Modelling Chart Trajectories using Song Features

by

Jonathan Perrie

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Over the years, hit song science has been a controversial topic within music information retrieval. Researchers have debated whether an unbiased dataset can be constructed to model song performance in a meaningful way. Often, classes for modelling are derived from one dimension of song performance, like for example, a songs peak position on some chart. We aim to develop target variables for modelling song performance as trajectory patterns that consider both a song's lasting power and its listener reach. We model our target variables over various datasets using a wide array of features across different domains, which include metadata, audio, and lyric features. We found that the metadata features, which act as baseline song attributes, oftentimes had the most power in distinguishing our proposed task classes. When modelling hits and flops along one dimension of song success, we observed that the dimensions carried contrasting information, thus justifying their fusion into a two-dimensional target variable, which could be useful for future researchers who want to better understand the relationships between song features and performance. We were unable to show that our target variables were all that useful for modelling more than two classes, but we believe that this is more a limitation of the features, which were often high level, rather than the target variables' separability. Along with our model analysis, we also carried out a re-implementation of a related study by Askin & Mauskapf and considered different applications of our data using methods from time series analysis.

Acknowledgements

I would like to thank my parents, friends, and supervisors all for their patience. “I’m almost done my thesis” has become one of my go-to sayings, and now that it’s almost over, I’ll have to find a new way to introduce myself to strangers.

My time at Waterloo was bumpy but enjoyable and maybe a little too enjoyable with weekly long runs with friends around Wilmot Line and Fischer-Hallman Road, pick-up soccer with CS grad students, and dirt cheap meals at the grad house.

I want to thank Mohammad for teaching me patience, Colin for his keyboard, and Ben for being a supportive office mate. I couldn’t have gotten through this experience without the people around me, so thank you.

Dedication

This is dedicated to the canyons, rivers, and mountains of Zion National Park.

Table of Contents

List of Tables

List of Figures

Abbreviations

AIC Akaike information criterion ix, 47–49, 53, 54, 58, 65, 66, 86, 87

API application programming interface 1, 5, 12–14, 71

BOW bag-of-words 7–9, 12, 16, 17, 26

DTW dynamic time warping xi, xii, 32–37, 42

LDA latent Dirichlet allocation ix–xi, xiii, 7–10, 12, 17, 18, 23, 24, 46, 47, 85–88, 90, 101

MIR music information retrieval 1, 2, 4–7, 11, 15, 17, 44

MIREX Music Information Retrieval Evaluation eXchange 5

MSD Million Song Dataset 2, 8–10, 12, 15–17, 82, 90

NLP natural language processing 17

SVM support vector machine 6, 7

VA valence-arousal 7

VIF variance inflation factor 47

Chapter 1

Introduction

1.1 Modeling chart trajectories

Music is an art form used by countless cultural groups to retell past histories and express present emotions at the individual and community levels. To better understand how culture evolves, it is important to study music's progress. In recent times, with advances in machine learning and information retrieval, this analysis of music, music information retrieval (MIR), has become more and more automated. For example, researchers have studied how sonic features have changed over time in modern popular music using computational methods to extract the features and then analyze them [27].

A well-studied subject with some controversy in MIR present since its beginnings is hit song science. It boils down to separating hits from flops, where a hit is a song from the top of the charts and a flop is, for example, a song that never even appears on the charts. Those interested in this problem include musicologists, who want to study the evolution of popular music, and label A&Rs, who want to discover the next big hit. At its simplest, as a computer science problem, this is a binary classification task using some chart measure to separate songs into two classes, and like any classification problem, one needs to train a classifier over a set of features to learn how to separate the classes. Over the years, researchers have incorporated various types of song features to train their classifiers including audio [15, 24], metadata [1], and lyric [7, 3] features. We incorporate all three types of features into feature sets used to distinguish classes of chart behaviour, which we describe as a song's activity on the charts with respect to both its relative appeal compared to other songs and its lasting power on the charts. We gather our features from multiple sources: the audio features come from the Spotify application programming interface (API), the lyric features come from the Million Song Dataset (MSD) [4], and the metadata features come from the weekly Billboard Hot 100 chart from 1958 to 2012.

Most of the tasks that we model in this thesis are binary, but we define different constraints for each task to learn specific relationships between the features and trajectories. This niche field is at the intersection of much larger disciplines, which include machine learning, musicology, cultural analysis, and product consumption, and our goal is primarily to just contribute more to the understanding of what it means to be a successful song and to learn more about the relationships between success and song features.

1.2 Contribution

We propose two target variables for class-based modelling that incorporate both temporal and position-based aspects of a song's popular appeal given its past chart data. Using a variety of feature sets and modelling tasks, we are able to demonstrate the importance of explicitly incorporating these components of song performance together into a target variable to better represent the relationship between song features and performance.

1.3 Organization

We organize the thesis into the following chapters:

- **Chapter 2 (Related work):** This is where we review some of related literature in MIR, which explore hit song science and other classification-based tasks using audio and lyric features.
- **Chapter 3 (Data and pre-processing):** In this chapter, we detail the steps taken to gather our data and perform pre-processing, so it will be ready for modelling.
- **Chapter 4 (Target variables for song popularity analysis):** In this chapter, we detail the processing steps required to construct our target variable classes. We consider two types of target variables: the first is based on discretizing aggregate measures of song performance, and the second is based on clustering song chart position trajectories using an alignment score as a distance measure.
- **Chapter 5 (Modelling):** This is where we model the classes using our feature sets. We examine three classification problems to uncover different types of relationships

between our features and the classes; the questions correspond to a simple binary modelling task, many one-dimensional binary modelling tasks, and a multi-class modelling task.

- **Chapter 6 (Critique):** Here, we review a related paper by Askin & Mauskapf [1], whose work inspired some of the work in this thesis, and examine their results in the context of our own findings.
- **Chapter 7 (Time series analysis):** In this chapter, we step away from modelling and touch on other areas involving time series analysis where our data could be applied to explore temporal trends in popular music.
- **Chapter 8 (Conclusion):** Finally, we highlight our findings, look at some of the critical elements in this work, and propose ways to address them in future studies.

Chapter 2

Related work

Our goal is to learn what constitutes a hit and what relationships exist between hits and song features by modelling multiple sets of distinct target variable classes with different feature sets, which include audio, lyric, and metadata features. This has been explored to some extent by past researchers who have modelled hit song science tasks with various feature types and used different data representations to study musicological aspects of their features including their relatedness [21], context [1, 3], and influence [2, 28]. In this chapter, we review some of these findings, first examining MIR modelling tasks and then examining feature analysis in MIR. This is done to highlight past research, justify some of our experimental decisions, and show how we contribute to the field.

2.1 Modelling

Hit song science is not the only modelling task that uses song-based features within MIR. Two other popular modelling tasks are genre classification and emotion detection, and together these three tasks make up one of the more accessible areas of MIR. In genre classification, the aim is to train a classifier using song features like a song's acousticness to learn how to distinguish songs based on their genre [34]. For example, a classifier might learn that songs with higher acousticness feature values are more likely to belong to the folk genre over the pop genre. The aim in emotion detection is to learn how to distinguish songs based on the emotions they convey given some model of emotion [20]. For example, a classifier might learn that songs with slower tempos are more likely to make listeners feel negative according to a sentiment model of emotion. Each of these modelling tasks has its own set of ambiguities around what is being modelled; however, there has been some work in developing research-friendly datasets for training and evaluating classifiers in MIR.

Genre is a straightforward way of categorizing music, as almost every modern commercial song is prescribed a genre on release, so it can be marketed to a specific demographic. The staple genre classification dataset is the GTZAN genre collection consisting of one thousand 30 second audio tracks from ten genres [34]. It has been widely used but also criticized for its lack of genre coverage and mislabellings [33]. A more general criticism of genre classification is that genre is a loose concept; some genres are more closely related than others and some artists operate between genres with music that fuses multiple genres together. Consider the recent controversies around Justin Bieber's use of dancehall [11] or Lil Nas X's use of country [31]. Do these songs belong in pop and rap or dancehall and country? In the latter case, even Billboard had trouble deciding when it manually removed Lil Nas X from the Hot Country Songs chart after his debut appearance [31]. While we do not model genre, we do use it as a metadata feature, and so we have to deal with these issues around what level of granularity to use for genre categorization. Spotify's API provides genre tags for each artist, which we make use of in chapter 3, but the tags have varying levels of granularity and are sometimes abstract, so we have to develop a method for generalizing them into conventional genres like pop and rap.

While we just presented arguments in favour of genre being subjective, relative to emotion detection, genre classification involves well-defined and distinct classes. In emotion detection, classes are much more subjective. There have been some crowd-sourcing efforts in the field using music streaming services like Last.fm [13], but for the most part, researchers have had to generate their own data by recruiting participants to describe a song along multiple dimensions of emotion. There is even a challenge hosted by the Music Information Retrieval Evaluation eXchange (MIREX), which was first introduced in 2007 [14], where emotion detection classifiers are evaluated on their ability to classify 600 songs into five emotion clusters annotated by domain experts. Some critics of this task have argued that music is a personal experience dependent on one's cultural background, and so ground truth cannot be easily defined [30]. Interestingly, in the most recent MIREX competition in 2018, the K-pop mood task was split into two tasks, one involving American annotators and the other involving Korean annotators. As added evidence against a universal ground truth for emotion, Singhi & Brown [30] observed a clumping of emotional responses based on ethnic group when they surveyed students' responses to mainstream pop music and lyrics. A parallel can be drawn from this problem to hit song science, where often the quest is to find some global hit, but very few songs reach the heights of Gangnam Style and Despacito. People's musical preferences are often quite personal and so recommendation systems are much more practical in finding what individuals will enjoy listening to [35]. Hit song science allows us to look at a higher level picture of what features were historically important for popular songs over some time period.

The last modelling task from this trio is hit song science, which is the focus of this thesis. It is often described as a binary classification problem where a classifier is trained to distinguish hits from flops, but we would like to argue that more complex problems can be considered because the charts allow for a higher degree of flexibility in designing popularity classes. It is standard practice for MIR researchers to scrape song data from charts when trying to establish hit and flop classes, and they often define hits as songs that were #1 on some chart, but this is not necessary. If a song is on a chart that updates regularly, then the song's performance can be defined in a way that incorporates both temporal and position-based aspects of song success using the song's chart activity. In chapter 4, we chose to incorporate these elements into our own definitions of target variable classes to model more specific chart behaviour patterns.

2.1.1 Prediction

Predictive modelling is where one trains a model and then evaluates it on unseen data. Within hit song science, this area has been contentious with debate around what it means to learn hit song features. In one early study in 2005, Dhanaraj & Logan [7] trained a support vector machine (SVM) on lyric features derived from a topic model and audio features transformed into a reduced cluster feature space. Their dataset consisted of 1,700 songs with 91 #1 hits, and they found their best results, identifying the most hits, when combining feature sets, but these results were only marginally better than their lyric feature results. Interestingly, their data spans a similar range to our own, and the lyric features they found with negative effects on popularity corresponded to unique diction used in specific genres like rap and metal. What this might imply is that the lyric features they found are biased against certain genres, which are underrepresented as hits or do not have their effects controlled for. As we also use topic model lyric features in our modelling tasks, we will be watchful of these effects appearing in our own models.

Dhanaraj & Logan along with other early hit song science researchers faced criticism for making claims about what song features lead to success. Patchet & Roy [24] authored one of these critiques where they took issue with the claims around learning something about song popularity. Generally, Pachet & Roy's issues with the field were with what they perceived to be biased experiments and the use of spurious data. They conducted their own hit song classification experiment with a negative result to try to argue against hit song science. They evaluated three feature sets using an SVM applied to song subsamples drawn from a dataset of 32,000 songs where the subsamples were balanced based on the feature proportions. Over three tasks for distinguishing hits, moderate successes, and flops, they found only modest success with state-of-the-art features relative to a dummy classifier. Because of these results, they argued that even hand-crafted state-of-the-art features were not sufficient to learn how to identify popular songs. Even with the advent of deep learning, researchers in genre classification have demonstrated that automatically learned features can be perturbed in a minor way using adversaries to drastically affect classification results [18], and if this is the case for genre classification, then the same vulnerabilities are likely true in hit song science.

Like Dhanaraj & Logan, some researchers from the field of emotion detection have found that combining audio and lyric features works best for modelling emotions in the valence-arousal (VA) space [19, 39]; however, Hu et al. [13] did not observe this fully when they modelled songs using 19 manually selected moods derived from Last.fm tags. They found that for some classes spectral audio features worked best, and in others, the best results came from either lyric features represented by a language model or an SVM or a combination of the lyric and audio features. VA modelling is often proposed as a 4-class classification task, and so it may seem odd that combining features into a larger set works better for a simple VA task and using fewer and more specific features works better for a complex emotional task with 19 classes, but this likely has to do with Occam's razor and the risks of overfitting because of there being too many features in a model purposed to distinguish too many classes. This contrast in models could also occur in our own data as we consider modelling tasks that are binary and multi-class. In our own experiments, we consider a variety of feature sets: the metadata features, the metadata features with The Echo Nest audio features, and those features combined with lyric features derived from a topic model of varying size. We use model estimators like accuracy or log-likelihood to evaluate the quality of each feature set and balance tradeoffs between model fit and simplicity, though using a larger feature set has another benefit as we are able to examine a wider ranger of relationships between the features and the target variables.

Using topic models to derive lyric features is a common practice in MIR, but it is not the only way to get text features. Hu et al. used a language model [13], and others have used vector space models [38], rhythmic features [29], and song structure [8]. With topic models, song lyrics are transformed from a count-based representation like bag-of-words (BOW) to a topic mixture representation, where each topic is a probability distribution of words from the count-based representation. For example, a topic model might find that a breakup song has two dominant topics, one referring to being sad and the other referring to falling in love. The being sad topic might have words like tear and cry as being more probable, whereas, the falling in love topic, might have more probable words like smile and heart. We use LDA [5], which is a generative probabilistic model, to derive our lyric features, as it has been previously used with the MSD to find meaningful topics [32]; however, there is one drawback to its application to our data: it assumes that words and documents are exchangeable. For a song, the words being exchangeable does not matter to us as we are using a BOW representation from the MSD, but the songs being exchangeable matters because we have songs from different time periods, and it is unrealistic to have topics that are equally influenced by songs released in, for example, the 1960s and 2000s. A more realistic approach would learn topics from songs released earlier on and then adjust those topics with each wave of new songs to represent the topics' changes. The reason we do not do this is because we are using these features for modelling, and so we want their definitions to be fixed and easily interpretable.

2.1.2 Explanation

Outside of predictive modelling, there has been some work by social science and marketing researchers looking at why some songs are successful while others are not. Music is a popular domain for these researchers because of its accessibility and lower complexity. Relative to other cultural markets, music data can be scraped from the web, and it can be represented by a well-established set of features like those found in the MSD [4]. These researchers are generally more interested in qualitative measures derived from a model fit to all of the data like feature coefficient magnitudes and statistical significance. Recently in this field, there has been a trend towards deriving context-based features. In particular, researchers have been interested in asking questions around how a song's context relative to other recently released songs is important to its success. By modelling song performance using peak position and weeks on chart as target variables, Askin & Mauskapf [1] attempted to answer this question. They hypothesized that the most successful songs would be those that were optimally differentiated from their competition, so they would have different feature values from the average song but that difference would not be so extreme that the hit song would be on the left tail end of some feature conformity distribution; instead, it would lie somewhere in the middle. Based on their models using Billboard chart data and The Echo Nest audio features, they observed an inverted U-shaped distribution over song success based on conformity to other songs on the chart. Another example of context-based feature analysis comes from Berger & Packard [3]. Instead of audio features, they used lyrics features derived from the topic model LDA applied to lyrics from songs sampled over three years from multiple genre-specific charts. As a result, they constructed topic-mixture profiles for all of the songs and compared them with each other using Ireland and Pennebaker's language style matching equation [16], a lyric dissimilarity measure. When they modelled the songs, they found that the lyric dissimilarity had a significant positive effect, so songs that were more differentiated from their genre were more likely to succeed.

In our own analysis, we make use of similar techniques outlined by these researchers. Some of our features are context-based measures where the context is dependent on other songs in the charts, in the same genre, or in the same artist's past discography. We also follow the same workflow of an explanatory model, fitting all of the data to the model and then evaluating the features based on the magnitude of their effects and their significance; however, we do not end our analysis here. We also evaluate features based on their predictive abilities. In this way, we get to look at things from both predictive and explanatory viewpoints. This thesis is largely inspired by the work of Askin & Mauskapf [1] and could be seen as an extension of their analysis as we draw from the same data but include lyric features into the analysis and look at more complex modelling tasks.

2.2 Feature analysis

Outside of the modelling realm, some research has been conducted to better understand how song features evolve, and learn what this means for relationships between songs, artists, and genres. This area is split into two parts: some researchers focus on analyzing lyric features, while others are interested in audio features. This is not the focus of our thesis, but in the last chapter, we briefly explore some of these topics using rudimentary methods from time series analysis to offer another perspective on how our data could be analyzed.

2.2.1 Lyric features

Of the two feature types, lyrics are easier to interpret because they are derived from words, but there can still be challenges around understanding specific lyric features like topic mixtures, since they are probability distributions over all of the words from the lyrics corpus, and that is likely a lot of words. The easiest way to interpret topics is by manual inspection of the most probable words in a topic, and we do this when we run LDA on our lyrics. It works for topics with unique words, but if a topic has a generic word in its set of most probable words, then it becomes more difficult to interpret the topic. Sterckx et al. [32] worked on this problem and tried to make topic models trained on lyrics more interpretable. First, they applied LDA to the BOW lyric features from the MSD and a supervised model, labeled LDA, to a smaller corpus with ground truth topics defined, and then they compared the topic distributions between the known and unknown sets using the cosine similarity and looked for outlier matches, two topics from different sets with a high similarity score not shared with other topics. In this way, they were able to append conceptual labels to some of the topics found by LDA applied to the MSD like the label Christmas to the topic distribution with the words christmas, bells, snow, santa, and ring having high probability. In another example looking at lyric trigrams instead of topic mixtures, Atherton & Kaneshiro [2] examined how artists, songwriters, and genres influence each other by building networks with these elements as nodes and their weighted lyric trigram tf-idfs as links. They found highly centralized networks form in all three domains, which they noted contrasts with previous work related to genre similarity where genres like rap and metal were believed to be isolated due to their expletive-filled lyrics [8].

Lyrics have also been looked at from a temporal perspective. Shalit et al. [28] combined three topic models together based on the dynamic topic model and the document influence model to model present topics, time-lagged past topics, and future topic influences. They gathered lyrics from the MSD and validated their findings using a mishmash of internet sources commenting on how influential songs were. By analyzing the evolving topics, they were able to look at the topics' genre composition and infer the influence genres had over each other. They were also able to identify a number of influential songs, like Run-D.M.C's *Is it Live* from the album *Raising Hell*, which has been heralded by many as launching hip hop's golden age [37], and with these influential songs, they identified two recent musical eras, the 1970s and 1990s, where music had a heightened influence. In another example of temporal analysis of lyrics, Johnson-Roberson & Johnson-Roberson [17] examined how rap differs temporally and geographically. They scraped lyrics from the web and gathered metadata from The Echo Nest and Spotify. Using LDA and Dirichlet-multinomial regression, a feature-based topic model, they were able to observe trends in word usage through time and within specific regions of the United States. This also allowed them to identify artists forming subgenres within rap.

2.2.2 Audio features

The other feature type, the audio feature, has also been studied from a temporal perspective. Serrá et al. [27] analyzed the MSD and found that over the last 50 years, pitch, timbre, and loudness have followed the same distribution patterns; however, they found that the average loudness of songs is rising, which they argue implies a decline in sound quality. While we do not use any of their features, pitch and timbre might have analogs in The Echo Nest feature set like valence and acousticness. Loudness is in The Echo Nest feature set, but we follow the lead of Askin & Mauskapf [1] and exclude it from our analysis because there can be a difference in a song's loudness based on the medium e.g. a CD or the radio. Another experiment involving analysis of audio features was conducted by Interiano et al. [15]. They modelled song popularity using features from AcousticBrainz, a crowd-sourcing platform for sharing music data, and some of the features they used were binary representations of acoustic features like timbre. Along with modelling, they performed primary temporal analysis of their features plotting the aggregate time series against each other, so mood labels were compared with binary acoustic features and genre labels. When we apply time series analysis methods to our data, we follow similar procedures to construct time series, but our analysis goes further than just a visual inspection of the variability in the time series.

2.3 Summary

We have reviewed some of the literature on modelling and feature analysis in MIR and provided some details about our own experiments as well the controversies that persist in this field.

Chapter 3

Data and pre-processing

As was mentioned in the introduction, we gather our song data from three sources. The Billboard Hot 100 weekly chart is used to gather metadata features and target variable data; Spotify's API is used to gather audio features from The Echo Nest; and the MSD is used to gather lyric features from Musixmatch. All of these sources of data are independent, and so they have different nomenclature, which present challenges around keeping track of the same song across sources. In this section, we review our strategies for stitching together song records from different sources and detail some of the steps taken to pre-process features before they are used in a model. These pre-processing steps include applying LDA to BOW lyric features to produce topic mixture features, deriving a genre label from a song's Spotify tags, and calculating a genre-weighted context score for each song based on its The Echo Nest features.

3.1 Billboard Hot 100 Weekly Chart

One of the goals of this thesis is to better profile charting songs, and in order to do this, we need to construct richer definitions for hits, flops, and in-between. Billboard has many charts, but the weekly Billboard Hot 100 has been around for the longest. Since it is a weekly chart, it gives us the level of granularity needed to build target variable classes with specific properties. While we leave our discussion of target variables to the next chapter, this chart is also used to derive metadata features, which we use as baseline control variables to represent the qualities of a song that are less interesting like when the song was released or what genre it is associated with. While these features are less intrinsic to a song's identify, they may still have discriminatory statistical power to distinguish songs that follow different chart behaviour patterns.

Billboard Hot 100 records were gathered using billboard.py, a Python API for accessing chart records [12]. Each *ChartData* API call returns a list of Billboard's top 100 songs given a specific date. The list contains individual song objects, which include attributes for artist name, song title, date, then-current number of charting weeks, and the current, last, and peak week's positions. In total, we gathered 283,900 weekly records for 25,349 songs from mid 1958 to the end of 2012.

3.2 The Echo Nest audio features

We used Spotify's API to gather audio data originally created by The Echo Nest. The Echo Nest was a company that provided music data to the media, developers, and researchers through a public API, but in 2014, it was acquired by Spotify, and its API was shut down. Since then, Spotify has integrated some of The Echo Nest's API features into its own API, and so while we do not know the number of songs that have The Echo Nest features available, we can make API calls to Spotify's library, which consists of tens of millions of songs. The Echo Nest features are high-level audio features, which makes interpretation simple but may limit model performance. We first became interested in these features after learning about Askin & Mauskapf's [1] study, as they used The Echo Nest features to investigate what properties were associated with successful songs on the weekly Billboard Hot 100 chart. The Echo Nest features include acousticness, danceability, energy, instrumentalness, liveness, speechiness, tempo, valence, song length, key, mode, time signature and loudness. We excluded loudness from our analysis because of inconsistencies in its value depending on the medium used to deliver a song.

In Table 3.1, we have The Echo Nest feature descriptions, detailing their scale in our models, whether they are used as controls, their definitions, and their base values. The three variables that were re-scaled are tempo, time signature, and song length. Tempo was normalized by dividing each value by the maximum tempo value, so it would be on the 0-1 scale like the other continuous The Echo Nest features. Time signature and song length were converted into control variables for 4/4 time and long songs. A long song is defined as a song that was two standard deviations above the mean song length of charting songs from the year before its debut. It was used as a control because longer songs would often be shortened for radio play.

Now that we have two data sources, we want to find overlapping records, so we can combine the Billboard chart data with The Echo Nest feature data for each song. This

Table 3.1: The Echo Nest audio features summary table.

Attribute	**Scale**	**Control**	**Definition**	**Base attribute value**
Acousticness	0-1	No	Confidence of song being acoustic	Electronic song
Danceability	0-1	No	Ability to dance to song	No dancing
Energy	0-1	No	Song intensity	Slow and quiet
Instrumentalness	0-1	No	Likelihood of song containing vocals	High confidence
Key	0-11	Yes	Overall song key	The key of C
Liveness	0-1	No	Presence of an audience	Professional recording
Mode	0-1	Yes	Song modality	Minor
Speechiness	0-1	No	Presence of spoken words	Instrumental
Tempo	0-1	No	Normalized beats per minute	A slow song
Time signature	0-1	Yes	Binary control for 4/4 time	Not 4/4 time
Valence	0-1	No	Measure of positiveness	A negative song
Song length	0-1	Yes	Binary control for long songs	Below $\mu + 2\sigma$ length

requires us to develop a match finding protocol across some set of reliable song fields. We used a song's title and its artist name as the fields to evaluate in the protocol because across sources, these fields have to be the same or very closely related for song records to correspond to the same song and match.

Between Billboard and The Echo Nest, we were able to find 20,563 matches. Matches were found by removing the following song nomenclature from the Billboard artist name and song title : "Featuring", "&", and "Or". Next, the filtered artist name and song title were used to query Spotify's API for matches using their *search* method and retrieving the most relevant result. Each *search* call returns a song ID and an artist ID. Song IDs were applied to the *audio_features* method to retrieve song audio features, and artist IDs were applied to the *artists* method to retrieve a list of genre tags associated with an artist. As an ad hoc test to validate our match finding protocol, we took a random sample of 50 matches and compared their Billboard and Spotify artist names and titles. While nomenclature was sometimes different, there were no mismatches.

