

7. Discussion

In this project, we employed a diverse set of machine learning models, including transformers, Naive Bayes, decision trees, logistic regression, and Support Vector Machines (SVM), to address various aspects of our NLP tasks. The selection of these models was deliberate, aiming to leverage their unique strengths and capabilities for different aspects of the problem. Transformers, as state-of-the-art language models, were particularly effective in capturing complex contextual information, which is crucial for tasks like language generation and sentiment analysis. Their ability to handle large datasets efficiently and generalize well to different NLP tasks made them a compelling choice for transfer learning and fine-tuning on specific tasks.

Naive Bayes and decision trees, on the other hand, are simpler models that proved useful for certain aspects of the project. Naive Bayes, based on the Bayes theorem and assuming independence among features, showed good performance for certain text classification tasks with limited feature space. Decision trees, with their ability to capture non-linear relationships and interpretability, were well-suited for tasks where feature importance and explainability were essential.

Logistic regression, a widely used linear classification model, served as a baseline for comparison with more complex models. It provided a straightforward approach to text classification tasks while offering an interpretable probabilistic framework. SVM, a powerful classifier based on finding the optimal hyperplane in the feature space, showed promising results when dealing with high-dimensional data and proved to be effective for certain tasks where decision boundaries were more complex.

By utilizing this ensemble of machine learning models, we were able to gain valuable insights into the strengths and weaknesses of different approaches in NLP. The combination of these models allowed us to tackle various challenges in text processing, ranging from sentiment analysis and text classification to language generation and more. Additionally, comparing their performances provided valuable information for understanding the impact of model complexity and the importance of feature engineering in NLP tasks. Overall, our comprehensive analysis demonstrates the versatility of these models and contributes to the growing body of knowledge in the field of NLP.

8. Conclusion

In conclusion, the performance evaluation of classical models, namely Decision Tree, Logistic Regression, Support Vector Machine, and Naive Bayes, reveals varying accuracy across different timeframes. Logistic Regression consistently outperformed the other models, closely followed by BERT transformers. Meanwhile, Decision Tree and Naive Bayes exhibited similar accuracy levels. Furthermore, datasets from 2021 to the present indicated increasing accuracy as sample size grew, with BERT transformers achieving the highest accuracy of 64.89%. In parallel, SVM yielded a notable accuracy of 58.9%.

These results highlight the competitive performance of BERT transformers in the specific context of the timeline from 2021 to the present. It underscores the importance of considering distinct models and timeframes when evaluating model performance.

In the domain of sentiment polarity analysis, exploring diverse music genres uncovers distinct emotional tones. Notably, ”country” and ”pop” genres project positivity with scores of 0.443754 and 0.423590, respectively. In contrast, ”rap” displays a mixed sentiment (-0.224863), while ”rb” resonates positively (0.470551). The sentiment in ”rock” is subtle (0.086154). These polarity scores offer a unique lens to appreciate the emotional essence underlying each genre's musical expression.

Examining songs mentioning drinking across music genres unveils diverse insights. ”Country” leads with 19.43%, while ”pop” has 7.69%. ”Rap” prominently features drinking in 24.40%, ”rb” follows with 8.38%, and ”rock” with 6.90%. These percentages showcase genre-specific approaches to this theme, highlighting their distinct lyrical narratives.

Due to computational and time limitations, segmenting our analysis into distinct timeframes is essential. This approach helps manage data processing complexities. However, potential implications on conclusions should be recognized. Addressing these challenges requires careful timeline selection, transparency, and considering confounding factors. Our commitment to maintaining rigorous experimental design and interpretation ensures meaningful insights, even within these constraints.

References

[1] Rebecca Bilbro Benjamin Bengfort and Tony Ojeda. Applied text analysis with python. 1

[2] Hinrich Schutz Christopher D.Manning. Foundations of statistical natural language processing. 1