When we applied a related match finding protocol to Musixmatch and the intersection between Billboard and Spotify, for which we had lyric and audio features, we found 7,726 song matches, which we denote as the complete dataset. We can observe The Echo Nest feature distributions in Figure 3.1. The red distributions correspond to the intersection between Billboard and Spotify, and the blue distributions correspond to the complete dataset. As the distributions for each set of songs follow similar shapes, we argue that the second match finding protocol does not significantly alter The Echo Nest feature distributions. The features themselves follow a variety of distributions, which we can approximate. Energy, liveness, danceability, and tempo follow skewed normal distributions; speechiness, song length, acousticness, and instrumentalness follow power law distributions of varying severity; valence follows a left-skewed beta distribution; and the controls follow unique distributions with time signature and mode being lopsided in favour of 4/4 time and major songs while key follows a more balanced distribution.

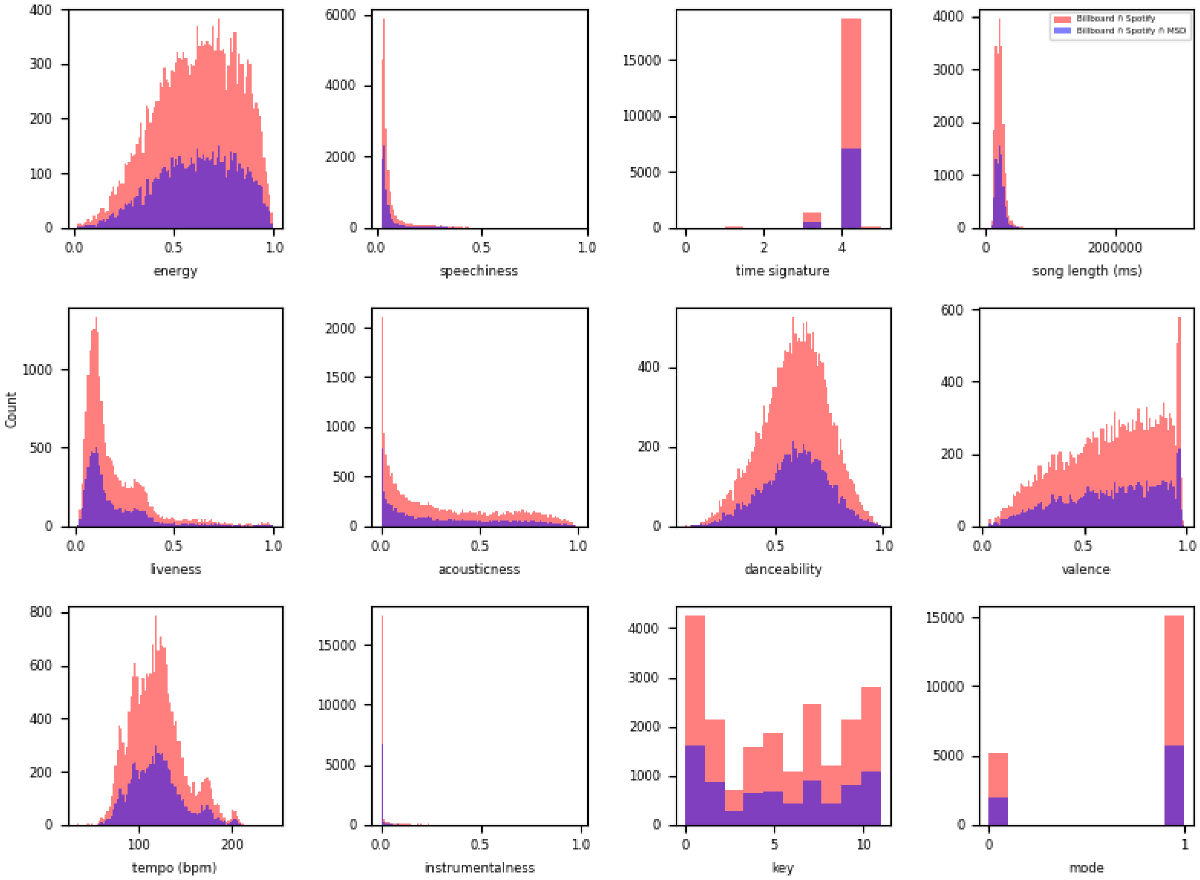


Figure 3.1: The Echo Nest audio feature value distributions. (Red) Songs from the Billboard and Spotify intersection. (Blue) Songs from the complete dataset.

3.3 Musixmatch lyric features

We gathered lyric data from Musixmatch through the MSD, a project that aims to provide MIR researchers with song features at a large scale [4]. Musixmatch features are in a stemmed BOW format, which contains counts of the 5,000 most frequent word stems. From Bertin-Mahieux et al.'s dataset of 237,662 songs, they are able to represent 92% of all song lyrics using the BOW representation. The MSD integrates many data sources together; however, one drawback of its breadth is that its naming protocols are not standardized to one form. As a result, we have to handle a larger set of song nomenclature. We filter out the following nomenclature from the MSD: square and curly braces, "feat", "ft", "Featuring", "&", and "Or". We decided to use a string matching method to find matches because we had to rely on finding matches ourselves instead of using Spotify's match finding black box. Given two strings, *SequenceMatcher* from the Python standard library *difflib* recursively searches for the longest contiguous string shared by the two strings and then scores them based on their similarity. We chose this method to score artist names because of its effectiveness at identifying true matches and found that using a threshold of 0.6 worked well for balancing false positives and false negatives. For song titles, we used exact matches after filtering; the reasoning against a string similarity for song titles was that titles are much more likely to be conserved versus artist names, which can deviate based on spelling, format, or version.

To summarize, for us to consider two song records as the same, they need an exact match between their filtered titles and a partial match with at least a *SequenceMatcher* similarity above 0.6 between their filtered artist names. We again performed another round of ad hoc tests to validate the match finding protocol. When we sampled 50 random songs, we found no mismatches, and when we sampled 50 songs with similarities below 1, we found 9 mismatches. In total, we had 659 songs with similarities below 1; the overall mean similarity is 0.985 with a standard deviation of 0.061, and the mean similarity for songs with a similarity below 1 is 0.820 with a standard deviation of 0.116.

As was mentioned earlier, we found 7,726 matches across all three sources. These matches were well distributed through the time range between 1958 and 2012. Figure 3.2 shows weekly counts of songs from the complete dataset. Every week, there were usually at least 30 songs on the charts that we could access from our data. The average number of accessible songs each week is 36.63 with a standard deviation of 8.78. What this means is that our results will not be temporally biased or limited in scope to one time period. Even so, we use time blocks, which we explain later in the chapter, to control for the effects of songs released over different periods.

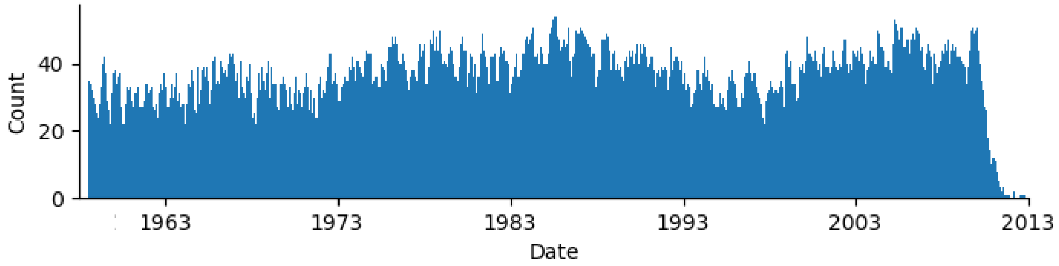


Figure 3.2: Weekly variation in the number of accessible song records from the complete dataset.

3.4 Feature derivation

We used a variety of feature derivation techniques to make the features more interpretable and easier to model. In the introduction to this chapter, we alluded to pre-processing techniques for deriving lyric features, genre labels, and a context measure. We detail these techniques along with others in the subsection below.

3.4.1 Topic mixtures

Using Musixmatch, we are able to retrieve a BOW representation of the lyrics for our data. While this representation is informative, it offers too much detail for most simple models. If we included a BOW representation for each song, then we would have to extend the songs' feature vectors by 5,000 more elements as there are 5,000 words in the BOW representation, and this would likely lead to overfitting. In natural language processing (NLP), a common technique for reducing count-based vectors is to use topic models, so instead of a BOW representation one will have a topic mixture representation. We follow this procedure and use the topic model, LDA, as it is a common choice for MIR researchers [32, 3] and has been applied to the MSD before. We use an LDA implementation from MALLET, a Java-based package for NLP [22].

Instead of using 5,000 lyric features, we considered LDA models with 10, 20, 40, and 80 topics resulting in that many topic mixture features. As topic size increases, topics become more specific, fine-grained, and better able to represent differences between documents. A rudimentary metric for evaluating topic model quality is perplexity, which is the log-likelihood of an unseen set of documents being observed given the topics and some topic hyperparameters used to define the prior topic distributions. Typically, perplexity is calculated using cross-validation. Lower perplexity scores correspond to a better topic model. Table 3.2 shows the 10-fold cross-validation perplexity scores for each topic model, which reveals that having more topic mixture features means we are better able to represent the song lyrics. We build our topic mixtures using LDA applied to the complete dataset instead of the entire Musixmatch corpus. One concern mentioned earlier is that LDA relies on two assumptions of exchangeability for words in a document and documents in a corpus. Our data is temporal, and our documents are songs that were released at different time periods, so the topic mixtures produced may not be realistic over all time periods, but we justify this as we are using them in modelling tasks, so we want the feature definitions to be fixed.

Table 3.2: Perplexity scores for LDA models trained on song lyrics from the complete dataset using 10-fold cross-validation.

Number of topics	**Perplexity**
10	592.34
20	538.62
40	476.52
80	408.45

Another approach for evaluating topic models is to look at individual topics and see how coherent their most probable words are. Do they point towards some unified theme? We use this approach when describing the importance of different topics in our predictive models. The benefit of this is that it is qualitative, but it cannot be easily used to evaluate the overall quality of a topic model. Note that when we refer to a specific topic m from an LDA topic model with n topics, we use the notation $\mathrm{T}_{n,m}$.

Table 3.3: Sample topics from separate LDA models showing how one topic from a smaller topic model can split into two topics in a larger topic model.

Topic	**Top five most probable words**
$\mathrm{T}_{10,0}$	light, rain, feel, like, fire
$\mathrm{T}_{20,15}$	away, walk, rain, run, sunshin
$\mathrm{T}_{20,18}$	light, sky, sun, blue, come

Across feature sets, we use separate topic models; however, they can point to the same ideas at different levels of granularity. Sometimes the topics will even split into separate concepts, so a topic in a smaller topic model will be represented by two topics in a larger topic model; Table 3.3 shows an example of this phenomenon. All three topics in the table refer to words related to nature and the classical elements, but they do not share the same topic model. $T_{20,15}$ and $T_{20,18}$ refer to some of the words found in $T_{10,0}$ e.g. light and rain, but they have little overlap with each other; thus, we can assume that $T_{10,0}$ splits into two separate topics.

3.4.2 Genre

Spotify's genre tags offer fine-grained detail at the artist level, which we can use to build higher level genre labels to control for genre effects in our models. Every artist has a specific set of Spotify tags to describe how their music is classified with the exception of a few duplicate artist records; these tags are likely used to help build relevant playlists for individual listeners on Spotify. The Beatles, for example, have the tags Merseybeat, British invasion, psychedelic rock, rock, album rock, and classic rock. These tags refer to the subgenre British rock and so Spotify might recommend Beatles listeners other British rock groups like the Rolling Stones. We did not use such specific labels for our study. Instead, we considered 13 genres common in Western music: hip hop, rap, rock, metal, folk, country, blues, R&B, soul, disco, funk, pop, and "none" of the above. These genres were chosen to mirror the analysis of Askin & Mauskapf [1] as they used related genres as control variables.

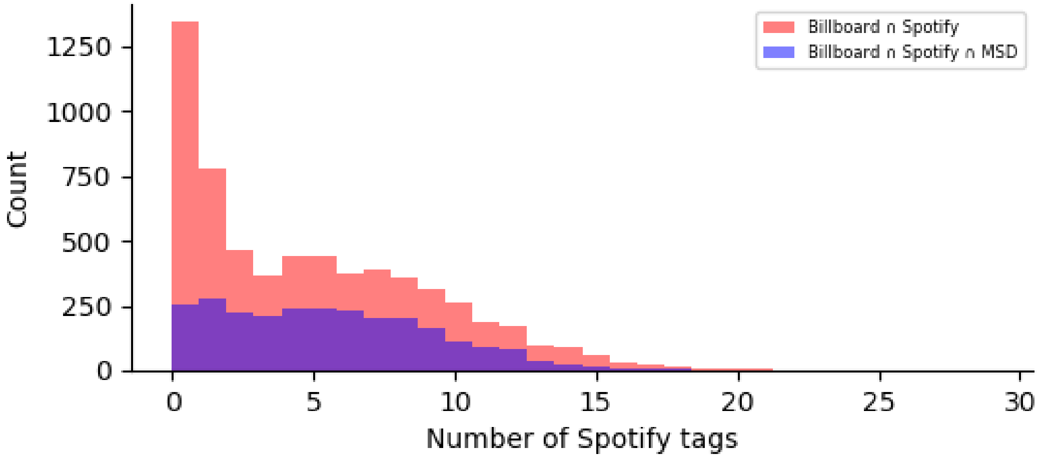


Figure 3.3: Distribution of Spotify tags associated with each artist. (Red) Songs from the Billboard and Spotify intersection. (Blue) Songs from the complete dataset.

From the Billboard and Spotify intersection, we found 20,563 songs produced by 6,026 artists, and from the complete dataset, we found 7,726 songs produced by 2,571 artists. Across both datasets, each artist had an average of around 5 tags. Figure 3.3 shows the tag distribution for both datasets. It is evident that a large portion of artists with few tags from the Billboard and Spotify intersection are not included in the complete dataset. This may be because these artists did not have lyrics in their songs or because of copyright restrictions around the sharing of their lyrics.

We transform an artist's Spotify tags into a genre label through a two step process. In the first step, we find a list of the most important tags for each genre, and in the second step, we find the genre with the highest overlap between the genres' tags and an artist's tags. The methods are described below at a high level with assumptions that we have access to song records with the artists' tags and a list of genre labels which map to some of their related tags, so for example, the rock genre label as a string partially matches some of the rock tags like album rock and classic rock.

Finding each genre's most distinguishable tags

1. Find all of the songs that include the genre label in one of their tags.
2. Count the occurrences of each tag in the set of songs.
3. Subset the tags over some count threshold; we used 30.
4. Calculate the tf-idf for each tag where the document is the genre label.
5. Subset the tags for each label based on some tf-idf threshold; we took the top 10 tags.

Finding the overlap between genre and artist tags

1. Count the number of matches between a genre label and the song's artist tags for each genre.
2. Assign a genre label to a song based on the most common label.
    - If all labels have no matches, then use the "none" label.
    - If one label has the highest number of matches, then use that label.

- If multiple labels have the highest number of matches, then randomly pick one and mark the song as a crossover.

For the first method, we chose thresholds to minimize the overlap between genres, but this can be challenging as many of the genres have different levels of granularity. Consider the case where we have the genres metal and rock in our list of desired genres that we want to use for annotation. We would start by finding all songs that have these genres in their Spotify tag lists (note that tags are artist-specific) and then we would count and threshold them. Imagine that the tags rock, metal, and hard rock all have counts above some threshold. Calculating the tf-idf for each tag, we might find that the rock genre consists of a set of frequently used tags, which include rock, while the metal genre includes the less frequently used metal tag. The hard rock tag could exist on both genre lists, and more generally, the overlap in tags between genres gives us a way to gauge how closely related genres are.

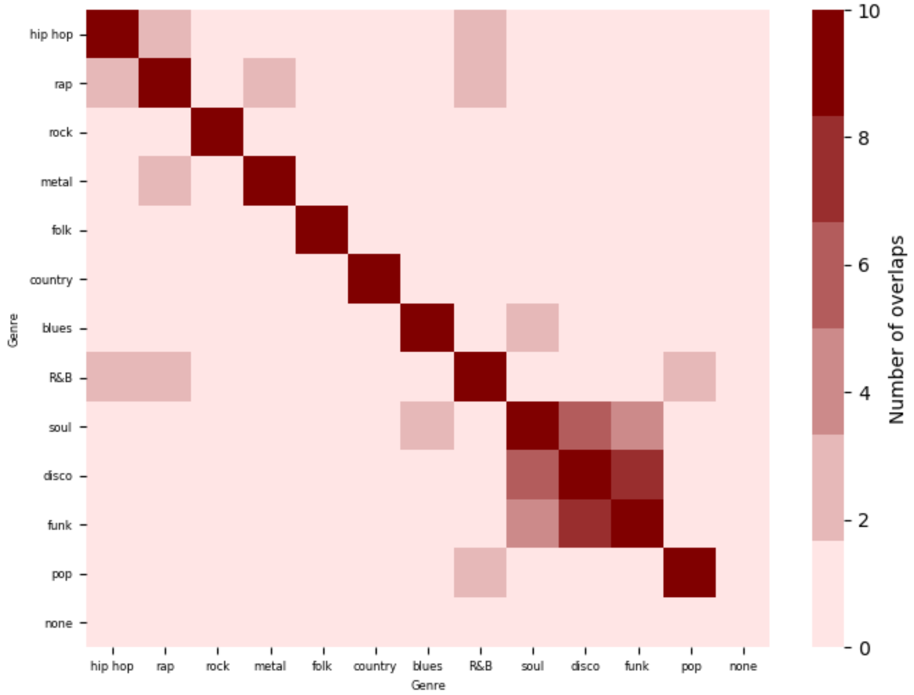


Figure 3.4: Overlap in the 10 top tf-idf song-artist tags between each genre.

Figure 3.4 shows the overlaps as a heatmap for all of our genres. Country is the most distinct with no overlaps; whereas, soul, disco, and funk are closely related to each other with high degrees of overlap. We chose to use separate genres for hip hop and rap because there was not much overlap between their top 10 tags. By increasing the tf-idf set size, one will have more ways to distinguish genres, but the individual differences between genres will shrink as they share more mutual tags.

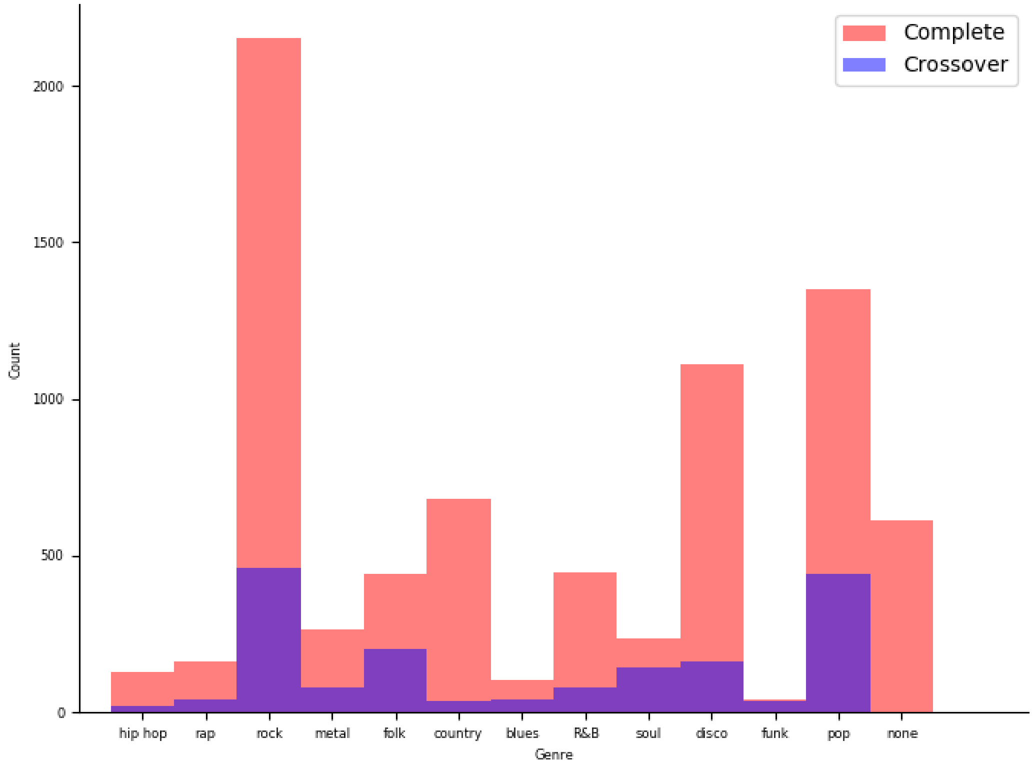


Figure 3.5: Distribution of songs in each genre. (Red) Complete dataset. (Blue) Crossover songs where more than one genre had the most song matches.

For the second method, we labeled each song's genre as the genre with the most overlap between the song's artist tags and the genres' tag sets with ties broken randomly. In the case of ties, the songs were also labelled as crossovers because we found them to be equally associated with two genres based on our tag mapping system. The crossover control also acts as a measure of genre distinctiveness. We would expect that genres with high overlap like funk, soul, and disco are more likely to have crossovers because more of their tags are shared, and there is less to differentiate between them. Let us now revisit the genre example to explain the second method. If a song had the artist tags rock and hard rock, then this method would label the song as a rock song because rock has the most tags matching the song's artist tags. On other hand, if the song had the artist tags rock, hard rock, and metal, then the method would denote the song as a crossover and randomly label it with either the rock or metal genres because both genres have the same number of matching tags.

In Figure 3.5, we can observe the song genre distribution in the complete dataset. The rock, pop, and disco genres make up the largest share of the songs, while the blues and funk genres make up the smallest share. Soul and funk, which have high levels of overlap with disco, have a high proportion of crossovers. It is likely that some of the songs annotated as disco are actually soul or funk songs.

3.4.3 Similarity

Inspired by the use of a typicality measure by Askin & Mauskapf to study how songs perform based on their distinctiveness from other songs also on the charts [1], we approximated their methodology for constructing such a feature and developed our own set of context-based similarity measures for audio and lyric features with temporal weights (they used genre weights). In total, we created 15 measures based on The Echo Nest features and the topic mixtures (four sets for LDA models with 10, 20, 40, and 80 topics) with chart, genre, and artist contexts. Each of the contexts is defined below.

1. Chart similarity: how similar is a song to other songs that have been charting over the previous year?
2. Genre similarity: how similar is a song to other songs from the same genre that have been charting over the previous year?
3. Artist similarity: how similar is a song to past charting songs previously released by the same artist?

All of the similarity measures were computed using the cosine similarity. For topic mixtures, this involved taking the cosine similarity between all of the features, and for The Echo Nest features, this involved taking the cosine similarity between all of the features

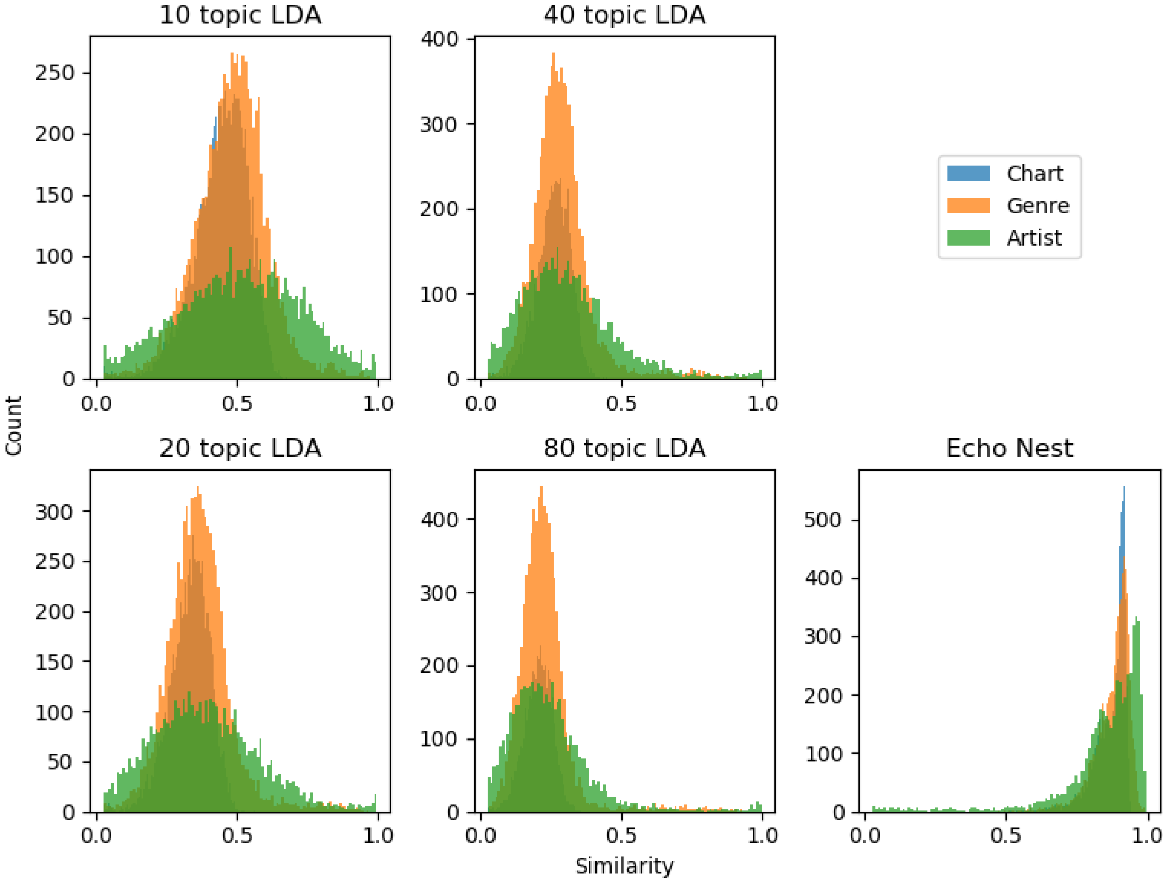


Figure 3.6: Context-based cosine similarity distributions for chart, genre, and artist contexts using LDA topic model mixtures and The Echo Nest features.

except song length. In both cases, all of the features used to calculate the cosine similarity were equally weighted. For a given song, we compared its feature values to all of the other relevant songs and then took its weighted normalized similarity, where the weight was a time decaying weight and the normalization was the sum of similarities over the sum of weight terms. Equation 3.1 shows the weight decay formula for weight w between a song released on some Billboard Hot 100 week and another song released wks weeks before.

$$w = \mathrm{e}^{\frac{-wks}{52}} \tag{3.1}$$

We can observe the similarity distribution for each context measure in Figure 3.6. Note that the histograms have been truncated to not include similarities equal to 0, which is is sometimes the case for artists with smaller catalogues and less popular genres. The topic mixture distributions are normal with different levels of skew based on the context type. Chart contexts have the least skew and artist contexts have the most. As the topic mixture size increases, the associated similarity distributions shift slightly towards 0. On the other hand, The Echo Nest distribution is a left skewed normal distribution with more noise; it is also less smooth. One might be able to argue that the noise is actually a bi-modal distribution at least for the artist and chart contexts. Askin & Mauskapf observed such a distribution when they calculated a similar context-based measure using The Echo Nest features. They attributed the bi-modality to the mode of the songs, which was used in calculating their measure [1].

3.4.4 Control variables

We develop control variables to represent the metadata song features. These are the baseline song values that we are less interested in analyzing but that might still have statistical power for distinguishing song chart trajectories. In The Echo Nest features, we have song length, mode, time signature, and key as controls, but we consider mode, time signature, and key as part of The Echo Nest feature set because we do not overly process them, and so because of this minimization of processing, they are included in calculating The Echo Nest similarity measures while song length is not. Other control variables we have also already covered include genre and crossover. These variables control for the effects of a song targeting a specific demographic or multiple demographics.

There are a few other control variables that we have not yet described outlined in Table 3.4. Like many features in our feature set, these features were also used by Askin & Mauskapf in their study on optimal differentiation [1]. All of these features were derived from the Billboard chart data.

Table 3.4: Billboard control variables summary table.

Attribute	**Scale**	**Definition**	**Base Attribute value**
Time range	1-11	Time in 5 year blocks	Released between 1958 and 1962
Artist success	1-4	Past artist popularity	First hit
Reissue	0-1	Past song activity	No reappearance

First, we have the time blocks, which were briefly mentioned before and are a control for the date of a song's release. Different features may become important through time e.g. a new genre less reliant on instrumentation becomes popular. We also know that the meaning of holding a specific chart position has changed over time because Billboard has adjusted their chart formula over the years for determining how songs are ranked and how long they can stay on the charts.

Next, we have artist success, which is a control for popularity; it is based on past hits. If an artist has had past hits, then listeners will be more familiar with their work, and it might be easier for them to make it onto the charts again. We use four groups: first hit, 1-2 previous hits, 3-9 previous hits, and 10+ previous hits.

Finally, we have reissue, which is a control for a song making multiple appearances on the charts; it is based on a song's chart activity. If a song has had multiple appearances, then listeners are more likely to be familiar with the song or it may signify a change in the promotion strategy for the artist. We define a reissue as a song making an appearance on the charts and then disappearing for at least six months before reappearing.

3.4.5 Summary

In this chapter, we reviewed the steps taken to build our dataset from the retrieval of features using different data sources like the chart data, the BOW lyrics, and The Echo Nest features to the derivation of different feature types like the topic mixtures, genre tags, similarity measures, and miscellaneous controls.

Chapter 4

Target variables for song popularity analysis

There is one more field from our data that gets processed; it is the variable of interest, the thing that we model, the target variable. We want to model song popularity, so the target variable must represent some concept of popularity, but what is song popularity? Is it the number of weeks that a song has lasted on the charts, the top position that a song has reached, or whether or not a song has crossed some chart position threshold? These are all possible definitions with their own tradeoffs around specificity and complexity. Askin & Mauskapf considered two definitions, one for the magnitude of a song's sales, its peak position, and one for the staying power of a song, the number of weeks it was on the charts [1], when they modelled song popularity. Their definitions represent two distinct components of song popularity, magnitude and time.

One limitation of their approach is that they model these components separately when they could have some higher-order interactions. We propose two types of flexible target variables, the interval and alignment target variables, to model these dimensions together. Interval target variables are derived from taking the product of Askin & Mauskapf's definitions of hit songs, whereas, alignment target variables are derived from clustering song chart trajectories with a variable-length distance metric. By utilizing these two types of target variables, we hope to understand how position magnitude and time, as components of a song's success, should be incorporated into definitions of song popularity.

4.1 Interval

We examine three interval target variables based on weeks on chart and peak position. Askin & Mauskapf modelled weeks on chart with a negative binomial regression model as it can be represented by a fixed number of values and peak position with an ordered logit model as it is ordered between 1-100 on the Billboard Hot 100 weekly chart [1]. As one of our goals is to develop target variables for classification, we need to further discretize their target variables to create groups that are easier to classify. First, we divide their target variables into groups based on value where values within a specific range fall into the same class. This allows us to create a merged weeks on a chart target variable and a merged peak position target variable. Next, we take the product of these merged target variables to produce the merged joint target variable, which has a number of classes equal to the number of classes in merged weeks on chart multiplied by the number of classes in merged peak position.

Weeks on chart can be found by retrieving a set of song records from the Billboard Hot 100 chart and finding the maximum value from the records' week field. In the top figure from Figure 4.1, the weeks on chart distribution is shown. It can almost be approximated by a normal distribution except that it is truncated on the left side at week 1, and it has an outlier spike at week 21. The reason for the outlier is because Billboard has a policy of removing songs from the charts after 20 weeks if they become less popular and move above some position cutoff. If such a policy did not exist, then we would expect a distribution with more spread to the right. For merged weeks on chart, we used the following ranges: 1-4, 5-8, 9-12, 13-16, 17-20, and 21+. We chose these intervals because four weeks makes up a month, which is a measurable unit of time one level above weeks, and we wanted all songs lasting more than 20 weeks to be grouped together because of Billboard's removal policy.

Peak position can be found by retrieving a set of song records and finding the minimum value from the records' position field since a song's minimum chart position corresponds to when the song is most popular. In the bottom figure from Figure 4.1, the peak position distribution is shown. It can be closely approximated by an exponential distribution, though this is largely because of the high frequency of songs with peak positions at #1. Like the position distribution, song sales over peak position also follows an exponential distribution, so the difference in sales between songs at positions #1 and #2 is much larger than #99 and #100. For merged peak position, we followed the intervals used by Askin & Mauskapf, which include: 1, 2-5, 6-10, 11-20, 21-40, 41-60, and 61-100. It could be that they chose these intervals because they describe different music sales classes.

By taking the product of the merged weeks and merged peak target variables, we

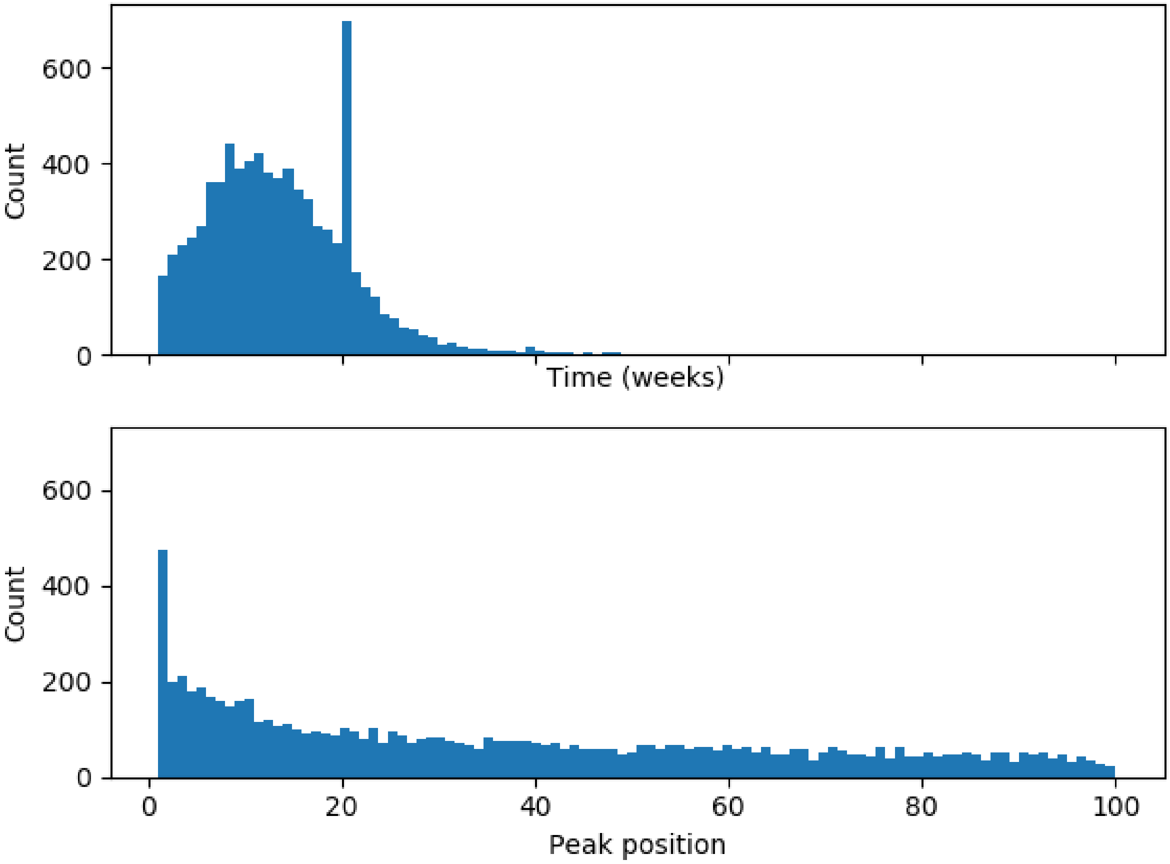


Figure 4.1: Weekly Billboard Hot 100 chart performance target variable distributions from Askin & Mauskapf's work [1]. (Top) Weeks on chart. (Bottom) Peak position.

can consider both time and magnitude in our definition of song popularity. We call this target variable the merged joint target variable, and Figure 4.2 shows its distribution in a mosaic plot. By examining the rows, we can see the distribution of merged weeks, and by examining the columns, we can see the distribution of merged peak.

Some may argue that by modelling magnitude or time, one does not need to model the other because there is a natural relationship at play between the two; songs that last a long time are more likely to have low peak positions, and songs that last a short time are more likely to have high peak positions. We can partially observe the truth of this claim in Figure 4.2 as songs that last 21+ weeks often have peak positions in the top 10 and songs that last 1-4 weeks often have peak positions in the bottom 40; however, in the middle interval classes, while this monotonic relationship between merged weeks and merged peak is still present, there are many songs that do not conform to it, and we do not want to overlook these songs; thus, we developed the merged joint target variable.

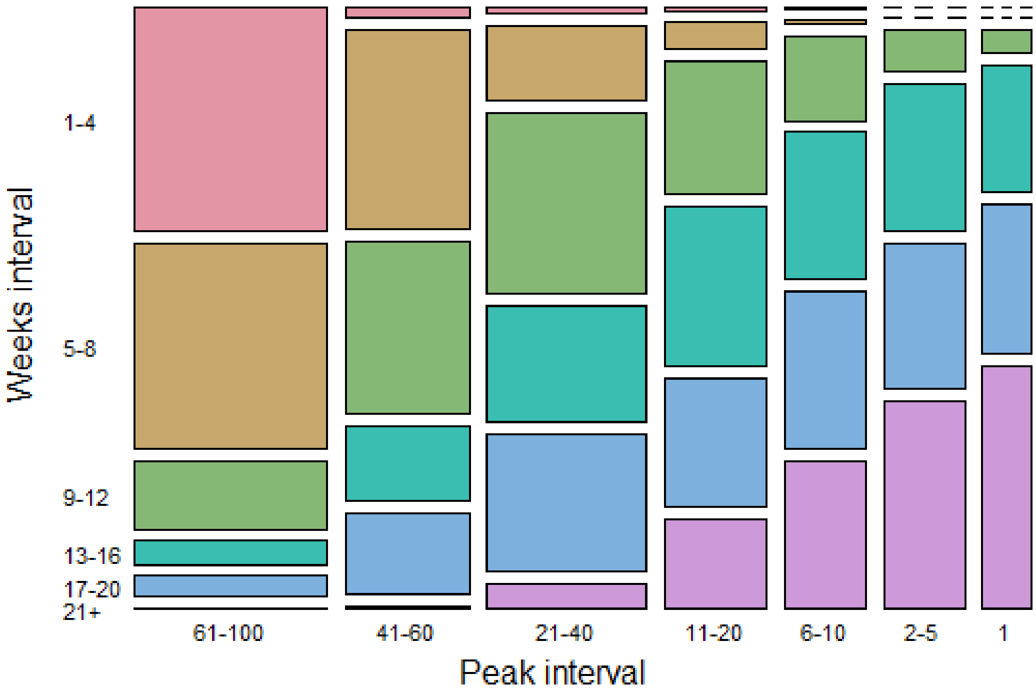


Figure 4.2: Mosaic plot showing the class size distribution for the merged joint target variable. Colors correspond to row values.

To prove that songs from our dataset can be separated based on some measure of song popularity, we must first consider a case of extreme contrast between two classes, modelling the differences between the two most distinct classes. These classes likely correspond to songs with opposite peak positions and weeks on chart values and as a result, opposite merged peak and merged weeks classes. As we observed in Figure 4.2, there exists a trend between peak position and weeks on chart; thus, we expect that of the two most distinct classes, one corresponds to songs with a low peak position and a long chart presence like Kanye West's *Gold Digger* featuring Jamie Foxx, which lasted 39 weeks and peaked at position #1, while the other corresponds to songs with a high peak position and a short presence like De La Soul's *All Good?* featuring Chaka Khan, which lasted three weeks and peaked at position 96.

If we can distinguish these classes of songs by modelling their feature differences in a simple classification task, then this will show that our features have some discriminatory power for distinguishing hits from non-hits, and we can move on to more complex questions about songs from classes that are in between these extremes. One benefit of having so many classes is that if a proposed task is unbalanced, then we can balance it approximately by merging neighboring classes. The downside to this is that the question behind the task becomes more broad as one class has a wider definition.

4.2 Alignment

The interval target variables are based on static measures of song popularity, yet we have dynamic chart data, which changes each week. In order to represent these dynamics, we need to build time series that track the weekly chart variations in position for each song. We would expect that most songs peak somewhere in the middle of their chart lifespan, and so they should have upward arch trajectories where they start at a high position, quickly reach some minimum peak, and then slowly return to a high position before leaving the charts.

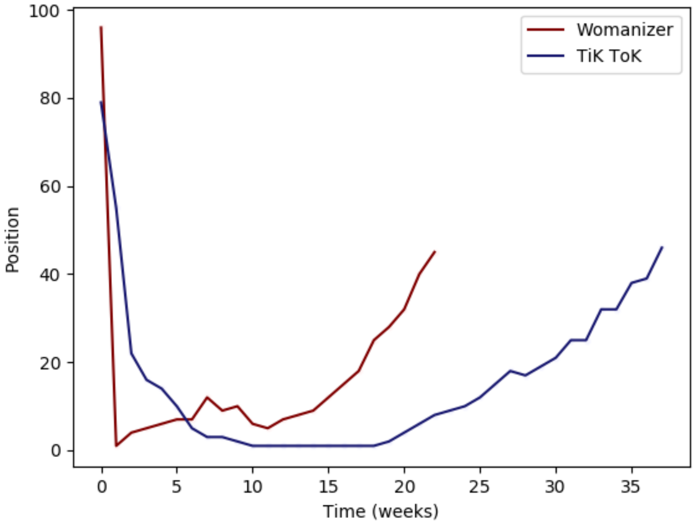


Figure 4.3: Weekly Billboard Hot 100 chart trajectories for Britney Spear's *Womanizer* and Ke$ha's *TiK ToK*.

Now, consider the case where we have two songs from the same genre that both peak at #1 and last for similar amounts of time. One song is released by an artist with millions of followers and the support of a big label, while the other song is released by a relatively unknown artist. One might expect that the former song reaches #1 much faster because of its artist's massive following and label support. An example of two songs from the pop genre that illustrate this contrast would be Britney Spear's *Womanizer*, released in 2008 after she had established a large fan base, and Ke$ha's *TiK ToK*, released in 2009 as her debut single. Figure 4.3 shows their chart trajectories. Both songs lasted over 20 weeks with *Womanizer* reaching #1 in two weeks and staying on the charts for 23 weeks and with *TiK ToK* reaching #1 in 11 weeks and staying on the charts for 35 weeks.

An interval target variable would not be able to distinguish these songs at the level of granularity that we defined since both songs belong to the same merged peak and merged weeks groups, so we developed the alignment target variable to better represent these chart trajectory differences among others. The alignment target variable classes are derived from a hierarchical clustering algorithm applied to chart trajectories that have had alignment distances calculated between them using DTW, which can be considered as an unconventional distance measure. The clustering method has the following workflow:

1. For each song, retrieve its weekly records, sort them by date, extract the position values, place them in a time series, and store the time series into a list.

2. For each pair of time series from the list, calculate the alignment distance between them and store it into a distance matrix.

3. Apply a hierarchical clustering algorithm to the distance matrix to produce a clustering.

4. Use some cutoff threshold on the clustering to retrieve cluster labels. These labels correspond to the song target variable classes.

The cluster labels are used as the target variable classes, and like before, they can be merged; however, they should only be merged if the classes are near each other in the clustering structure. We considered three alignment target variables based on the hierarchical clustering algorithms: single linkage, average linkage, and complete linkage. While we cut the clusterings at different levels, the idea was to produce 40 clusters, so we would have comparable number of classes between the alignment target variable and the merged joint target variable, which has 42 classes.

Some songs have gaps in their chart activity where they disappear from the charts for a number of weeks. We ignore these gaps as it would complicate the DTW algorithm.

Alternatively, one could shorten time series that have a gap longer than some threshold as there are a few songs that reappear on the charts many years after their initial debut. With that said, the vast majority of song gaps are only a few weeks long.

The DTW algorithm uses dynamic programming to align time series by warping their time axes [26]. While there may be some concern around the integrity of the trajectories as their time axes are warped, songs will still be closest to each other if they share the same trajectory occurring around the same time. Songs that follow the same trajectory at different times will be aligned together but they will receive some penalty for the warping that has to take place, and songs that follow different trajectories will not be well aligned, so they will be dissimilar.

4.2.1 Toy alignment example

To further support these claims, we constructed a toy dataset with two sets of time series, which were 10 and 20 time units long, belonging to four patterns. The patterns included flat trajectories, which have no arch, peak trajectories, which have have an upward arch, delay trajectories, which have a peak trajectory arch that has been delayed a few time units, and short trajectories, which have a smaller peak trajectory arch at the same time as the peak trajectory. Figure 4.4 shows all of the time series with the 10 time unit set on top and the 20 time unit set on bottom. The goal of this subsection is to show how DTW compares time series of varying length with different patterns. In this example, we are comparing time series with 10 and 20 time units, and in our real dataset, we have songs that last for a variety of different numbers of weeks on the charts.

DTW is one of the few time series analysis methods that can handle variable-length input. If DTW were used to compare the time series in Figure 4.4, then we would expect that it would find the time series of similar time unit lengths and patterns to be most similar and the time series of different time unit lengths and patterns to be most dissimilar.

In both this toy example and the application of DTW to the real data, we use the DTW implementation from the *dtw* R package [10] with the default parameters to get normalized distances, which means that the algorithm is looking for a global alignment between time series where the first and last positions must be aligned using the Euclidean distance as the local distance function with no windowing and a symmetric step pattern. The normalization counteracts the effects of comparing longer time series with one another as their raw distances are naturally going to be larger.

Figure 4.5 shows two distance metrics applied to the time series data to produce distance matrices; the Euclidean distance matrix is on the left, and the normalized DTW distance

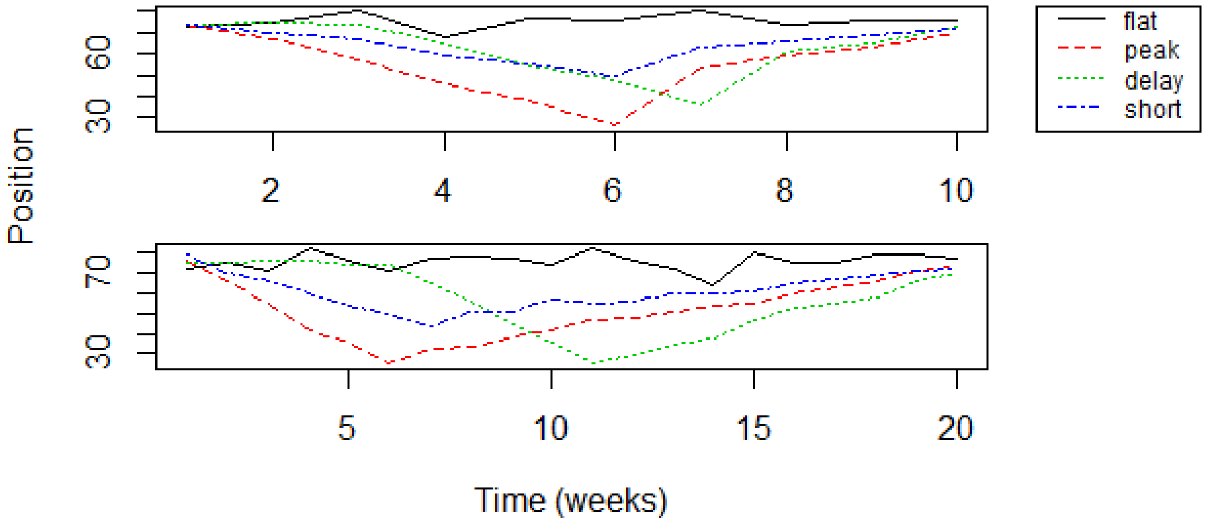


Figure 4.4: Toy time series that follow different trajectory patterns. (Top) 10 time units long. (Bottom) 20 time units long.

matrix is on the right. Note that time series can only be compared with each other using the Euclidean distance if they have the same number of time units.

With the Euclidean distance for both sets of time series, the flat trajectories are furthest away from the other time series, and the peak and delay trajectories are most closely related to the short trajectories. The reason for the difference in distances between sets is because the distances have not been normalized and the longer set has more time unit elements to compare when calculating the Euclidean distance.

With the normalized DTW distance, we can observe that the time series are often closely related to their other length variant even though there is a difference in distance of 10 time units. The flat and short trajectories are most closely related to their other length variants, but the short trajectories are also closely related to the other time series, while the flat trajectories are dissimilar. When the peak and delay trajectories are 20 time units long, they are most closely related, but when they are 10 time units long, their similarity is less unique. The most distinct time series is the flat trajectory with 10 time units; this is likely because when it is aligned to the longer time series, it incurs penalties for expanding the time axis, and it has no peak.

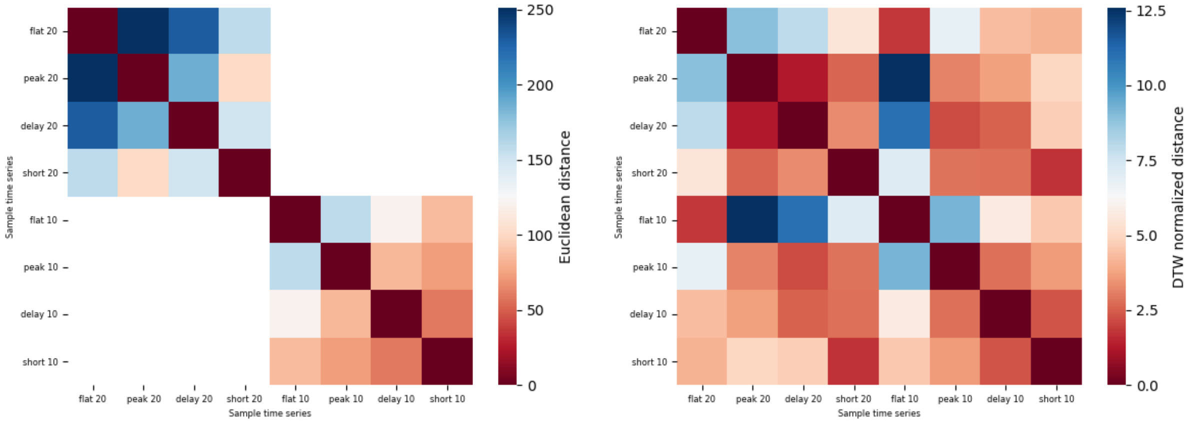


Figure 4.5: Distance heatmaps between toy chart trajectories. (Right) Euclidean distance. (Left) Normalized DTW distance.