[3] D.Cope. Comupters and musical style. *A-R Editions*, 1991. 1

[4] L. Hiller and L.Isaacson. Experimental music. *McGraw-Hill,1959*. 1

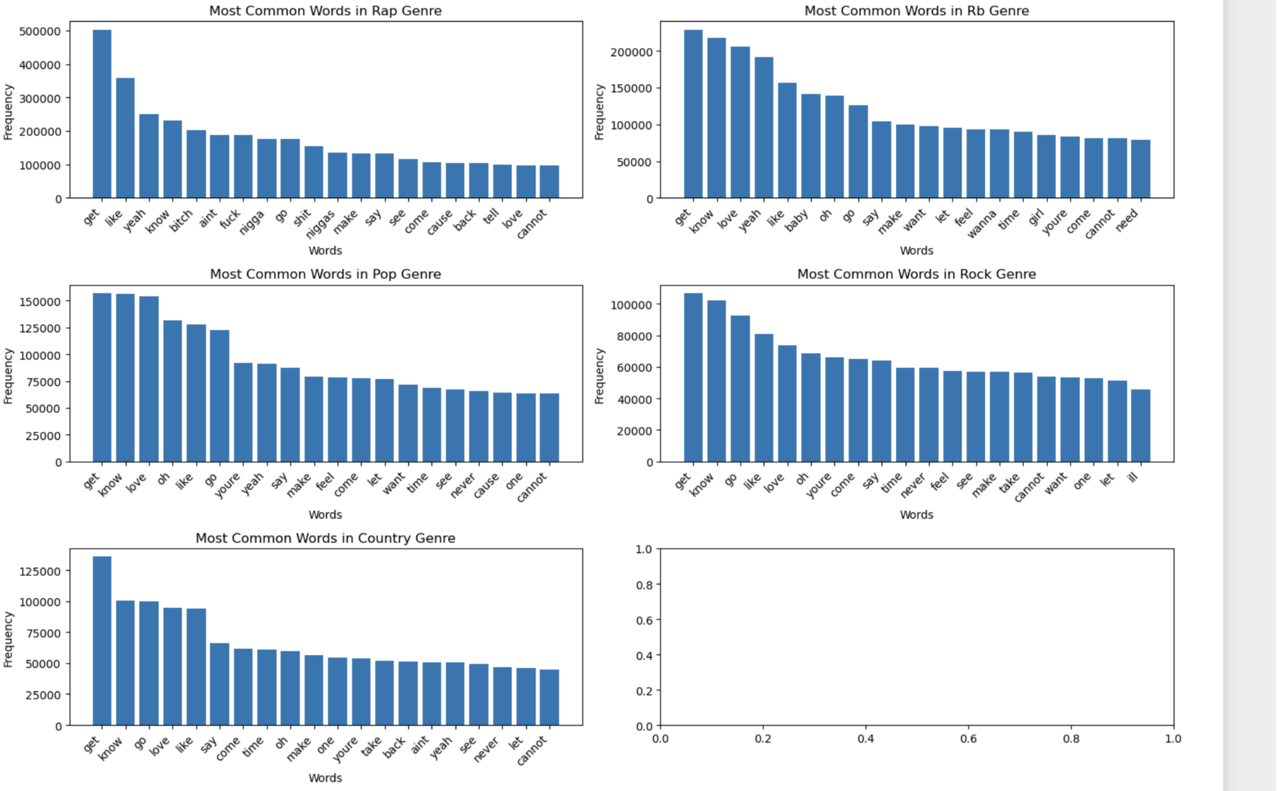


Figure 6. Most Common Words

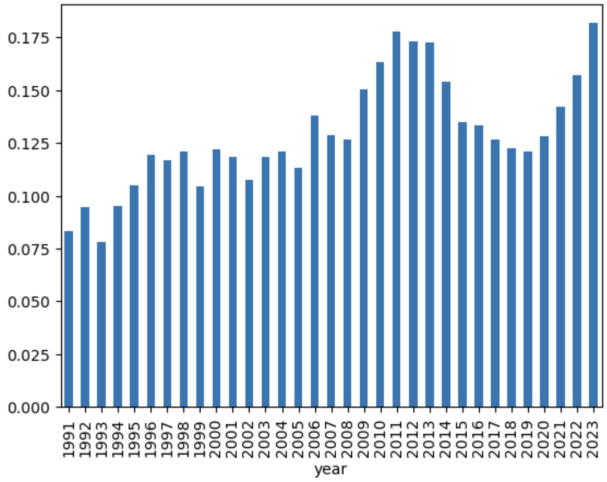


Figure 7. Drinking Words Over Time

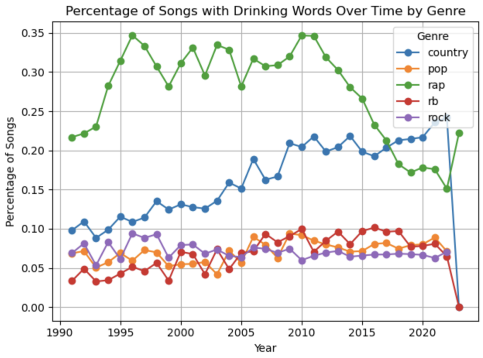


Figure 8. Percentage of Songs with Drinking Words Over Time by Genre

[5] Nitin Indurkhya and Fred J.Damerau. Handbook of natural language processing. 3

[6] Howard Lane and Hapke. Natural language processing in action. 3

[7] Sumit Pandey Palash Goyal and Karan Jain. Deep learning for natural language processing. 4

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556, 2014.*, 57(1):21–52, 2002. 5