4.2.2 Clustering alignments

We have shown that DTW can represent toy examples in a fair manner, so we apply this distance metric to all of the song chart trajectories in the complete dataset to produce a distance matrix. Now, we want to evaluate how that distance matrix can be clustered using three separate clustering algorithms to find a clustering, which is comparable to the merged joint target variable in terms of its class size distributions.

Single linkage merges clusters based on the similarity of their most similar members. Because of this, the clustering can sometimes suffer from a chaining effect where clusters are loosely connected and have no coherent structure. On the other hand, complete linkage merges clusters based on the similarity of their most dissimilar members. While this yields more clusters of equal size, once a cluster accepts an outlier, that outlier determines the cluster's merging pattern at higher levels, which may not represent the other members of the cluster well. Average linkage acts as a compromise between the two other algorithms by merging clusters based on their average distance.

When we apply the clustering algorithms to the DTW distance matrix, we get the cluster size distributions shown in Figure 4.6. The clustering from single linkage suffers from the chaining problem as a majority of songs are in one cluster, and many clusters contain just one song. The clusterings from average linkage and complete linkage are more balanced, but the majority of clusters in average linkage's clustering are small with less than 30 songs. On the other hand, the complete linkage clustering only has 12 clusters with less than 30 songs, which is comparable to the merged joint target variable with its 8 clusters with less than 30 songs. This indicates that their class size distributions are more comparable; thus, we use these target variables in our modelling tasks as they wlll be easier to compare.

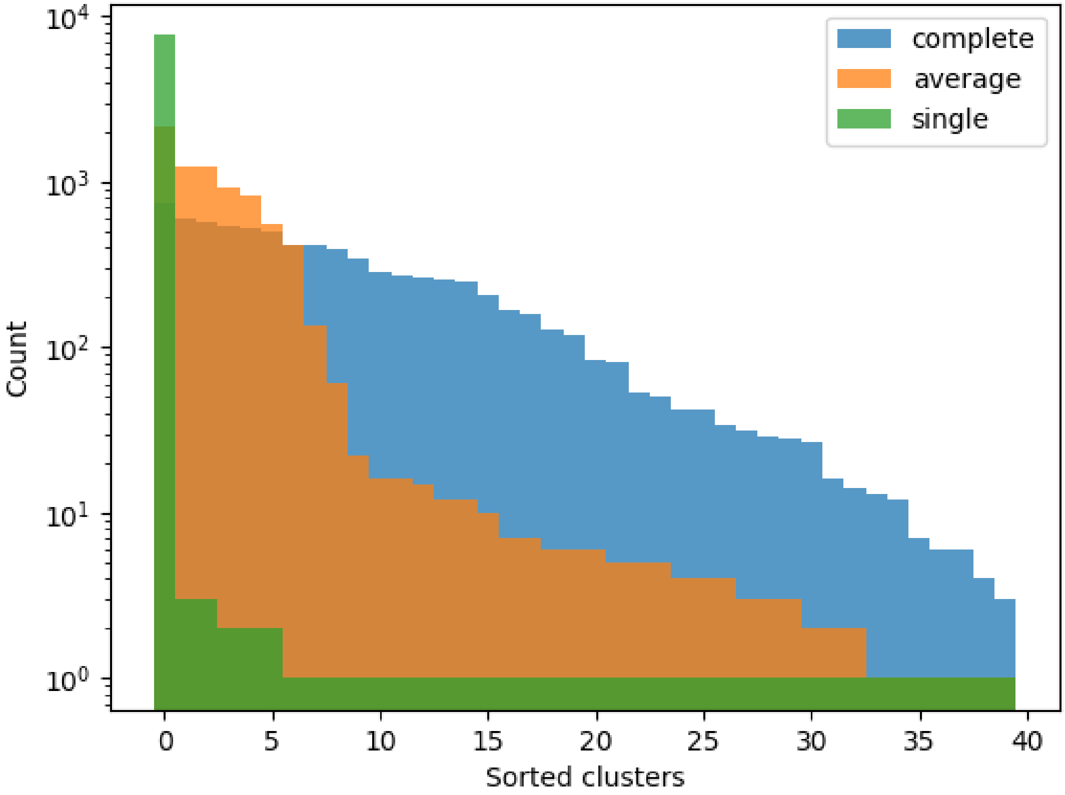


Figure 4.6: Song chart trajectory cluster class size distributions for single linkage, average linkage, and complete linkage clusterings with 40 clusters.

When we apply complete linkage to the DTW distance matrix, we generate a clustering structure, which is shown in Figure 4.7 with clusterings on the left and top sides as dendrograms and a distance matrix comparing the clusterings in the middle. The vertical dendrogram shows the clustering structure with no cutoffs, and the horizontal dendrogram shows the clustering structure cut to yield 40 clusters. The distance matrix allows us to observe how the songs within each cluster are related to each other. Along the diagonal, one can observe more closely related songs forming square groups, which represent some higher level relatedness between songs from clusters adjacent to each other in the dendrograms. On the far left side of the horizontal dendrogram, we have the most distinct grouping of clusters; its clusters are only similar to each other; these include clusters 7, 5, 4, and 8. This dissimilarity with the rest of the data could be because the songs in these clusters have much shorter lifespans.

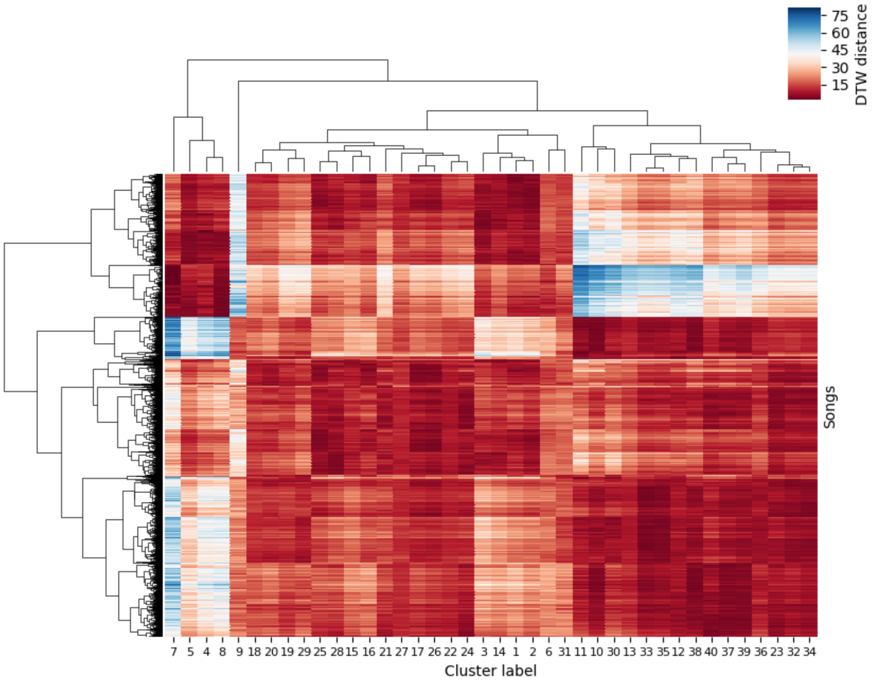


Figure 4.7: Song chart trajectory complete linkage clustering structure as dendrograms organized in a heatmap showing the normalized DTW distances between clusters and songs. (Top) Dendrogram with 40 cluster cutoff. (Left) Dendrogram with no cutoff.

Like with the interval target variable, having so many clusters is an advantage because it gives us the ability to control what we model as we can select specific classes to merge together into higher level groups. Figure 4.8 shows the class size distributions for the 40 clusters produced by complete linkage. Using class size information is also important when building higher level groups as it allows us to construct modelling tasks with balanced class sizes.

As we have chosen complete linkage clustering as the alignment clustering approach,

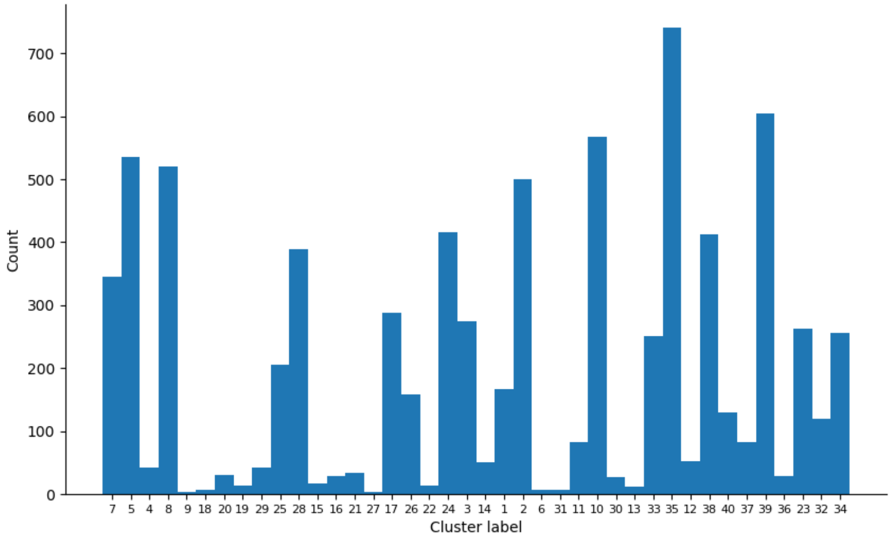


Figure 4.8: Class size distributions for complete linkage clustering with 40 clusters.

we now refer to its target variable classes as the alignment target variable, and we refer to the merged joint target variable as the interval target variable.

4.3 Comparison

Now that we have established our target variables and know that they share similar class size distributions, it is useful to visually compare their trajectories. Figure 4.9 shows the trajectories for the interval target variable classes, and Figure 4.10 shows the trajectories for the alignment target variable classes. The points represent individual weekly observations with overlap measured by the intensity of point colours. The lines represent the average position value through time. Note that here the interval classes are described with numerical labels instead of merged peak and merged weeks categories; classes 1 to 6 refer to songs that have peaked between positions 61-100 and lasted on the charts from 1-4 to 21+ weeks. Each subsequent column refers to a new peak position range.

One noticeable difference between the target variables is that the interval target variable contains more classes with arch-like trajectories, whereas, the alignment target variable has more classes with unusual trajectories like the oscillations in clusters 13 and 32 and the

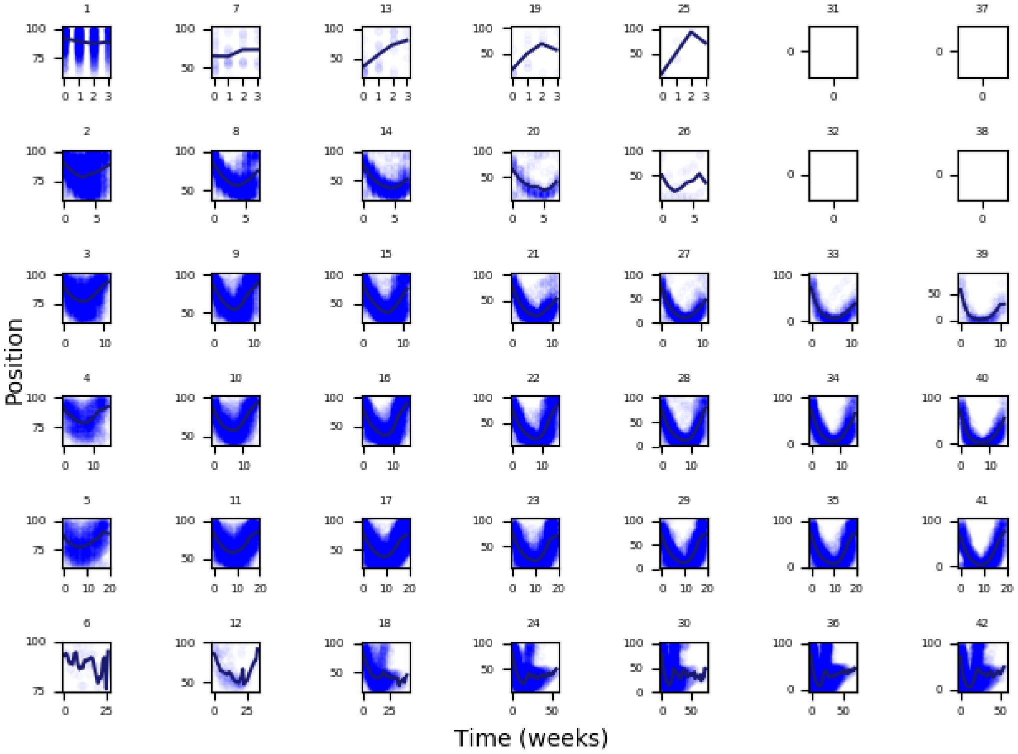


Figure 4.9: Interval target variable song chart trajectory classes.

sharp upward trends in clusters 19, 20, and 29. The alignment target variable can also find classes with arch-like trajectories, but they have less agreement on where the minimum peak position is, so they look less well-defined.

We can also compare the target variables together by finding their closest analogs in the other target variable. We do this to make the comparisons between target variable classes easier to see. In Figure 4.11, we consider a select few alignment target variable classes and find their most closely related interval target variable classes according to tf-idf where the term frequency refers to the frequency of the interval groupings that share the same alignment cluster label and the document frequency refers to the overall frequency of the interval groupings. The target variables are then plotted together, and we comment on their similarities and differences. There are five sample trajectories in Figure 4.11. We can observe how closely related these clusters are by revisiting the horizontal dendrogram from Figure 4.7.

Cluster 7 comes from the leftmost group of clusters and is plotted with the merged

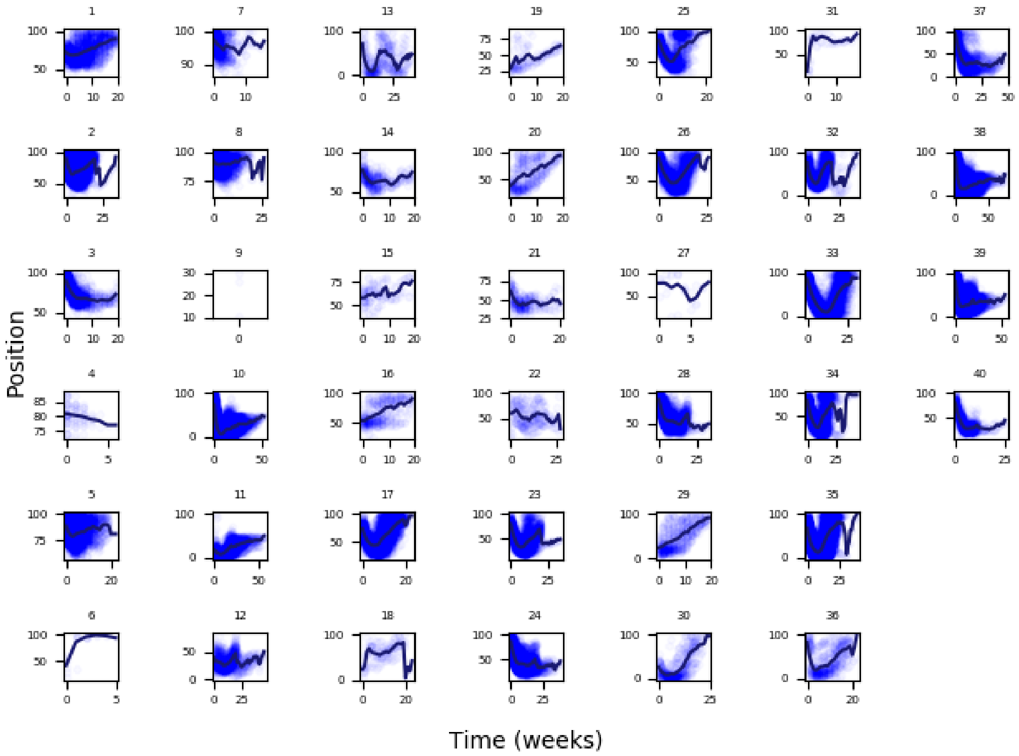


Figure 4.10: Alignment target variable song chart trajectory classes.

group that lasts for 1-4 weeks with a peak between 61-100. These classes refer to flops because they peak very high and do not last long. In both cases, the class trajectories are flat, but the alignment target variable is more varied in size. Clusters 28 and 2 are more closely related in the dendrogram and are located in the center, but they are on opposite ends of a large subtree. Their respective merged groups also share the same peak group between 41-60, but they differ in their lifespans, which are 5-8 and 13-16 weeks respectively. These classes refer to moderate successes because of their mid range peak and weeks values. The interval target variables have more well-defined arch trajectories in both of these classes. Clusters 10 and 34 also share the same subtree on the right side of the dendrogram, but they are also on opposite ends. Their merged joint groups do not share the same weeks or peak values with cluster 10 corresponding to the merged group that lasts for 9-12 weeks with a peak at #1 and cluster 34 corresponding to the merged group that lasts for 17-20 weeks with a peak between 11-20. These classes refer to hits as they have low peaks and last long. There are likely other interval target variables that match cluster 10 better, but most of the songs belonging to that interval cluster are found in the alignment cluster.

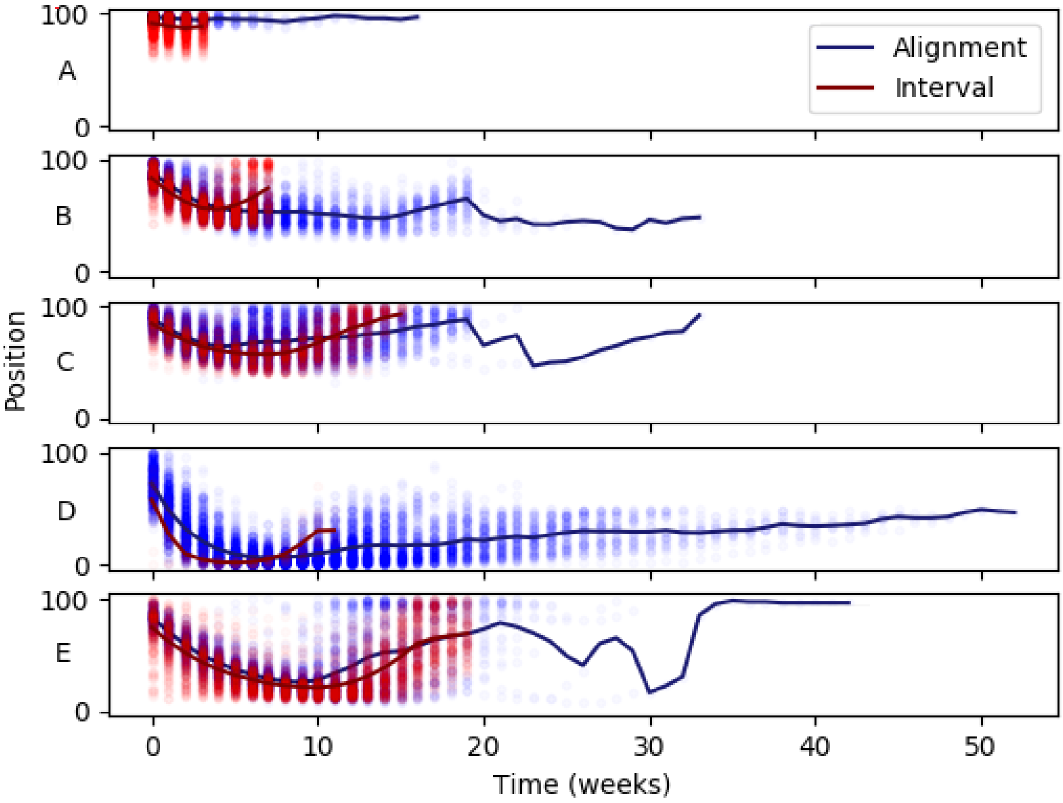


Figure 4.11: Sample time series from alignment and interval target variables. (A) Cluster 7 and group 0, 1-4 weeks and 61-100 peak. (B) Cluster 28 and group 7, 5-8 weeks and 41-60 peak. (C) Cluster 2 and group 9, 13-16 weeks and 41-60 peak. (D) Cluster 10 and group 38, 9-12 weeks and #1 peak. (E) Cluster 34 and group 22, 17-20 weeks and 11-20 peak.

We know that the interval target variables have to follow stricter bounds as all of the songs within a group must share the same range of peak and weeks values. We can observe similar arches between the target variables; however, the alignment target variables have more variability in their arch trajectories. One benefit of the alignment target variables is that they can identify rarer trajectory patterns that are not arch-like.

4.4 Summary

In this chapter, we reviewed a number of potential target variables and settled on an interval target variable, the product of interval groupings for peak position and weeks on chart, and an alignment target variable, the complete linkage clustering of chart trajectories based on their DTW distances. They will be used at different stages in the modelling chapter to help answer questions around the validity of various modelling tasks, how time and peak magnitude should be incorporated, and what relationships exists between specific features and chart trajectory patterns.

Chapter 5

Modelling song popularity

Previously, we gathered and crafted song features and target variables in preparation for modelling song chart behaviour. In this chapter, we perform the modelling and ask a variety of questions to look at relationships between audio and lyric song features and chart behaviour at different levels of granularity. We examine the following questions:

1. Is it possible to model songs as hits and flops?
2. Is there a difference in modelling song popularity using temporal or peak-based definitions of success?
3. Can the song popularity modelling problem be represented as a multiclass classification task?

For the first question, the goal is to prove that this sort of modelling task is possible. We attempt to prove this by showing that the most dissimilar classes from each target variable can be separated using some feature set in a way that is better than random or a baseline feature set. This question is important because its success is a requirement for us to look at multiclass problems. It also allows us to evaluate different feature sets and target variables for use in later models.

In the second question, we only use the interval target variable. This is done because we want to explicitly model hits and flops along one dimension of their chart performance. Recall that the interval target variable is the product of two discretized target variables, merged peak position and merged weeks on chart. The goal here is to justify the use of target variables that have both time and magnitude components by identifying disagreements between models tasked with distinguishing hits and flops based on separate time or magnitude definitions of performance.

Finally, for the third question, we model multiple classes of song chart behaviour in a multiclass classification task to illustrate how the feature sets and target variables can be incorporated into more complex questions. The aim of this task is to push the MIR field towards looking at more complex questions beyond the two or three class modelling paradigm commonly used in hit song science. Another aim of this section is to find explicit links between features and chart trajectory patterns.

5.1 Hit or flop?

As was just mentioned, we first consider a simple question: can we model songs into two classes based on their performance? We attempt to do this by taking a subset of songs from our data, representing them as hits and flops based on their target variable labels, and then modelling their feature differences. This question has been examined many times over the years, yet it still remains controversial in MIR [1, 3, 21, 24, 15]. Its advocates [3, 15] argue that modern audio and lyric features are sufficient to train a hit song detection model while its detractors [24] argue that past models have relied too heavily on unrepresentative data and that there is too much complexity in the real world to learn anything that can be applied to songs outside of existing datasets. We cautiously approach this question intending to develop a model that looks at the historical differences between hits and flops from the Billboard Hot 100 chart. We do not make any claims about using our model to predict chart behaviour in songs outside of our dataset; however, we hope that our findings are at least applicable to other Billboard songs released over the same time period. Across feature sets, we have metadata, audio, and lyric features. We are most interested in the relationships between chart behaviour and the audio and lyric features because as listeners, we primarily evaluate songs based on their audio and lyric content. The metadata features are less important because they are the baseline attributes of a song, which are also less likely to have been actively chosen by an artist. For our models, the metadata features primarily act as control variables for things like song genre and date of release. Also with this first question, we want to determine which target variable-feature set pairing is optimal for use in later models, and in order to do this, we need to first find the most separable classes resembling hits and flops for each target variable.

For the interval target variable, one can imagine that the most distinct classes will be on opposite ends of the spectrum for both interval dimensions; thus, in Figure 4.2, these classes will be on opposite diagonal ends of the mosaic plot, and the difference in their average song weeks on chart and peak position values will be quite large. As there are very few songs in the bottom left and top right classes, it becomes clear that the classes we must compare have to reside in the top left and bottom right corners of the mosaic plot. We refer to the top left class as the flop class because it corresponds to songs that last 1-4 weeks and peak at positions 61-100 and the bottom right class as the hit class because it corresponds to songs that last 21+ weeks and peak at #1. As the hit class is much smaller, basic models will more generally favour the larger class, so to counter this effect, we crudely balance the hits and flops by pooling the hit class with its neighboring classes, so it has an expanded definition, which includes songs that last 17+ weeks and peak at positions 1-5; as a result, there are 855 hits and 783 flops.

For the alignment target variable, we want to find a similar contrast between classes of songs with widely different peak position and weeks on chart values; however, as classes do not explicitly correspond to specific chart performance metrics, it can be less clear which classes to compare. Alternatively, we can use the horizontal dendrogram from Figure 4.7 to find distinct classes because it shows how closely related classes are to each other. Previously, we identified the leftmost subtree in the dendrogram as being most distinct, containing classes 7, 5, 4, and 8. The classes this clade is furthest from are classes 10, 11, and 30. In Figure 4.10, we can observe that classes 7, 5, 4, and 8 correspond to songs with short chart lifespans and high peak positions while classes 10, 11, and 30 correspond to songs with long lifespans and low peak positions; thus, it becomes clear that these two sets of classes resemble hit and flop classes similar to those described for the interval target variable. For this task, we model songs from classes 10, 11, and 30 as hits and songs from classes 5 and 4 as flops. Classes 7 and 8 are excluded from the flop class to keep the class sizes balanced, and so there are 676 hits and 575 flops.

There are some differences between the classes across each target variable, and we can observe them by plotting the target variables' aggregate chart trajectories for their hit and flop classes. In Figure 5.1 on top, we have the interval target variable class trajectories, and on bottom, we have the alignment target variable class trajectories. The interval target variable's flop class contains songs with much shorter lifespans than the alignment target variable's flop class, and its hit class contains songs with two types of trajectories: those that sharply fall off the charts just before week 20 and those that reach the Billboard position cutoff, position 50, after lasting 20+ weeks and then get removed, while the alignment target variable's hit class contains mostly songs that follow just the latter type. We expect different modelling outcomes from the target variables because of the differences in their trajectories; and we predict that the interval target variable will be easier to model because its flop class is much more constrained along the time axis.

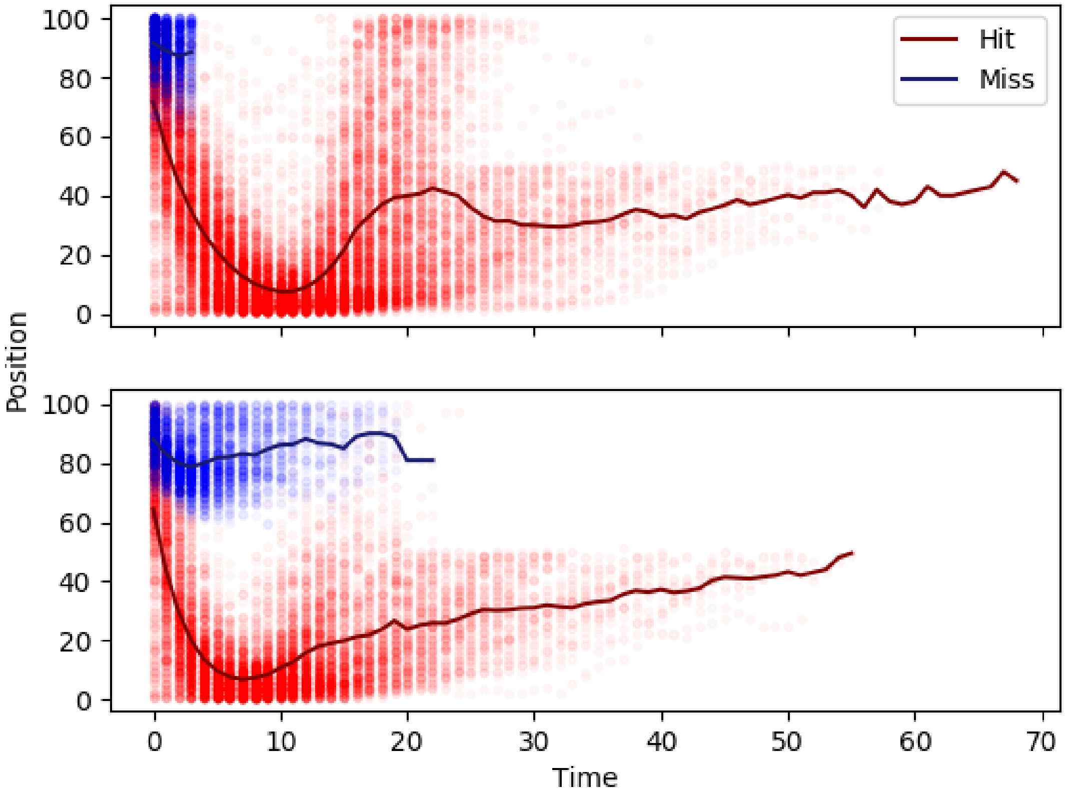


Figure 5.1: Binary task song trajectories for hit and flop classes. (Top) Interval target variable. (Bottom) Alignment target variable.

Along with evaluating target variables, we also compare feature sets of varying size for use in later tasks. The main question here is determining how many topic mixtures from the topic model, LDA, should be included in the ideal feature set. We know that larger topic models are better able to represent documents in a corpus, so we would like to see if this principle holds for representing hits and flops. We compare the following feature sets across both target variables using logistic regression:

1. Baseline features
2. Baseline, The Echo Nest, and similarity features

3. Baseline, The Echo Nest, similarity, and LDA with 10 topic mixture features
4. Baseline, The Echo Nest, similarity, and LDA with 20 topic mixture features
5. Baseline, The Echo Nest, similarity, and LDA with 40 topic mixture features
6. Baseline, The Echo Nest, similarity, and LDA with 80 topic mixture features

It should be noted that some of our features can be highly correlated with each other such that they can be represented as combinations of each other. This phenomenon is known as multicolinearity and is something that we want to avoid as it negatively affects model convergence. This is why one topic mixture from each feature set must always be excluded as the mixtures sum to one; across each feature set we exclude topic 0. For the other continuous features, we compute their variance inflation factor (VIF) from the R package *usdm* [23] and exclude highly correlated features with VIF scores above 4. The VIF score is a ratio of variances between two models, one with all of the features and the other with only the feature of interest.

When evaluating feature sets, we first fit a logistic regression to the data and look at the model's AIC, which is a measure of model quality that penalizes having too many parameters; lower AIC scores indicate better fits. AIC is asymptotically equivalent to leave-one-out cross-validation, so we expect that the model with the lowest AIC will have the best performance. We verify this through 10-fold cross-validation where we compute the average accuracy, precision, recall, and F1 score for each model. AIC scales with sample size, so it cannot be directly used to compare models built from datasets of different sizes. When comparing target variables, we address this issue by dividing the models' AIC scores by their respective sample sizes [9, equation 7.27].

5.1.1 Hit or flop model fit

First, we constructed logistic regression models for each target variable-feature set pair. Table 5.1 shows the summary tables for the interval models, and Table 5.2 shows the summary tables for the alignment models. We can observe that the alignment models have lower AIC scores; however, this is likely due to their smaller sample sizes. The interval model with the lowest AIC score is built from the feature set with 20 topics. Its performance is marginally better than the models built with the 10 topic and The Echo Nest feature sets. The alignment model with the lowest AIC score is built from the feature set with The Echo Nest features. The models built from the control, 10 topic, and 20 topic feature sets have modestly worse AIC scores. What these scores indicate is that for the interval target variable, lyric topic mixture features can be helpful for distinguishing the hit and flop classes as defined in this binary task, and for the alignment target variable, these features' effects are negligible. We can also observe that the larger topic mixture feature sets did not fare well under the AIC metric; however, their negative log-likelihood scores were the highest of all models, which suggests that they are able to better represent songs as hits and flops, but based on the AIC scores, their improvements in class representation were not sufficient to justify feature set expansion. We can compare the target variables' separability by taking the ratio of their model AIC scores and sample sizes, and when we do this, we find that the task defined with the alignment target variable is more separable as shown in Table 5.3. Across each feature set, the alignment target variable yields a lower AIC ratio. This goes against what we initially hypothesized given the class trajectories from Figure 5.1, which showed a constrained interval target variable flop class, though it may be that its hit class was too broadly defined with a lot of variability across many song features, which resulted in a poorer separation.

Across the interval models from Table 5.1, many of the same control variables have statistically significant effects. Take for example the reissue control, which has a significant and large effect in each feature set. What could justify this control variable having such an important role in the task? One possibility is that because flops only last 1-4 weeks and reissued songs must have a six month gap between two chart appearances, the reissued songs are much more likely to be hits; thus, the reissue feature almost becomes an indicator for hit songs. If it were a perfect indicator, then it would need to be removed from the feature set because it would negatively affect model convergence. Other features with unbalanced class distributions can also be detected by looking at the magnitude and significance of their coefficients. We are most concerned when these features are control variables as it may mean that the model overlooks the discriminatory power of other features of interest like the audio and lyric features.

Similar to the control metadata features, the continuous The Echo Nest features also had significant effects. From the sample table in Table 5.1, we can reason that songs that are more acoustic, danceable, and with negative (valence) are more likely to be hits. On the other hand, the control The Echo Nest features had weaker, less significant effects as shown in Table C.1. Similarly, most of the similarity features did not have much influence except for the genre topic mixture similarity. Previous works have found how a context-based similarity measure can improve models [1, 3]; however, it may be that there is only room for one similarity measure as the measures we developed, namely the chart and artist similarity measures, are closely related and may be overly redundant.

The topic mixtures played a role in contributing to some of the models; however, their effects were generally specific to an individual feature set, and most topic mixtures did not have significant effects. Before we explore some of the individual topics that contribute to the models, it is important to note that topics are not directly comparable across models. Firstly, topics with the same label in separate models do not refer to the same topic, so topic 1 across each model can refer to different themes. Secondly, as was shown in Table 3.3, even if two topics share the same top words in separate models, the topic from the larger model will have more specificity, so the topics will not be exactly the same. Even so, we still attempt to identify topics derived from separate topic models that share the same top words as this indicates some level of consistency in the topics' effect. Interestingly, even though the largest feature set was deemed weak by the AIC metric, it contained a number of obscure topics with significant effects not found in other feature sets.

Below are some sample topics from the feature sets. One topic from Table 5.1 related to dancing had a significant positive effect in $T_{10,2}$ and $T_{20,3}$; the topics shared four of the five most probable stemmed words: hey, danc, rock, and shake. We also know that the audio feature danceability had a positive effect in both feature sets, so there could be a relationship between these audio and lyric features. One of the obscure topics we observed with a significant negative effect was $T_{80,78}$, which referred to black power. Another topic with a significant positive effect referring to musical passion was present in $T_{20,11}$. Two words from this topic stood out: fire and burn. What is interesting about these words is that they stick together in each topic model feature set, but instead of the a fine-grained narrowing of the topic as shown in Table 3.3, the topics containing these words drastically shift meanings from one topic model to the next. In $T_{40,20}$, the topic refers to nature and religion, and in $T_{80,11}$, the topic refers to passion and survival. Like the complete summary tables for each feature set, the top five words for each topic in each topic model can be found in Table B.1.

Table 5.1: Interval target variable logistic regression models for various feature sets.

	Dependent variable:					
	Interval target variable					
	(metadata)	(The Echo Nest)	(10 topic)	(20 topic)	(40 topic)	(80 topic)
10+ hits	−0.629***	−0.522**	−0.614***	−0.569**	−0.523*	−0.486*
	(0.195)	(0.252)	(0.232)	(0.263)	(0.267)	(0.278)
reissue	6.729***	6.890***	6.777***	7.229***	7.066***	7.631***
	(1.349)	(1.346)	(1.367)	(1.443)	(1.452)	(1.542)
genres						
rap	1.136**	1.282**	1.521**	1.795***	1.958***	1.928***
	(0.565)	(0.597)	(0.620)	(0.643)	(0.656)	(0.669)
country	−0.358	0.133	−0.027	−0.280	−0.434	0.029
	(0.500)	(0.563)	(0.604)	(0.633)	(0.657)	(0.674)
pop	1.088**	1.773***	1.459***	1.084*	1.104*	1.611***
	(0.444)	(0.502)	(0.545)	(0.579)	(0.609)	(0.620)
time blocks						
1988-1992	1.078***	1.140***	1.101***	1.217***	1.228***	1.213***
	(0.278)	(0.343)	(0.354)	(0.359)	(0.366)	(0.381)
1993-1997	0.441	0.423	0.540	0.625	0.653	0.565
	(0.324)	(0.389)	(0.405)	(0.413)	(0.423)	(0.437)
The Echo Nest similarity						
genre similarity		−1.530*	−0.956	−0.751	−0.500	−1.259
		(0.925)	(1.013)	(1.022)	(1.039)	(1.047)
The Echo Nest features						
acousticness		0.760**	0.832**	0.986***	0.921***	0.912**
		(0.340)	(0.345)	(0.353)	(0.357)	(0.368)
danceability		3.593***	3.646***	3.748***	3.772***	4.050***
		(0.588)	(0.606)	(0.616)	(0.630)	(0.656)
valence		−1.448***	−1.531***	−1.624***	−1.538***	−1.582***
		(0.380)	(0.388)	(0.393)	(0.402)	(0.419)
topic mixture similarity						
genre similarity			−1.180	−2.690***	−3.074**	−1.399
			(0.814)	(1.007)	(1.209)	(1.401)
topic mixtures						
X2			1.780**			
			(0.763)			
X3				1.776*		
				(1.018)		
X11				3.041**		5.031*
				(1.208)		(2.934)
X20					1.215	
					(1.550)	
X78						−9.277**
						(4.022)
Observations	1,638	1,638	1,638	1,638	1,638	1,638
Log Likelihood	-873.199	-817.253	-806.656	-791.428	-781.379	-752.163
AIC	1,804.399	1,738.506	1,739.311	1,730.857	1,750.758	1,770.326
Note:						*p<0.1; **p<0.05; ***p<0.01

One area of contrast between the alignment and interval models is that the alignment models have fewer metadata features with significant effects. Consider the reissue or the 10+ hits control variables. In Table 5.1, reissue had a strong positive effect, and the 10+

hits control had a weaker negative effect; both were significant. In Table 5.2, the reissue feature has neither a large coefficient nor a significant effect, and the 10+ hits control has an effect in the opposite direction with less significance. Songs by artists who have had 10+ past hits are now more likely to be hits, whereas, they were previously less likely. These examples indicate two things: first, the alignment feature set is more balanced across most metadata features, and second, feature effects can be very dataset dependent even if one assumes that tasks have similar classes. The one exception to the alignment metadata features being weaker was the time blocks. Their effects were much stronger and significant in the alignment models, which indicates that there is not as much overlap between when their hits and flops were released. With these changes in feature values, it can be challenging to identify broader relationships between features and tasks especially if the tasks have different imbalances, but by building larger datasets with less missing data, we predict that the impact of these imbalances can be reduced.

In Table 5.2, The Echo Nest features still have somewhat significant effects. We see the same relationship as before with danceability and valence. The genre topic mixture similarity is also important for a few tasks and has a negative effect indicating that the less unique a song's features, the more likely it is that the song will not be a hit. The significant topic mixture effects in the alignment models were not consistent with those highlighted by the interval models; however, there was consistency in one topic's effect across feature sets; the topic played on themes around baby talk. It was present in $T_{10,5}$, $T_{20,13}$, $T_{40,39}$, and $T_{80,17}$. This consistency offers strong evidence that this topic is important to hits; furthermore, baby talk usage has fluctuated over the years with the rise and fall of different genres, which could explain the stronger time block effects. Baby talk, love, and romance all fall under the same category of singing about someone you admire, but the topics that more directly referred to love did not generally fair well; their effects usually were not significant, but we observed one exception, again in the 80 topic model feature set, $T_{80,57}$, which referred to falling in love and had a significant positive effect.

Table 5.2: Alignment target variable logistic regression models for various feature sets.

	Dependent variable:					
	Alignment target variable					
	(metadata)	(The Echo Nest)	(10 topic)	(20 topic)	(40 topic)	(80 topic)
10+ hits	0.477^{**}	0.723^{**}	0.674^{**}	0.646^{*}	0.532	0.573
	(0.236)	(0.315)	(0.283)	(0.331)	(0.344)	(0.357)
reissue	0.039	0.165	0.339	0.118	0.092	0.308
	(1.102)	(1.164)	(1.182)	(1.226)	(1.191)	(1.111)
genres						
rap	1.721^{**}	2.128^{**}	2.014^{**}	2.119^{**}	2.115^{**}	2.312^{***}
	(0.781)	(0.835)	(0.850)	(0.856)	(0.864)	(0.897)
country	-1.481^{**}	-0.938	-1.677^{*}	-1.931^{**}	-2.385^{**}	-2.286^{**}
	(0.712)	(0.812)	(0.863)	(0.893)	(0.932)	(0.980)
pop	1.585^{**}	2.434^{***}	1.613^{**}	1.255	1.049	1.247
	(0.665)	(0.770)	(0.819)	(0.860)	(0.901)	(0.951)
time blocks						
1988-1992	-3.095^{***}	-3.415^{***}	-3.467^{***}	-3.422^{***}	-3.401^{***}	-3.551^{***}
	(0.396)	(0.441)	(0.453)	(0.458)	(0.468)	(0.493)
1993-1997	-1.712^{***}	-1.964^{***}	-1.987^{***}	-1.858^{***}	-1.699^{***}	-1.862^{***}
	(0.366)	(0.416)	(0.434)	(0.444)	(0.453)	(0.472)
The Echo Nest similarity						
genre similarity		-2.223^{*}	-1.432	-1.229	-1.544	-1.628
		(1.152)	(1.202)	(1.215)	(1.280)	(1.327)
The Echo Nest features						
acousticness		-0.133	-0.179	-0.099	-0.026	-0.007
		(0.374)	(0.379)	(0.387)	(0.396)	(0.415)
danceability		2.266^{***}	2.209^{***}	2.283^{***}	2.309^{***}	2.105^{***}
		(0.684)	(0.703)	(0.725)	(0.735)	(0.771)
valence		-0.823^{*}	-1.076^{**}	-1.176^{**}	-1.025^{**}	-0.975^{*}
		(0.468)	(0.481)	(0.491)	(0.500)	(0.519)
topic mixture similarity						
genre similarity			-2.096^{**}	-3.039^{***}	-2.886^{**}	-2.872^{*}
			(0.927)	(1.151)	(1.424)	(1.608)
topic mixtures						
X5			2.430^{***}			
			(0.787)			
X13				2.824^{**}		
				(1.369)		
X17						9.471^{***}
						(2.756)
X39					4.957^{**}	
					(2.133)	
X57						9.659^{***}
						(3.399)
Observations	1,251	1,251	1,251	1,251	1,251	1,251
Log Likelihood	-610.513	-584.974	-574.620	-564.524	-554.912	-537.279
AIC	1,277.025	1,271.947	1,273.241	1,275.047	1,295.824	1,338.558
Note:						*p<0.1; **p<0.05; ***p<0.01

So far, what we have observed is that features can have different effects even in datasets that are perceived to be closely related. This is in part because each task-specific dataset has its own set of unbalanced features, which affect what the logistic regression model can find. The imbalances in the two binary tasks' control variables, which include reissue in the interval task and time blocks in the alignment task, influenced what songs the models were able to distinguish and may have masked the importance of other features. Still, we were able to observe significant effects for some features of interest including The Echo Nest features, the genre topic mixture similarity, and some topic mixtures. We also saw how the alignment models had lower AIC ratios, indicating that the alignment task data had a better separation.

Table 5.3: Ratio between AIC scores and task sample sizes for each feature set to produce an unbiased metric for evaluating target variable separability.

	metadata	**The Echo Nest**	**10 topic**	**20 topic**	**40 topic**	**80 topic**
interval	1.102	1.061	1.062	1.057	1.069	1.080
alignment	1.021	1.017	1.018	1.019	1.036	1.070

5.1.2 Hit or flop model performance

AIC offers a quick way to evaluate models based on fit. A more rigorous evaluation can be performed by training a model on some subset of the data and then testing its predictive ability on another subset. We follow this approach when we employ 10-fold cross-validation to test logistic regression's predictive abilities on our dataset. 10-fold cross-validation works by averaging the results of a dataset being partitioned into ten folds with nine folds used to train a model and the other fold used to test the model where each fold takes a turn as the test fold.

Logistic regression models predict the probability of some observation belonging to a class, in this case, hit or flop, and by thresholding the probability scores, one can give each observation a class label. We select a threshold that maximizes the F1 score because it balances considerations towards model specificity and sensitivity. Along with the F1 score, we also compute the model's precision, recall, and accuracy.

Table 5.4 shows the predictive results from the interval models. The 20 topic model feature set produced the highest F1 score, which is consistent with its AIC score; however, the difference in model performance across feature sets is only 2% for each metric, which suggests that most features do not offer much useful information. In the table, precision scores are much higher than accuracy and recall scores indicating that each model finds more false negatives, hits as flops, than false positives, flops as hits. In general, what this table shows us is that AIC is a good proxy for evaluating model performance as the order of models based on AIC scores is the same as the reversed order based on F1 scores.

Table 5.4: 10-fold cross-validation results for logistic regression models fit to interval target variable feature sets.

	metadata	**The Echo Nest**	**10 topic**	**20 topic**	**40 topic**	**80 topic**
accuracy	0.717	0.722	0.723	0.737	0.720	0.723
precision	0.896	0.932	0.893	0.905	0.917	0.908
recall	0.677	0.671	0.687	0.692	0.673	0.680
F1	0.769	0.778	0.772	0.783	0.774	0.775

Table 5.5 shows the predictive results for the alignment models. Like in the interval models, the order determined by AIC is preserved by the cross-validated F1 scores. The Echo Nest feature set performs best while the 80 topic feature set performs worst. Precision scores are again higher than recall and accuracy scores here. One difference between the target variables is that the alignment models have a larger deviation range between their best and worst models; notice how recall is highest in the 10 topic feature set at 75.2% and lowest in the 80 topic feature set at 70.1%. This may signify that the features are more important in the alignment models than in the interval models.

Table 5.5: 10-fold cross-validation results for logistic regression models fit to alignment target variable feature sets.

	metadata	**The Echo Nest**	**10 topic**	**20 topic**	**40 topic**	**80 topic**
accuracy	0.764	0.779	0.779	0.772	0.761	0.734
precision	0.922	0.910	0.896	0.895	0.894	0.892
recall	0.721	0.742	0.753	0.744	0.731	0.701
F1	0.809	0.816	0.815	0.811	0.802	0.783

These results show us how AIC is a good proxy for model performance, but conducting the modelling ourselves with 10-fold cross-validation is still important as it offers us an opportunity to observe the predictive abilities of each model across a range of metrics. For both target variables, we did not observe a wide difference in the feature sets' model performance, but we saw more variability in the alignment models, which indicates that the features in those models might contribute more to distinguish classes.

5.1.3 Time block control

Across each feature set in Table 5.1 and Table 5.2, we can observe that many of the time block controls have significant effects, and some of these effects are quite large, which may suggest that for the binary task-defined datasets, there is a target variable class imbalance over these control variables. Because of the size of their coefficients, we can gauge how important they are for distinguishing hits from flops in each of the models. More generally, an imbalance in one's data is not necessarily a bad thing; if some feature value is highly associated with some target variable label, then one is able to learn about the link between the two. In our case, if the feature is an audio or lyric feature, then it is useful because we have learned something about an artistic choice in a song and its relation to the song's success. If the feature is a metadata feature, then it is less interesting because the metadata features are not generally chosen by an artist, so the imbalance only tells us about the distribution from which our data was drawn. In this subsection, we look at the time block control to see what information we can gather from its imbalance; we assume that this analysis could be replicated for other control variables as well.

If we look at the time block effects in each table, we can see that the alignment models have larger coefficients, though both target variable feature sets have significant effects for many of their time blocks. Given the time block controls' size and significance, we decided to examine both target variable distributions over the control. In Figure 5.2, we used mosaic plots to visualize the distributions with time block categories on the x-axis, which correspond to 5 year release intervals starting in 1958, and target variable classes on the y-axis with color also corresponding to class. Note that interval target variable classes 1 to 6 correspond to songs that have peaked between positions 61-100 with class 1 being for songs that last 1-4 weeks and class 6 being for songs that last 21+ weeks. Each subsequent set of six classes corresponds to another peak interval range.

From Figure 5.2, one can see the interval target variable-time blocks distribution on the right and the alignment target variable-time blocks distribution on the left. While most of the interval target variable classes are present in each time block, some of the alignment target variable classes are very imbalanced and disappear after a specific time block. In particular, the alignment classes 40, 39, 38, 37, and 10 are present in the early time blocks but disappear after some time only to make a diminished return later; they are replaced by the classes 35, 34, 33, and 32, which are absent in the beginning but make a strong appearance a few time blocks in only to disappear again some time later.

One could reason that this clear imbalance in the target variable explains why the alignment models have such large time block effects. Also, recall that the alignment target variable has a 101 song difference between the number of hits and flops over 1,251 songs, whereas, the interval target variable has a 72 song difference over 1,638 songs; thus, it is reasonable to assume that the alignment target variable will be more prone to the time block effects because it is less balanced over fewer songs.

From Figure 5.2 on the right, we can observe a number of unbalanced alignment target

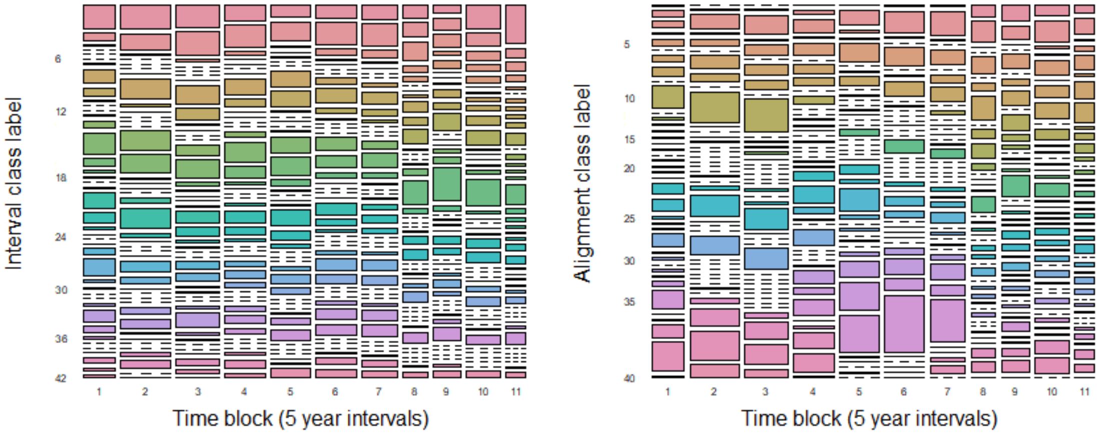


Figure 5.2: Mosaic plots showing target variable distributions over time blocks. (Right) Balanced interval target variable. (Left) Unbalanced alignment target variable.

variable classes. 40, 39, 38, 37, and 10 are all present early on but are replaced by classes 35, 34, 33, and 32 after some time block. From Figure 4.10, we can observe that all of these classes have low peak positions indicating that they correspond to hit classes. By plotting their trajectories, we can better understand what changed with hit songs. These periods were marked by different musical genres; one typically associates the 1960s with the blues, folk, and rock and the 1980s with disco.

Figure 5.3 shows the trajectories for the two sets of classes from separate periods. Both sets share similar peak positions, but only the later period songs have an arch trajectory. The early period songs seem to disappear much quicker after peaking, which might suggest that they were not marketed as heavily. Interestingly, the end of the second period coincides with Billboard's introduction of a policy to remove songs after 20 weeks on the chart if they fell below position 50. While the number of songs with varied peak positions and lifespans remains consistent in our dataset, the trajectories that they follow have changed over time; thus, in order to model trajectories, one must include time blocks as a control feature.

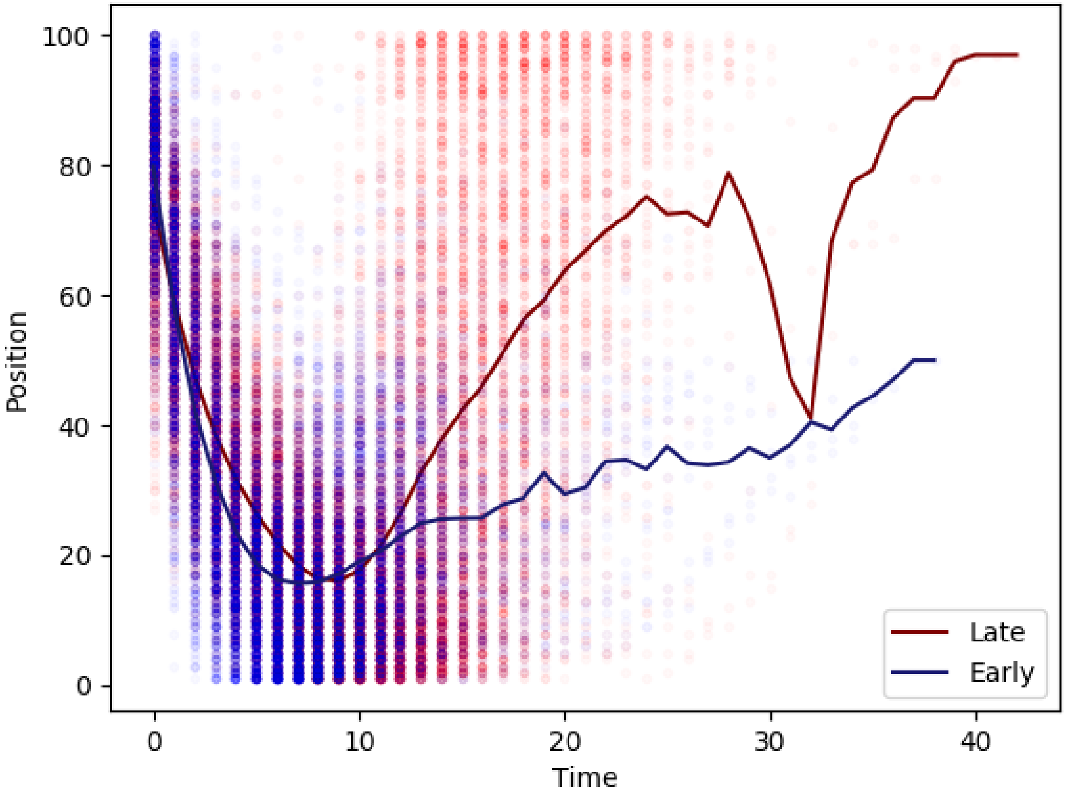


Figure 5.3: Alignment target variable chart trajectories popular over different time periods. Trajectories popular earlier correspond to classes 40, 39, 38, 37, and 10. Trajectories popular later correspond to classes 35, 34, 33, and 32.

5.2 Longevity and peak position

We have demonstrated that one can use a regression model to distinguish songs with strong performance from songs with weak performance along the weeks on chart and peak position dimensions. Now, we want to show that these two dimensions do not necessarily carry the same information by modelling them separately. In doing so, we hope to argue that these dimensions should be incorporated together into whatever target variable definition one uses to model chart performance. We consider two sets of tasks, each which models a separate dimension. The time tasks separate songs based on their longevity, and the peak tasks separate songs based on their peak position; there are three tasks modelled along each dimension.

We derived our task categories based on the interval target variable because its classes have longevity and peak position explicitly defined. We also used the 20 topic model feature set because it was the best performing feature set for the interval target variable in the binary task. These tasks are designed to examine the differences between smaller sets of songs, which means that a logistic regression model may not work well. This is because if there are too many features relative to the number of observations in a model, then the likelihood of a perfect separation between some feature and some class goes up. To avoid this, we opted to use a regularized logistic regression, the lasso logistic regression, to model our songs for these tasks. The lasso regression makes use of a penalty term, λ, to determine the extent to which features are excluded from the model; λ also determines the model's robustness to overfitting. When we model our data with a lasso regression, we first perform 10-fold cross-validation to estimate the λ parameter, then we select the λ one standard error from the minimum error denoted λ_{1se} and use it in a model fit to all of the task data.

Instead of using AIC, we use another metric to evaluate model fit, the percentage of deviations explained by the features, which corresponds to the log-likelihood of the model being fit to unseen data, and we also consider an additional feature set for each task where the metadata features have been removed as we previously observed how they had a lot of statistical power, which may have been overshadowing the other features. In this way, we can directly measure the impact of the audio and lyric features.

Now, we outline each of the modelling tasks in each set. For the time tasks, we fixed the data in each task over a peak position range and separated songs into three groups: hits, flops, and the middle ground. The songs in the middle ground group were not modelled; they acted as a cushion between the other classes to ensure that those classes were more distinct. We chose to fix peak position over different ranges corresponding to high, medium, and low peak positions to get a complete view of how songs with short and long lifespans differ. Table 5.6 shows the class information for each task, which we denote with the labels A, B, and C. It can be observed that some classes are not balanced and that this may lead to their models performing better along certain metrics simply because the model could favour the more frequent class. On the extreme end, task A has a ratio worse than 1:2 between its hits and flops. While this is a problem, we chose to overlook it in this subsection to ensure that classes were more distinct and because downsampling would drastically shrink some of the tasks' sample sizes.

For the peak tasks, we followed the same approach as the time tasks but fixed the data over weeks on chart intervals instead of peak position. We were unable to create tasks

Table 5.6: Time tasks with hit and flop classes.

Task	Peak position	Hit weeks on chart	Flop weeks on chart	# Hits	# Flops
A	61-100	9-20	1-4	384	783
B	21-40	17-21	1-12	469	753
C	#1	21+	9-16	212	130

with the same level of coverage from long to short lasting songs because the songs that lasted only a few weeks were much more likely to have high peak positions; thus, our time interval ranges were selected for songs that lasted medium and long amounts of time on the charts. Table 5.7 shows the class information with labels A, B, and C, and unlike the time tasks, these tasks are smaller and more balanced, which could mean less noisy.

Table 5.7: Peak tasks with hit and flop classes.

Task	Weeks on chart	Hit peak position	Flop peak position	# Hits	# Flops
A	21+	#1	11-40	212	242
B	13-16	#1	41-100	110	248
C	9-12	1-10	61-100	203	235

From section 5.1, we saw how two data subsets were modelled based on separate target variable definitions. The alignment task was more separable than the interval task even though the interval task had more constraints on its flop class. Still, both tasks' hit and flop classes were able to be separated for the most part by their logistic regression models. For a smaller task in one dimension, one might not expect the same outcome. It is reasonable to assume that as two classes become less separable, their models will have more difficulty distinguishing them from each other. By plotting the class trajectories for each tasks, we can examine the degree of separability in each tasks and speculate on how their lasso regression models will perform. The classes in each time task are plotted in Figure 5.4, and the classes in each peak task are plotted in Figure 5.5.

Visually, it looks like the time tasks from Figure 5.4 have a higher degree of overlap when compared to the peak tasks from Figure 5.5, but remember that the time tasks are not evaluated based on separability along the peak position axis. All three tasks from Table 5.6 have one interval weeks on chart ranges as their cushion, so it is hard to tell which one will be most separable. For task A, it is songs lasting 5-8 weeks; for task B, it is song's lasting 13-16 weeks; and for task C, it is songs lasting 17-20 weeks. The question becomes: is the difference between songs that last less time on the charts more significant than the difference between songs that last more time? If it is less time, then task A will

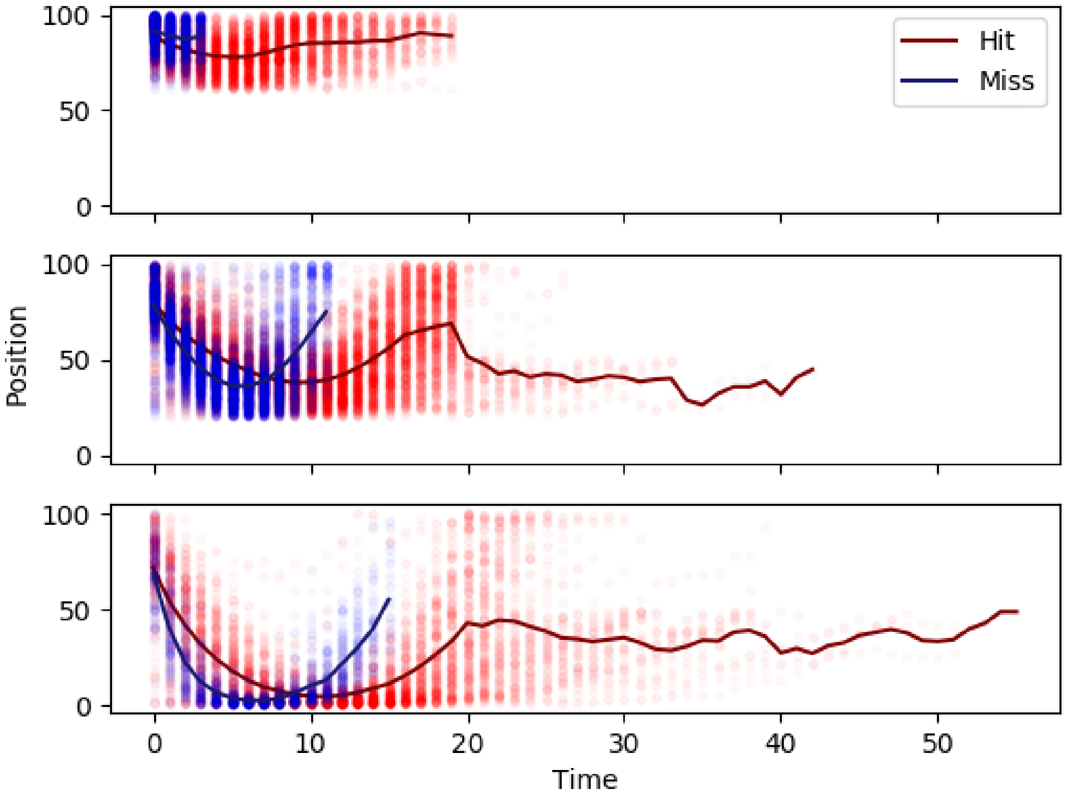


Figure 5.4: Chart trajectories for time tasks. (Top) Task A, comparing songs lasting 9-20 weeks and 1-4 weeks. (Middle) Task B, comparing songs lasting 17-21 weeks and 1-12 weeks. (Bottom) Task C, comparing songs lasting 21+ weeks and 9-16 weeks.

be much more separable than tasks B and C, and vice versa. It could be argued that the variability in musical attributes for songs from task A is going to be larger than the variability for songs from tasks B and C because there is a lower bar to reach, so more songs from different genres can reach it, and while variability is necessary for modelling the differences between classes, the variability in task A is more likely to stem from the diversity associated with having a lower bar instead of lasting a short or long amount of time on the charts. On the other hand, if we visually look at the trajectories, we can see that tasks A and C have a lot less variability in their peak position than tasks B.

In Figure 5.5, task A has a fair amount of overlap between its peak position especially

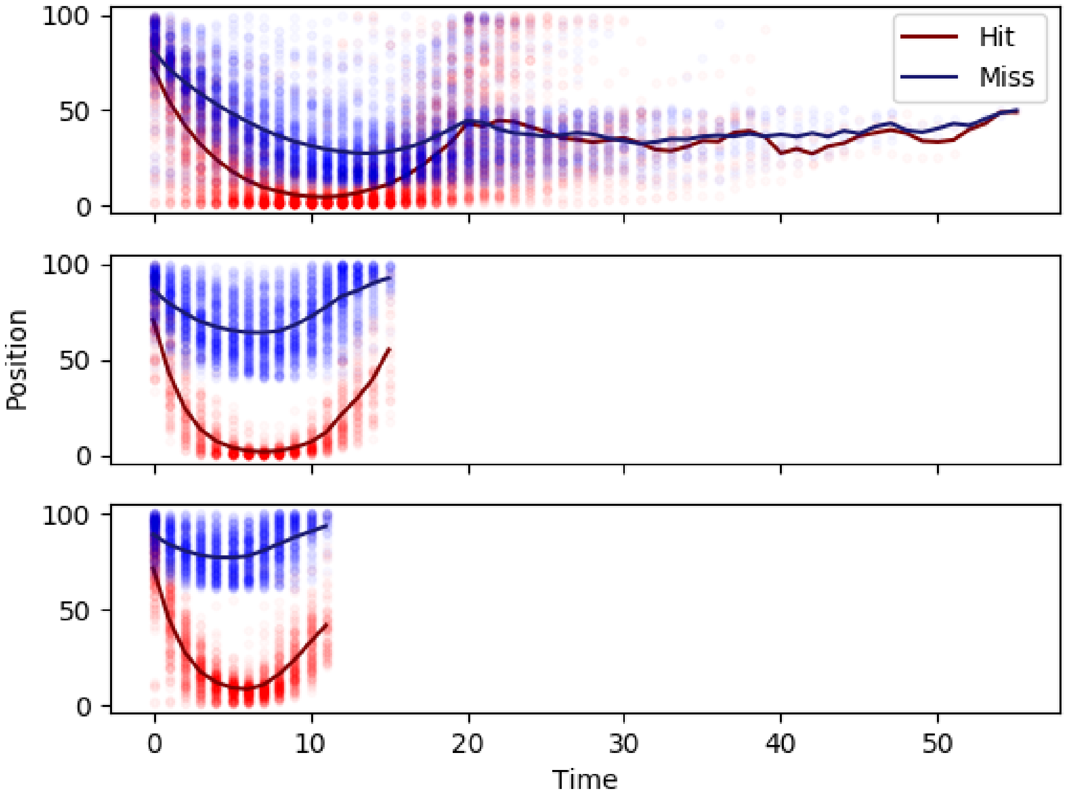


Figure 5.5: Chart trajectories for peak tasks. (Top) Task A, comparing songs peaking at #1 and 11-40. (Middle) Task B, comparing songs peaking at #1 and 41-100. (Bottom) Task C, comparing songs peaking at 1-10 and 61-100.

after week 20 even though this set of tasks is for modelling peak position differences. Unlike task A, tasks B and C have distinct trajectories with a cushion of 2-40 and 11-60 peak positions respectively. It is quite clear that task A's model will perform the worst of these tasks because its classes are not very separable, but the question of whether task B or C is more separable is interesting . Task C has a larger cushion between its classes, but as was mentioned in chapter 4, chart position corresponds to an exponential distribution for song sales, where songs with lower peak positions sell exponentially more units; thus, it could be argued that task B will be more separable because it involves a larger cushion and has no overlap between its classes.

5.2.1 Longevity and peak position model fit

In this subsection, we fit the data from each task to lasso logistic regression models and then evaluated the models on their ability to separate hits from flops. We looked to identify and compare important features between models within each set and across sets.

The time task summary tables are shown in Table 5.8, and we can observe that task C has the best separation of the three tasks; however, when its metadata features are removed, the associated model has a similar performance to task B's model with no metadata features. Task A's model has the lowest percent deviations explained score, which indicates that the task's classes were not well separated by the model. The contrast between task A and tasks B and C may suggest that there are dynamics beyond just the longevity of a song at play when modelling song performance. As was suggested earlier, task A's failure may stem from the fact that the songs involved in its classes had lower weeks on chart positions, which means that a wider variety of songs could meet that threshold of success, including failed commercial songs or niche successes, which makes them harder to distinguish across some performance metric.

Table 5.8: Lasso logistic regression models for various time tasks.

	Dependent variable:					
	Interval target variable					
	(A)	(A control)	(B)	(B control)	(C)	(C control)
10+ hits	0	.	−0.557	.	0	.
reissue	0.952	.	1.6	.	1.227	.
genres						
rap	0	.	0	.	0	.
country	0.489	.	0.725	.	0	.
pop	−0.338	.	0	.	0	.
time blocks						
1988-1992	0.156	.	0.837	.	0.636	.
1993-1997	1.363	.	3.38	.	0.831	.
The Echo Nest similarity						
genre similarity	0	0	0	0	0	0
The Echo Nest features						
acousticness	−0.765	−1.375	−0.624	−1.469	−1.044	−1.669
danceability	0.123	1.319	0	2.693	0.009	3.522
valence	−0.331	−1.163	−0.327	−2.793	−0.236	−2.254
topic mixture similarity						
genre similarity	0	0.425	0.705	3.481	0	0
topic mixtures						
X3	0	0	−0.524	0	0	0
X9	0	0	0	0	0.415	1.019
X11	0	0	0	−0.136	0.115	0
Observations	1,167	1,167	1,222	1,222	342	342
DF	18	7	23	13	25	13
λ_{1se}	0.028	0.027	0.014	0.021	0.025	0.036
%Dev	0.204	0.087	0.541	0.24	0.63	0.253
Note:						*p<0.1; **p<0.05; ***p<0.01

Across each time task model, The Echo Nest features danceability, acousticness, and valence had consistent effects. Acousticness and valence had negative effects, while danceability had positive effects. The acousticness effects here contrast with acousticness effects from earlier interval target variable models, which were positive. This implies that there are differences between modelling song performance in one and two dimensions.

Of the metadata features, the time block effects were the most consistent, whereas, others had sparser effects. Like acousticness, the genre topic mixture similarity had a contrasting effect between the time task models and the original interval target variable models. In models A control, b, and B control, it had a positive effect, and in the 20 and 40 topic mixture feature set models, it had a negative effect. This supplies us with more evidence that there are differences between these two modelling tasks.

Topic mixture effects were sparser than in the earlier models, but we noticed that when removing the metadata features, more topic mixture effects contributed to the models. The effects found here again differ with those in the earlier interval target variable binary tasks. $T_{20,9}$, referring to derogatory content, had a positive effect in task C and its control. It likely refers to some elements of rap music because it includes a racial term for African Americans in its most likely set of words. This is interesting because Dhanaraj & Logan [7] similarly observed a topic associated with rap and derogatory content emerge from their topic model features when they modelled song performance. We also observed other topics contribute to the models like $T_{20,3}$, which refers to dancing, in task B with a negative effect and $T_{20,11}$, which refers to musical passion, in task B control and task C with mixed effects.

In Table 5.9, we have the summary tables for the peak tasks. We found that task C is the most separable, but with the metadata features removed, task B became most separable. Task B's model also had many more active topic mixture effects, which could have made it more robust to removing the metadata features; this can be seen in the full table in Table C.4. Task C and its control's model only had one non-zero topic mixture effect each, whereas, task B had three, and its control had six. One explanation for this could be that because task B has a much more refined hit class, songs peaking at #1, lyric features were more useful for distinguishing its hits from flops; on the other hand, task C's hit class contained songs peaking between positions 1-10.

As we predicted, task A is the least separable; its control model had only one non-zero feature, danceability. This is likely due to the high degree of overlap in position between classes, and while peak position is not the same as position, it is derived from position. A more musicological explanation for this might be that songs from task A all lasted 21+ weeks, and so they needed a strong level of support from fans and labels, which means that their hits and flops are more likely to be closely related.

Compared with The Echo Nest feature effects from the time tasks, those features' effects here are reversed. Valence had negative effects, acousticness had positive effects, and danceability had mixed effects (positive in task A and its control and negative in task B and task C's controls). Relative to the interval target variable binary tasks, this means acousticness is consistent with those models' features, whereas, valence is not. Again, this offers evidence that there is a difference between modelling songs based on peak position or the number of weeks on chart that they last; thus, one could argue that the underlying elements of those target variables, time and magnitude, should be modelled together.

Table 5.9: Lasso logistic regression models for various peak tasks.

	Dependent variable:					
	Interval target variable					
	(A)	(A control)	(B)	(B control)	(C)	(C control)
10+ hits	0	.	0	.	0.259	.
reissue	0	.	0	.	0	.
genres						
rap	0	.	0	.	0	.
country	0	.	−0.169	.	−0.674	.
pop	0	.	0.256	.	0.307	.
time blocks						
1988-1992	0	.	0	.	−2.94	.
1993-1997	−0.107	.	−0.606	.	−2.781	.
The Echo Nest similarity						
genre similarity	0	0	0	0	0	0
The Echo Nest features						
acousticness	0.292	0	2.584	3.015	0.386	2.187
danceability	1.42	1.016	0	−0.672	0	−2.034
valence	0	0	0.727	2.658	1.078	2.9
topic mixture similarity						
genre similarity	0	0	0	−0.565	0	−1.386
topic mixtures						
X9	0	0	−0.448	−2.598	0	0
X11	0	0	0	0	0.158	0
X13	0	0	0.952	0.79	0	0
Observations	454	454	358	358	438	438
DF	14	1	20	11	23	9
λ_{1se}	0.043	0.063	0.024	0.031	0.019	0.038
%Dev	0.088	0.018	0.615	0.352	0.67	0.266
Note:						*p<0.1; **p<0.05; ***p<0.01

Other feature effects like the time blocks and the genre features had consistent effects in the models for tasks B and C. The genre topic mixture similarity had a negative effect in task B and C's control models, and the topic mixtures, $T_{20,11}$, and $T_{20,13}$ had negative, positive, and negative effects for select tasks. Notably, $T_{20,13}$ is the baby talk topic that was present across all topic model feature sets for the alignment target variable.

We can see how The Echo Nest, metadata, and some topic mixture features are important for distinguishing hits from flops across these two sets of tasks. Within a task set, there is generally some agreement on the order and importance of features, but between sets, there is a noticeable reversal of direction for many of the features like acousticness, valence, and $T_{20,9}$. This offers evidence that there is a need to model some target variable that incorporates peak position and weeks on chart together as separately they do communicate a consistent message about what features are important for distinguishing hits from flops.

5.2.2 Time and peak tasks model performance

We previously observed how AIC was a good indicator of a model's performance, but for the lasso logistic regression models, we have used a different metric, deviations explained, as a proxy for model fit. In this subsection, we evaluate our models and see how good of a proxy deviations explained really is for measuring model performance.

Again, we perform 10-fold cross-validation, and for each fold, we train a model on the other folds, but because we are using a lasso logistic regression where we need to estimate λ, we first re-partition the nine training folds into 10 new folds and perform another round of 10-fold cross-validation to find λ_{1se} and then use it in a model to evaluate the test fold. In total, we use 100 training sets with folds, but each training set contains at least nine folds. The resulting accuracy, precision, recall, and F1 scores were averaged over all of the higher level folds and recorded.

It is also important to note that some of these datasets are unbalanced with 1:2 ratios between hits and flops, and as a result, this can yield divergent precision and recall scores. We expect that models with lower deviation scores and fewer degrees of freedom will have larger differences between their precision and recall scores because they will just predict the majority class. We describe these models as dummy models because they do not make use of many features.

The 10-fold cross-validation results for the longevity tasks are shown in Table 5.10, and the F1 score is consistent with the deviation scores. Task C had the highest percent deviation score and has the highest F1 score; task B follows closely behind, and task A's model performed worst. Task A's control model even had a higher precision than task A's model, which may seem odd as the control only has 7 degrees of freedom, but recall that dummy models are more likely to favour the majority class, hence the higher precision score.

Table 5.10: 10-fold cross-validation results for lasso logistic regression models fit to time tasks.

	A	**A control**	**B**	**B control**	**C**	**C control**
accuracy	0.747	0.572	0.880	0.730	0.929	0.757
precision	0.740	0.878	0.872	0.851	0.976	0.929
recall	0.613	0.431	0.839	0.616	0.919	0.753
F1	0.661	0.575	0.849	0.710	0.946	0.828

Next, the 10-fold cross-validation results for the peak tasks are shown in Table 5.11. Again the model with the highest deviation score, task C, performs the best, and the model with the lowest deviation score, task A, performs worst. In two folds for task A's control, no true positives or false negatives were found resulting in recall not being computed; thus, we took the average recall and F1 scores using the remaining eight folds.

Table 5.11: 10-fold cross-validation results for lasso logistic regression models fit to peak tasks.

	A	**A control**	**B**	**B control**	**C**	**C control**
accuracy	0.621	0.533	0.933	0.821	0.938	0.774
precision	0.918	0.706	0.918	0.845	0.966	0.881
recall	0.570	0.504	0.877	0.692	0.911	0.735
F1	0.697	0.638	0.894	0.750	0.936	0.790

For certain tasks we were able to separate the hits from flops relatively well; however, only a small proportion of this separation can be explained by The Echo Nest and topic mixture features. When the metadata features were removed, model performance dropped significantly across the F1 scores. Along with this observation, we also saw that the deviations explained metric like AIC was a good proxy for model fit in lasso logistic regression models.

5.3 Multiclass classification of chart trajectories

From our binary task results, it may seem difficult to imagine that more than two classes can be modelled in a meaningful way given the over-reliance on metadata features and the balancing issues between classes, but we believe that the changes in feature values across tasks indicates that multiple classes of songs can be modelled.

Given that the alignment models from the first question had lower AIC ratios, we decided to use them for this task. While The Echo Nest feature set previously performed best for the alignment target variable binary task, it does not have any topic mixture features, which limits the potential for observing specific links between features and trajectory classes; thus, we chose the 10 topic model feature set instead as its model performance was only marginally worse, and it allows us to find links between trajectories and lyric features. We also decided to use the lasso logistic regression model again because we are using more than two classes, and that increases the likelihood of a perfect separation existing between some class and some feature.

Previously, we also saw how unbalanced classes could lead to dummy models with high precision and low recall scores; thus, we decided to balance the classes for this task through downsampling, so each class has the same proportion of songs as the smallest class. As mentioned earlier, the downside to this is that our task dataset becomes less representative of the world, but it is a tradeoff that we make here so that class predictions are less skewed. Also, as classes are balanced, we do not need to threshold the class prediction probabilities as the model's only consideration for computing class probabilities will be the feature values.

In Figure 4.11, we showed five distinct alignment target variable classes in blue with related interval target variable classes superimposed on them in red to illustrate the usefulness of building target variables from clustered song trajectories. We use these classes here because they are all large and fairly distinct from each other.

We followed the same procedures as previously outlined in the second set of tasks. Instead of a binary model, we used a one-against-all model, which is essentially a set of binary models. We also, compared the full feature set against a reduced feature set with no metadata features to again learn about the contributions of the non-metadata features.

Each class contained 256 songs after downsampling, and we modelled 5 classes, so in total, we analyzed 1,280 songs. Each class is given an alphabet label, but we can also describe them based on how successful their trajectories were. Class A songs do not peak low or last long, so they are likely flops. Songs from classes B and C peak in the medium range and last a reasonable number of weeks, so they are likely moderate successes. Songs from classes D and E peak close to the bottom and last a long time, so they are likely hits. Classes B and C differ in their average peak position and weeks on chart. Class B songs have longer chart trajectory arches and peak lower. The same difference is apparent in classes D and E with class D having the longer arches and lower peaks.

5.3.1 Multiclass model fit

When we fit our models, we find that fewer features are important. Table 5.12 shows the model fit using the full feature set, and Table 5.13 shows the reduced feature set. The deviation percentage explained by the full model is 20.9%, while it is only 3.5% for the reduced feature set, which is comparable to model A from the peak tasks, a dummy model. This indicates that for our data, separating more than two classes comes down largely to what their metadata features are; if they are separable, then the classes will be separable.

We do still see some activity in The Echo Nest features, but it is much sparser here. Acousticness and danceability can distinguish some classes in both models. Songs that are less danceable are associated with class B, and in the control, songs that are more danceable are associated with class E. Songs that are less acoustic are associated with class C, songs that are more acoustic are associated with class D and to a lesser extent class B but only in the control. Surprisingly, valence has no power to distinguish any of these classes even though it was useful in many of the binary models.

Table 5.12: Multinomial lasso logistic regression model for modelling chart trajectories using full feature set.

	Dependent variable:				
	Alignment target variable				
	(A)	(B)	(C)	(D)	(E)
10+ hits	0	0	0	0	0
reissue	0	0	0	0	0
genres					
rap	0	0	0	0	0
country	0.159	0	0.23	−0.004	0
pop	0	0	−0.022	0.3	0
time blocks					
1988-1992	0	−0.477	0.081	−0.445	1.115
1993-1997	0	0	0.219	0	0
The Echo Nest similarity					
genre similarity	0	0	0	0	0
The Echo Nest features					
acousticness	0	0	−0.93	0.361	0
danceability	0	−0.166	0	0	0
valence	0	0	0	0	0
topic mixture similarity					
genre similarity	0	0	0	0	0
topic mixtures					
X1	−0.216	0	0	0	0
X8	0	0.003	0	0	0
Observations	1,280				
DF	22				
λ_{1se}	0.025				
%Dev	0.209				
Note:			*p<0.1; **p<0.05; ***p<0.01		

The topic mixtures also contribute a little in these models. $T_{10,1}$ has a negative effect on class A in the full model and refers to the baby talk. On the other hand, $T_{10,4}$ has a positive effect on class A songs in the control model and refers to derogatory content. These are both topics that we have observed before, but we also find a new topic $T_{10,9}$, which is associated with class B. It refers to being active and has a positive effect.

We have observed that some The Echo Nest features and topic mixtures can separate the songs to a small degree, and we have attempted to pair feature effects with the likelihood of a song following some trajectory; however, these effects pale in comparison to the metadata features, specially the genre and time block control variables, which explain a majority of the variation in our data that these models can find.

Table 5.13: Multinomial lasso logistic regression model for modelling chart trajectories using feature set with metadata features removed.

	Dependent variable:				
	Alignment target variable				
	(A)	(B)	(C)	(D)	(E)
The Echo Nest similarity					
genre similarity	0	0	0	0	0
The Echo Nest features					
acousticness	0	0.419	−0.843	0.885	0
danceability	0	−0.243	0	0	0.821
valence	0	0	0	0	0
topic mixture similarity					
genre similarity	0	0	0	0	0
topic mixtures					
X4	0.023	0	0	0	0
X8	0	0.225	0	0	0
Observations	1,280				
DF	7				
λ_{1se}	0.03				
%Dev	0.035				
Note:			*p<0.1; **p<0.05; ***p<0.01		

5.3.2 Multiclass model performance

We performed 10-fold cross-validation, training a lasso model with the same procedures outlined for the time and peak tasks. As we have multiple classes, instead of using F1, recall, accuracy, and precision to measure model performance, we use confusion matrices. We can calculate the accuracy of the models by taking the sum of counts along the diagonal and dividing them by the overall sum.

What we observe is that certain classes are more likely to be predicted. In Figure 5.6 on the left, we have the confusion matrix for the full feature set, and on the right, we have the confusion matrix for the reduced feature set. In the full feature set, classes C, D, and E are somewhat more distinguishable than classes A and B, and in the reduced feature set, this disparity is amplified; the model's predictive ability for classes A and B is essentially random. As a result, the accuracy falls from 42.9% for the full feature set to 29.8% for the reduced feature set.

Class C has a majority of the predictions in both models, and this can be explained by taking a look at its features. Recall from the summary tables how less acoustic songs are more likely to belong to class C. If we examine the 1,280 songs, on average the songs from class C have a lower average acousticness with a smaller variance; thus, it is reasonable to assume that songs with low acousticness will be predicted as belonging to class C especially in the reduced feature set model.

Something else to note is that classes A and B were the least distinguishable; however, they were the only classes where topic mixtures were important according to both models; thus, this indicates that the topic mixture features were not very useful in the end.

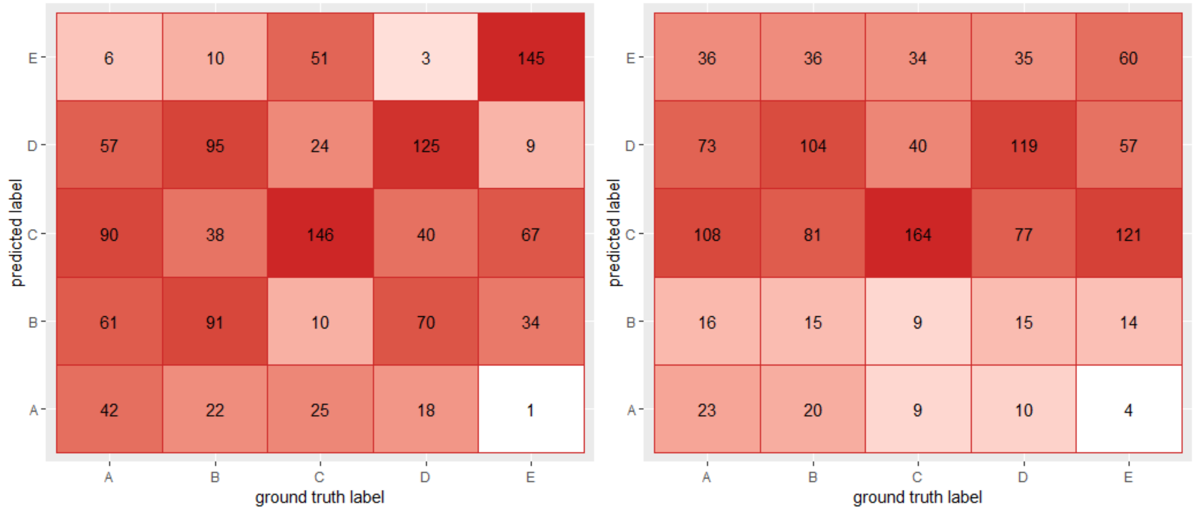


Figure 5.6: Confusion matrix showing multiclass logistic regression class predictions for chart trajectories. (Left) Full feature set. (Right) Metadata features removed.

We are only somewhat successful at classifying songs into multiple chart trajectories, and this partial success can largely be attributed to our metadata features. We can apply various models to our data with different levels of complexity; however, until we find more discriminatory features or data that is not so dependent on control variables, the links found between classes and features will not be strong or meaningful. As we saw, the topic mixtures identified as important could not separate their classes well. Even though these features cannot be used effectively for modelling, they can still be used to explore how music has evolved over the past half century within and between genres and this is what we would like to touch on in the next chapter.

5.4 Summary

In this chapter, we fit logistic and lasso logistic regression models to various feature sets to answer specific questions exploring the relationships between musical attributes and chart trajectories. We were not overly successful at these tasks, but we showed the importance of modelling peak position and weeks on chart together. There is still much work to be done in finding meaningful features for modelling chart trajectories.

Chapter 6

Critique

Our modelling results were not decisive as we did not reach any grand conclusions about the importance of a particular feature of interest. In contrast, Askin & Mauskapf observed how important their typicality measure was for modelling song success [1]. In this chapter, we review their paper in more detail and apply their methodology to our complete dataset to compare findings and see if we can recreate their results.

6.1 Experimental summary

First, we outline some of the experimental differences between the two studies. Askin & Mauskapf hypothesized that a song's chart success is related to how similar it is to its neighbors, the other charting songs, and that the most successful songs are more likely to be optimally differentiated from their neighbors, not too similar and not too distinct. They used a typicality measure to represent a song's similarity to its neighbors and argued that if their hypothesis was true, then they would observe an inverted U-shaped distribution for the likelihood of a song reaching a top position over typicality where the likelihood would be at a maximum when typicality is at some mid-range value. We did not conduct our experiment with any hypotheses around the importance of a specific feature. Instead, we simply wanted to justify our two-dimensional target variables and identify possible links between song features and chart patterns.

Askin & Mauskapf gathered data from a variety of sources, many of which we also used in our study. These sources included the weekly Billboard Hot 100 chart to gather chart position data, The Echo Nest API to gather audio and metadata features, and Discogs, an online music database, to gather genre data. While we used the same chart and audio data sources, our choices for genre classification were different. The Discogs genres include pop, blues, brass and military, children's, classical, electronic, folk, world, and country, funk and soul, hip hop, jazz, Latin, non-music, reggae, rock, and stage and screen. Instead of using Discogs' genre definitions, we aggregated Spotify's genre tags into higher level categories. This choice to derive our own genres was in part because we did not agree with Discogs' categorization of country, folk, and world music as belonging to one group.

Both studies also shared controls for long songs, past hits, crossovers, reissues, and date of release (five year time blocks), but Askin & Mauskapf also developed controls for songs being major label releases and artists having multiple memberships in separate musical acts. We did not use these controls in our study or in this re-implementation because we were less certain on how to distinguish major labels from subsidiaries and band members from other musical affiliates like audio engineers.

The feature of interest in their study, typicality, is a genre-weighted cosine similarity measure based on The Echo Nest features where genre weights were defined as the average similarity between genres over the past year. In our study, we followed their approach to develop a set of context-based measures; however, we computed the cosine similarities for both lyric topic mixtures and The Echo Nest features, resulting in 15 similarity measures instead of just one. We compared songs if they charted within one year of each other, had the same genre and similar chart dates, or were recorded by the same artist, but instead of using a genre weight, we used a temporal weight to penalize songs released further from the song of interest's debut.

In their work, they used eight models to study the relationships between song features and chart success, which included modelling peak position and weeks on chart. Models #3 and #4 from their study as shown in Table A.1 used an ordered logit to model an inverted peak position (order of positions #1 to #100 is reversed) with audio features, control variables, and the typicality measure in model #3 and the typicality measure and its square term in model #4. In both models, they found that typicality was an important feature with a large absolute magnitude and a high degree of statistical significance. In model #3, they observed a negative effect associated with typicality indicating that songs more closely related to their neighbors were less likely to reach the top of the charts, and in model #4, they observed a significant squared term. As polynomial terms are less easy to interpret, they opted to visualize the results from model #4 by plotting the predicted marginal probabilities of each song having a peak position within some range of peak positions over its typicality value. What they observed is an inverted U-shaped distribution for top 40 peak position ranges and a U-shaped distribution for 41-100 peak position ranges as is shown in Figure A.2; thus, they argued that this was evidence that song success is tied to typicality.

In contrast, when we fit our models with context-based similarity measures, only one similarity feature had a consistent effect and that was the genre topic mixture similarity. Across most of our models with the exception of those constructed for the temporal tasks, it shared the same effect as Askin and Mauskapf's typicality feature where less similar songs were more likely to be successful, but they differed in the magnitude of their effects. The genre topic mixture similarity's magnitude was dwarfed by other features like the time blocks, whereas, their typicality measure had a large effect relative to other features. This may be because they represented the modelling task as a regression problem, whereas, we chose to represent it as a classification problem. Interestingly, when they modelled weeks on chart in their models #5 and #6 as shown in Table A.1, they were able to observe the same effect for the typicality feature as was observed in models #3 and #4, but it was associated with more error, and the time blocks had a larger effect, which is in line with what we observed in our own models.

6.2 Re-implementation

We attempt to re-implement Askin & Mauskapf's experiment with song data and features from our complete dataset. They used 25,102 songs from the charts between 1958 and 2016, whereas, we used 7,726 songs between 1958 and 2012, so our coverage is much lower. We gathered genre data by querying Discogs' *search* method with song titles and artists names and were able to find 7,683 matches. They used an ordered logit method from the statistical software package Stata, but we did not have access to this, so we used a similar ordered logit method, *polr*, from the R package MASS [36]. Another difference between implementations is that some of our controls may follow different definitions because we sometimes had to rely on high level descriptions from their paper.

We were able to construct a close approximation to their genre-weighted typicality measure, and Figure 6.1 shows its distribution in the red, though this is difficult to see as it overlaps with the distribution for typicality values computed using Spotify genre definitions in blue, which is explained later. In their paper, typicality followed a bi-modal distribution and spanned a range from 0.26 to 0.92 as is shown in Figure A.1, and with our recreation, the distribution is similarly bi-modal; thus, we claim that the distributions are closely related.

Following Askin & Mauskapf's approach, we were able to reproduce some of their statistical results. Models #1 and #2 from Table 6.1 refer to models #3 and #4 from their paper as shown in Table A.1. Comparing our typicality measures with theirs, we can observe that they have similar magnitudes and direction, but our features are less

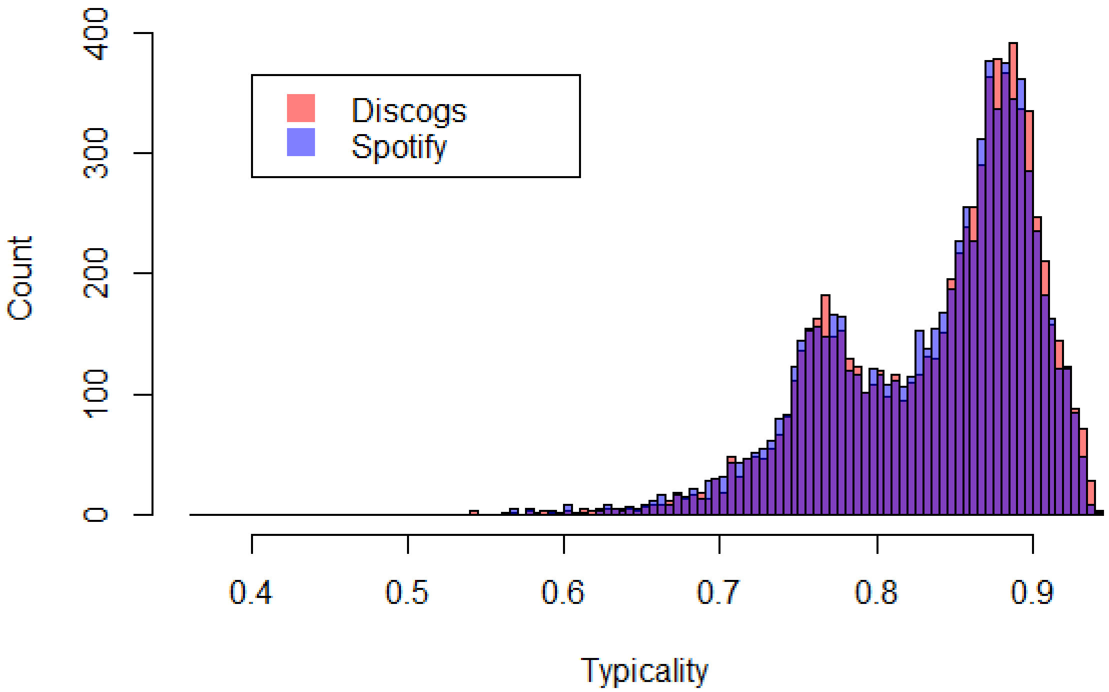


Figure 6.1: Typicality distributions for songs from the complete dataset using different genre definitions based on approach from [1].

significant. Other features in our models also have larger effects than their features, but relative to typicality, they are still quite small, which is a surprising contrast with the weak but significant effect of the genre topic mixture similarity measure in our main experimental models. The models in Table 6.1 show that the typicality feature plays a large role in influencing how a song will perform because of its significance and magnitude. Model #1 is consistent with model #3 from their paper and shows that songs that are more unique will have a higher likelihood of reaching the top of the charts. As their interpretation of model #4 is more broad, we do not interpret model #2's features directly except to note that the magnitude of the typicality features is consistent with their model, but it is less significant for us.

Table 6.1: Ordered logit regression models fit to complete dataset using Discogs genre definition in models #1 and #2 and Spotify genre definition in models #3 and #4.

	Dependent variable:			
	Peak position (inverted)			
	(1)	(2)	(3)	(4)
typicality	-2.020^{***} (0.681)	8.079 (5.486)	-1.175^{*} (0.673)	9.999 (6.205)
typicality^2		-6.413^{*} (3.482)		-7.071^{*} (3.917)
long song	-0.308^{***} (0.108)	-0.306^{***} (0.108)	-0.302^{***} (0.107)	-0.294^{***} (0.107)
2-3 hits	-0.319^{***} (0.058)	-0.321^{***} (0.058)	-0.364^{***} (0.059)	-0.364^{***} (0.059)
6-10 hits	-0.112^{**} (0.054)	-0.112^{**} (0.054)	-0.178^{***} (0.055)	-0.178^{***} (0.055)
10+ hits	-0.091 (0.063)	-0.092 (0.063)	-0.153^{**} (0.064)	-0.150^{**} (0.064)
crossover	0.215^{**} (0.104)	0.215^{**} (0.104)	0.062 (0.052)	0.059 (0.052)
reissue	0.553^{***} (0.212)	0.556^{***} (0.212)	0.490^{**} (0.212)	0.476^{**} (0.212)
time blocks				
1988-1992	-0.371^{***} (0.122)	-0.403^{***} (0.124)	-0.360^{***} (0.121)	-0.394^{***} (0.123)
1993-1997	-1.086^{***} (0.133)	-1.134^{***} (0.136)	-1.050^{***} (0.133)	-1.100^{***} (0.136)
Discogs genres				
hip hop	0.391^{***} (0.113)	0.393^{***} (0.113)		
Spotify genres				
rap			0.680^{***} (0.210)	0.671^{***} (0.210)
Observations	7,683	7,683	7,683	7,683
Note:		*p<0.1; **p<0.05; ***p<0.01		

Instead of analyzing the summary table for model #4, Askin & Mauskapf used a plot of marginal probabilities to support their hypothesis around optimal differentiation, and we attempt to recreate their plot using model #2 where we plot the marginal probabilities of songs having peaks within specific ranges. Figure 6.2 shows the marginal probabilities of each song belonging to some class with an order two polynomial fit over the song probabilities. Unfortunately, we were not able to recreate the order two relationships found in Figure A.2. Instead, we find the opposite trends present, albeit with a very weak curvature associated with all of the model fits. Songs with higher peak positions in the top 20 follow a U-shaped distribution, songs with peak positions from 21-40 follow a positive trend, and songs with peak positions from 41-100 follow an inverted U-shaped distribution. This is not what we would expect given the model coefficients from Table 6.1 as we know from model #1 that hit songs are negatively associated with typicality; thus, assuming that the models convey similar messages around the typicality feature, one would expect that songs in lower charting positions would have a higher likelihood of success when their typicality scores were lower, but we observe the opposite effect. The songs with typicality scores around 0.7 should have the highest probability of belonging to the hit classes. This suggests that we might not have enough data to draw the same inference as Askin & Mauskapf around songs with lower typicality scores.

Another difference between the plots is that the stationary point of each polynomial model fit is between 0.7 and 0.8 in our models while it is between 0.5 and 0.6 in Askin & Mauskapf's models. This is important to note because their mean typicality score was 0.81, whereas, it is 0.836 for us.

6.3 Areas of concern

The reasons why the results from our models in chapter 5 are so different from Askin & Mauskapf's results is likely because of distinct experimental design choices (regression versus classification) and the different amounts of data used for modelling. In this chapter, we attempted to re-implement their experiments using the data we had access to and were able to reproduce some of their results like the typicality distribution and the ordered logit summary tables while failing to do so for others like the marginal probability polynomial model fits. Below, we re-examine some of their claims and highlight some of our concerns with reference to the re-implementation. The areas of concern include the following topics:

1. Typicality range
2. Result robustness
3. Predicted marginal probability curves

6.3.1 Typicality range

Originally, Askin & Mauskapf hypothesized that songs that are optimally differentiated from their competition are more likely to succeed on the charts. They referenced one #1 song from their dataset as an example, the Beatles' *Come Together*, with a typicality score of 0.66, over two standard deviations below the average score for songs at that time. In our dataset, it has a typicality score just above the bottom 1% of typicality scores for songs. Clearly, this is not an example of an optimally differentiated song but instead a popular song that is very dissimilar from its competition.

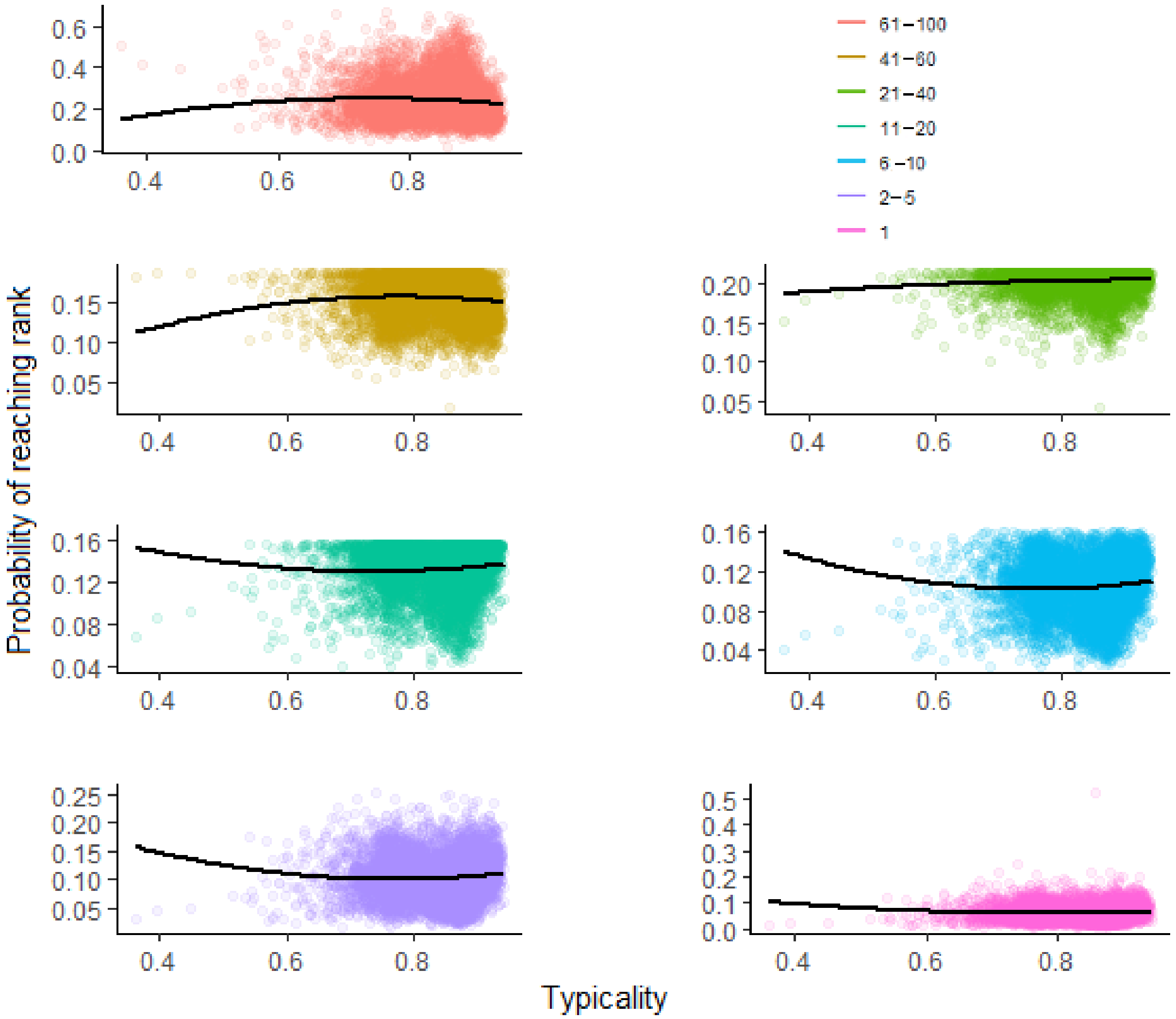


Figure 6.2: Likelihood of songs from the complete dataset reaching specific peak position intervals given typicality scores from model #2 in Table 6.1.

We recreated their typicality measure in Figure 6.1 with the same distribution in red, though it can be observed that the vast majority of our songs have scores that lie between 0.7 and 0.9. Overall, their mean typicality score was 0.81 with a standard deviation of 0.06, and we observe a mean score of 0.838 with a standard deviation of 0.065 for the typicality scores of songs in our dataset. This calls into question their claims around their hypothesis being validated as the hypothesis casts a much wider net over typicality values than the range between 0.7 and 0.9. The idea of an inverted U-shaped distribution has only weak support because the polynomial curves from Figure A.2 reach their stationary point at very low typicality scores where so few songs exist. Instead, if they were to make the case for typicality having a negative linear effect as in shown in their model #3, then they would have much stronger evidence to support that.

6.3.2 Genre Sensitivity Analysis

Along with the typicality range, another concern of ours was that Askin & Mauskapf may have placed too much weight on the significance and magnitude of the typicality feature, when these aspects of that measure could be significantly impacted by a small change in the measure's definition. To test the sensitivity of the typicality feature, we computed typicality scores for each song using another genre definition based on the Spotify tags as was outlined in subsection 3.4.2. Recall that genre is used to define the weights for the typicality measure. Figure 6.1 shows the typicality values for the Spotify tags in blue, and we can observe that its distribution is nearly identical to the Discogs distribution in red. While the typicality distributions are equal, the distributions have different genre categories, and we did not allow songs to have multiple genres in our experiments in chapter 5, and so that carries over here for the Spotify genre definition; also, this resulted in the need for a baseline genre, which we chose as hip hop.

We can observe the differences between the models constructed using the separate genre definitions in Table 6.1. Models #1 and #2 are based on the Discogs genres and models #3 and #4 are based on the Spotify genres. Comparing models #1 and #3, we can observe how the typicality feature becomes much less significant, whereas, models #2 and #4 have the similar typicality coefficients and significance levels. Other features have some changes in their significance as well like speechiness and having 10+ past hits as shown in Table C.7, but this can be explained by the baseline genre, hip hop, which corresponds to songs with low speechiness scores. While there might be some criticism of our choice to make the genres not have multiple levels, these results suggest that typicality is not necessarily as robust as one might think. Especially in model #3, its effect is comparable to other features like danceability instead of being this highly important feature with a significant and large effect.

6.3.3 Predictive Margins

We were not able to follow Askin & Mauskapf's approach for plotting the marginal probabilities, but even if we were, we would be unsure if this validates the hypothesis. The procedure for generating the marginal probability plot starts by training an ordered logit model using all of the data, then by re-fitting the data to the model to predict each song's probability of belonging to each peak position value from 1-100, and finally by aggregating the probabilities based on peak position intervals.

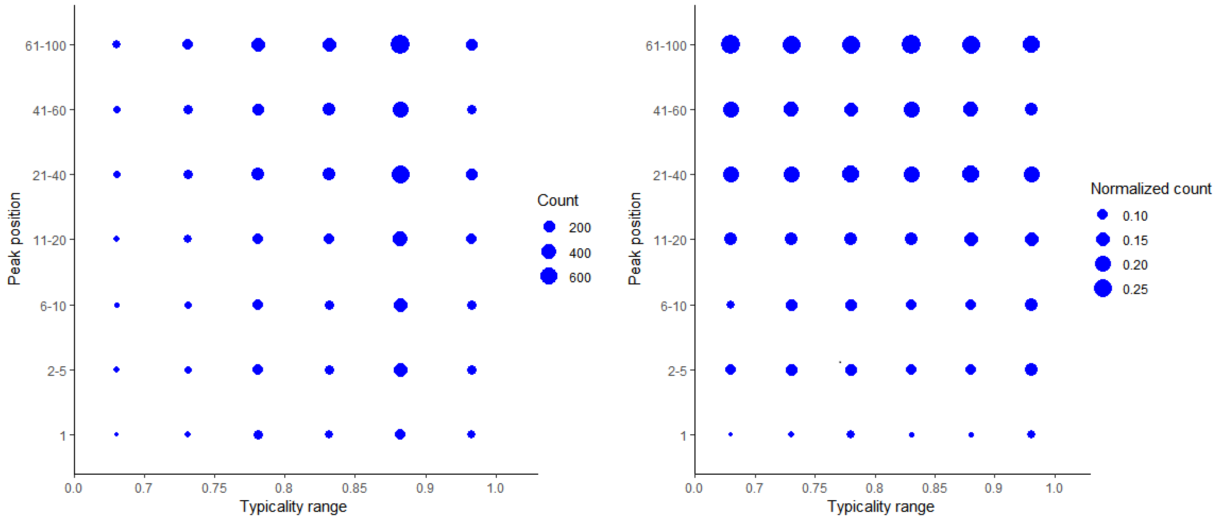


Figure 6.3: Proportion of songs from complete data within a specific peak position interval and typicality range. (Left) Count. (Left) Normalized count.

We have three concerns about drawing conclusions from this plot. We already described the first one in subsection 6.3.1 as there are too few observations to make inference about typicality values below 0.7. Half of each curved model fit in Figure A.2 is based on songs more than three standard deviations below the mean typicality value. Of course the distribution is not normal; it is bi-modal, but still, it is a very small proportion of the songs to make such a large claim about. If we were to draw an inference from Figure A.2, then it would be based on the songs with typicality scores around 0.7 to 0.9 because that is where most songs fall on the typicality distribution. It would also point towards a negative linear trend between typicality and song performance because the songs that lie within this range follow a positive trend when they have high peak positions and a negative trend when they have low peak positions. More simply, Askin & Mauskapf's work shows that as songs become more typical, they are less likely to succeed on the charts.

The second concern is that any trend fit to this data may not be because of an underlying relationship between typicality and peak position but because typicality has a bi-modal distribution. This is more of a concern for our marginal probability re-implementation than their own model as our stationary points in the polynomials are at the local minimum in the typicality distribution, whereas, theirs are much lower. As there will be less songs with typicality values around 0.8, it is reasonable to assume that there will be a reduced spread of the data around this area, which could explain the curve fitting in Figure 6.2.

Our third concern is that typicality may not be that important in the end. Yes, there may be a significant effect, but the majority of songs have typicality values within a 0.2 range, which makes their effect look a lot larger in Table 6.1 as that is the log odds ratio for one unit of change. We can observe the distribution of typicality scores over different peak position ranges in Figure 6.3, and what we see is there is not much change in peak position proportions across typicality values. Songs within different typicality ranges have closely related proportions of hits, flops, and in between.

6.4 Conclusion

Askin & Mauskapf argued that there exists a relationship between typicality and the likelihood of song success. In this chapter, we compared their experimental design choices with our own, re-implemented their study using the features and data that we had access to, and critically examined their claims around the importance of typicality by looking at its distribution, sensitivity, and marginal probability model fits. We argue that their claims should be re-examined because some of their hypotheses were only validated using weak evidence. This might suggest that typicality as a feature is not as important as was originally imagined.

Chapter 7

Time series analysis

In chapter 5, we looked at modelling songs, and in chapter 6, we analyzed the typicality feature through its ability to be modelled. While modelling has been the recurrent theme throughout this thesis, there are also other applications for this song data. When we designed and evaluated our modelling tasks, we controlled for the temporal effects of songs being released over different time periods. Instead of controlling for these effects, one could use them to study how song features change over time. For example, as a new genre emerges onto the charts and becomes popular, the makeup of the charts will change with more songs from the new genre on the charts, and as a result, a new set of average feature values will come to represent the charting songs.

Over the lifespan of the Billboard Hot 100 weekly chart, there have been many genres that have risen and fallen like rock, disco, and rap. It would be interesting to see how songs from these genres first present themselves on the charts and how they interact with songs from other genres as their genres become more established. We also know from our past models how important the time blocks were, so it makes sense to look at how other features have changed with respect to time. In this chapter, we represent our features as time series and apply simple time series analysis techniques to explore possible ways that the song data could be analyzed more thoroughly in the future.

7.1 Temporal variation

To study the temporal variations present in song features, one possible way to represent that data would be to represent the features as time series, so for example, the valence feature would have a time series associated with it, which could tell us if music has become more positive or negative over time. Note that this question could be refined to only examine songs with particular traits like being played in some key or belonging to some genre. The valence time series could be compared with other related features like a topic mixture for heartbreak, $T_{80,45}$, and this could then be used to draw deeper insights into how features relate to each other. We construct feature time series by calculating the average feature values for all of the songs on the charts on a given week and then mapping those values to each time point.

Previously, in chapter 2, we described a paper by Serrá et al. [27], where they explored how pitch, timbre, and loudness vary over time in songs from the MSD. While The Echo Nest features are summary measures for acoustic features, we can still use them to see how songs have changed at a high level. For our models in chapter 5, we included the features valence, danceability, and acousticness in the shortened statistical summary tables because of their consistent significance and relatively strong effects. In Figure 7.1, we can see how that have changed as they have been represented by time series where each time point corresponds to the average feature value for all of the songs charting on a given week.

From Figure 7.1, we can observe how both features have modest changes through time with danceability values increasing and valence values decreasing, and in the summary tables from chapter 5, hit songs were often associated with being more negative (low valence) and more danceable (high danceability). This might offer some evidence of artists and labels gravitating towards the more successful formula of songs being danceable and less upbeat.

There is a much larger change in acousticness values over time. The average acousticness value has a sharp decline from 1958 to the 1980s and then it stabilizes at those lower levels for the rest of the time series. This could be explained by the emergence of new genres and musical technologies, which are less acoustic, and for our models in chapter 5, what this explains is why the acousticness feature had less consistent effects. Across the models, its effects varied from positive to negative, large to small, and significant to not significant. Because of its rapid fall in the first half of the time series, comparing songs from different time periods could mean comparing songs with very different acousticness values.

What this shows is how The Echo Nest features have changed, and how we can generate hypotheses around those changes that could be validated with more clearly defined modelling tasks even though in this chapter that is what we are trying to shift away from. Having more data is always useful, and it could be that the 75% of Billboard Hot 100 songs absent from our dataset makes acousticness stable and danceability and valence more variable, but we can only draw conclusions from the data that we have access to, and we have to assume that the missing data is drawn from the same sample as our data.

Previously, we described a scenario where a new genre emerges on the charts and

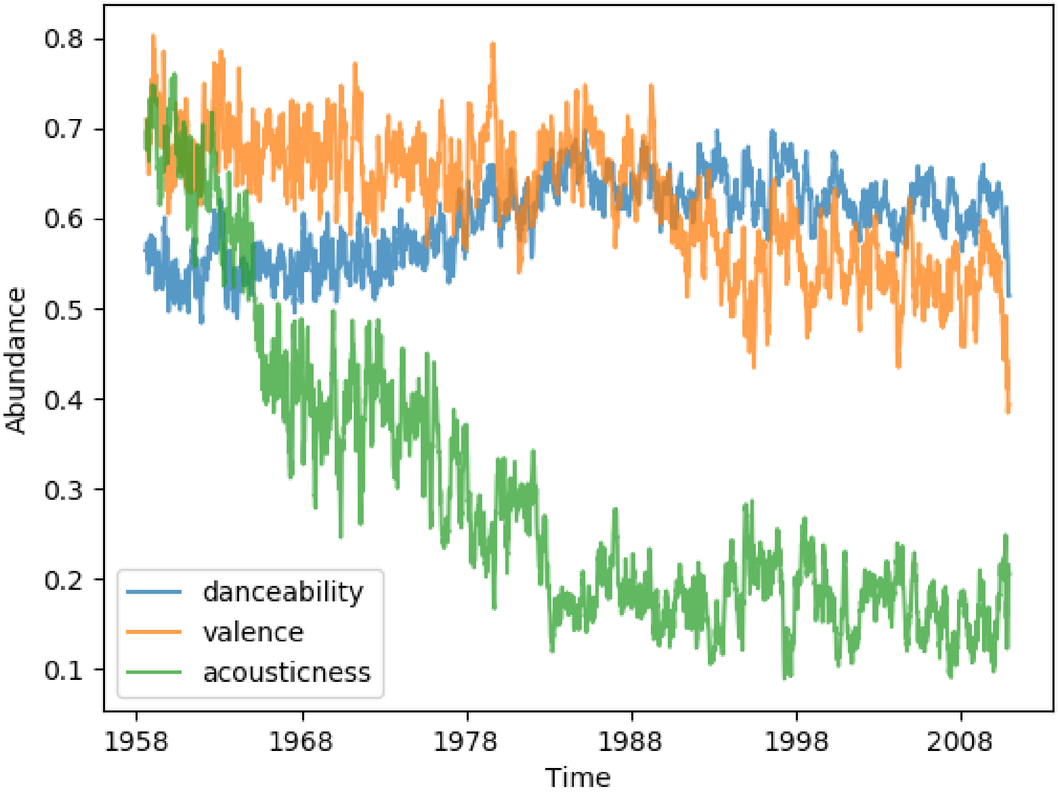


Figure 7.1: Select The Echo Nest feature time series for songs from the complete dataset.

displaces other genres. Just like with The Echo Nest features, we can look at how genres have varied through time, and this can provide us with a better understanding as to why other features change because certain genres are more likely to be associated with specific features ranges like rap and high speechiness values. Recall from chapter 5, how there was a profane language topic mixture, $T_{10,4}$, which was significant in some of the later models. We hypothesized that $T_{10,4}$ might correspond to rap because of the inclusion of a racial term in its top ten most probable words. We can perform an ad hoc test to see if it is related to our rap genres, hip hop and rap, by plotting their time series and looking for shared patterns. As was explained in chapter 3, we have two rap genres because the Spotify tags associated with hip hop were not close enough to those associated with rap at the level of granularity we chose to define genre with.

Figure 7.2 shows the time series plotted together. We can observe how $T_{10,4}$ closely resembles the genres, which offers evidence in favour of the hypothesis that the topic mixture is related to rap. Interestingly, rap and hip hop correspond to different periods in the rap timeline. Spotify used hip hop tags to describe rap from the 1980s and 1990s, while it used rap tags to describe contemporary rap songs. Another observation is that when we compare Figure 7.2 with Figure 7.1, we can see that acousticness reaches its floor value and stable state when rap becomes popular. An interesting future area of work could look at how other features change within the rap genres over the transition period from hip hop to rap; this was previously touched on by Johnson-Roberson & Johnson-Roberson[17].

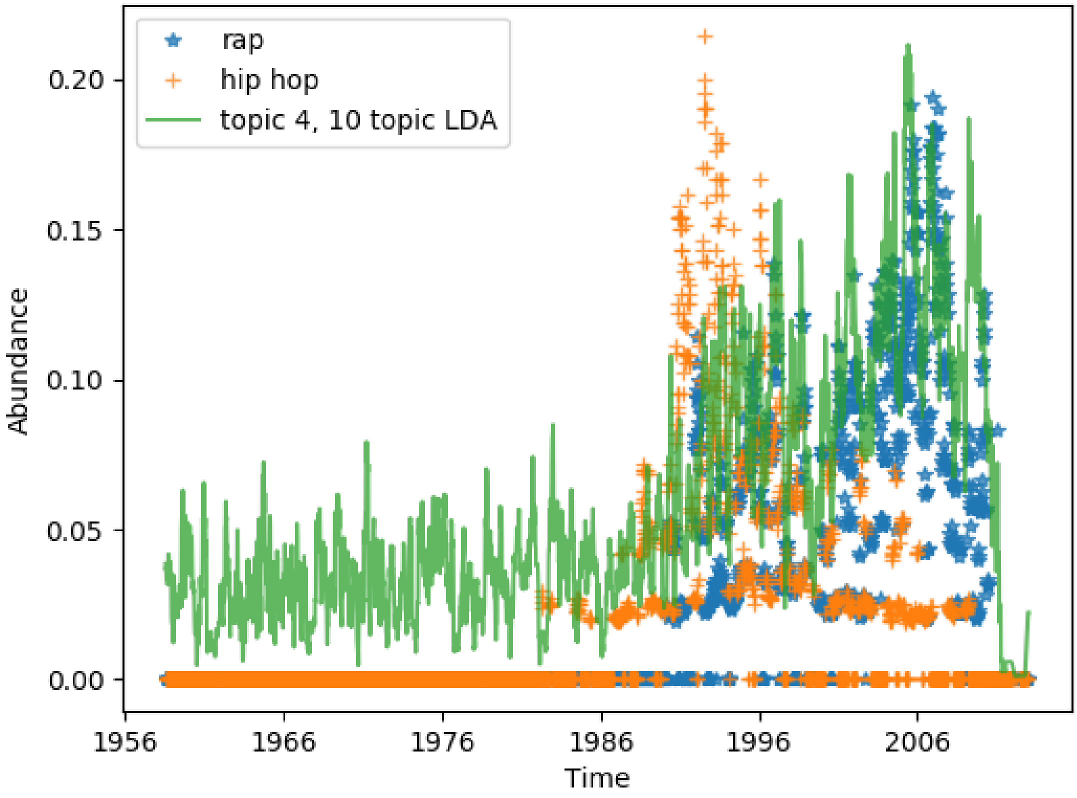


Figure 7.2: Temporal variations of profane language topic, $T_{10,4}$, with Spotify rap genres in complete dataset.

7.2 Topic mixtures

More generally, lyric topic mixtures did not contribute much to our models. They were not very useful for distinguishing chart trajectory classes. Only a few topics like the profane language and baby talk topics had consistent effects in multiple models. For topics with fewer distinct most probable words, it was difficult to track them across differently-sized topic models. Previously, we showed how topics can splinter into more specific topics, which can share a common theme like in chapter 3 or be distinct like in chapter 5. Instead of comparing the most probable words between two topics from separate models, an alternative way of finding related topics might involve looking at how the topics' time series correlate with each other.

In chapter 3, we presented an example of a topic, $T_{10,0}$, which referred to nature, splitting into two specific topics, $T_{20,15}$, which referred to weather and active movements, and $T_{20,18}$, which referred to the sky. Figure 7.3 shows their time series representations. We can observe a shared temporal trend between $T_{10,0}$ and $T_{20,18}$, and if we calculate Pearson's correlation between $T_{10,0}$ and all of the topics from LDA with 20 topics, we find that only three topics have Pearson correlations above 0.1. $T_{20,15}$ and $T_{20,18}$ have correlations with $T_{10,0}$ of 0.661 and 0.244 respectively. The other topic not shown in the figure is $T_{20,14}$, which refers to people and religion, and it has a correlation of 0.415. By aggregating these topics together, they have a correlation with $T_{10,0}$ of 0.756; this suggests that the topics are related and that these topics from LDA with 20 topics are $T_{10,0}$'s descendants.

$T_{20,14}$, $T_{20,15}$, and $T_{20,18}$ essentially form a group because of their high correlation with $T_{10,0}$, and one can imagine that similar relationships are present among other topics. One way to further explore these relationships across models at a larger scale would be to cluster the topics using a simple clustering algorithm. In this way, we would be able to see how topics are related to each other, and it would be easier to find possible splintered topics when comparing feature sets with different topic models.

While topic model features were employed in some of the models, the models with more topics were not favoured because they had too many features and lead to overfitting. Instead of modelling, if one is interested in exploratory analysis, then having more topic mixtures is beneficial because it allows us to look at more relationships and the data will be represented at a finer level of granularity. Clustering offers a scalable way to group related topics together.

We demonstrate this by applying k-means to the 80 topic mixtures from LDA trained on the complete dataset lyrics. Figure 7.4 shows nine clusters of topic mixture time series, five of which correspond to large groups and four of which correspond to outliers, one being the profane language topic. These clusters can be compared based on a variety of time

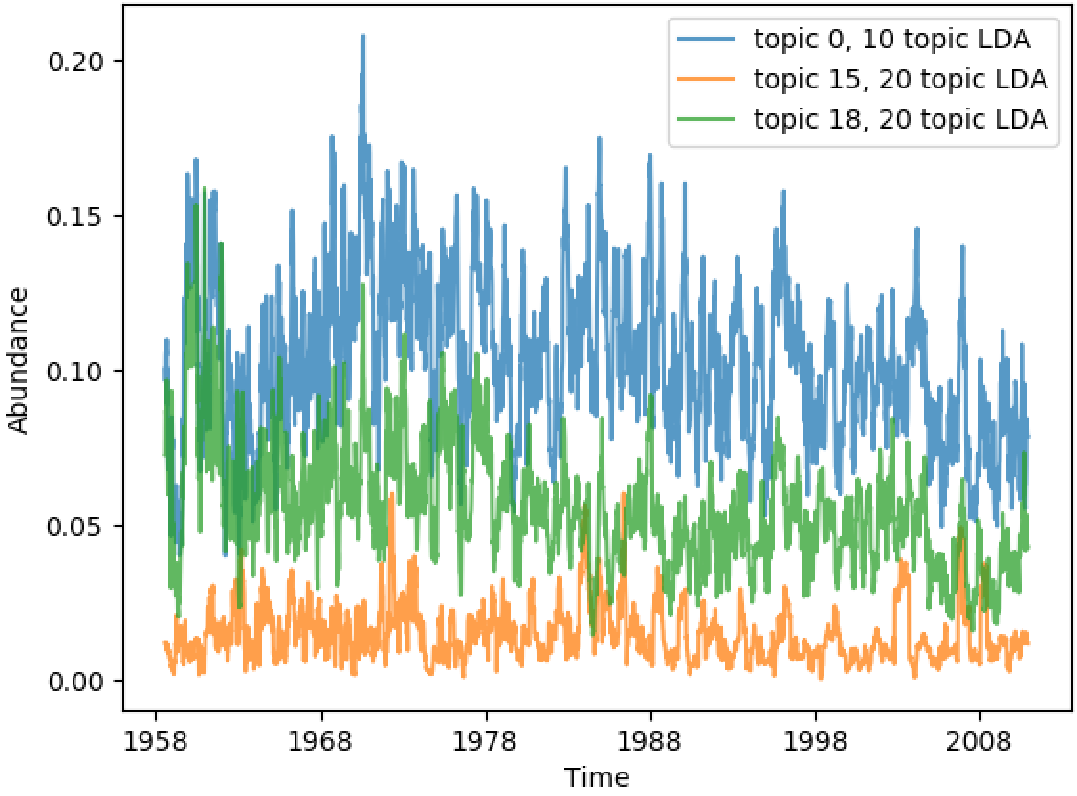


Figure 7.3: Correlated topic mixture time series with shared most probable words from separate LDA models trained on complete dataset lyrics corpus.

series metrics like their mean and variance, their long-term trends, the presence of sharp peaks, or more traditional time series measures like their seasonality and time lags.

For the more traditional measures, one needs to first establish stationarity, that is, fixed mean and variance through time, which definitely disqualifies the profane language topic, $T_{80,69}$. One way of testing this is by performing a unit root test where the null hypothesis is defined as the root's presence, and the alternative hypothesis is defined as the lack of a root and stationarity for the time series. One type of unit root test is the adjusted Dicky-Fuller test [25], which incorporates a lag term to test varying-length auto-regressive processes. These processes are evaluated against each other using a model selection criterion like AIC.

We apply this test to the individual topic mixture time series to find non-stationary

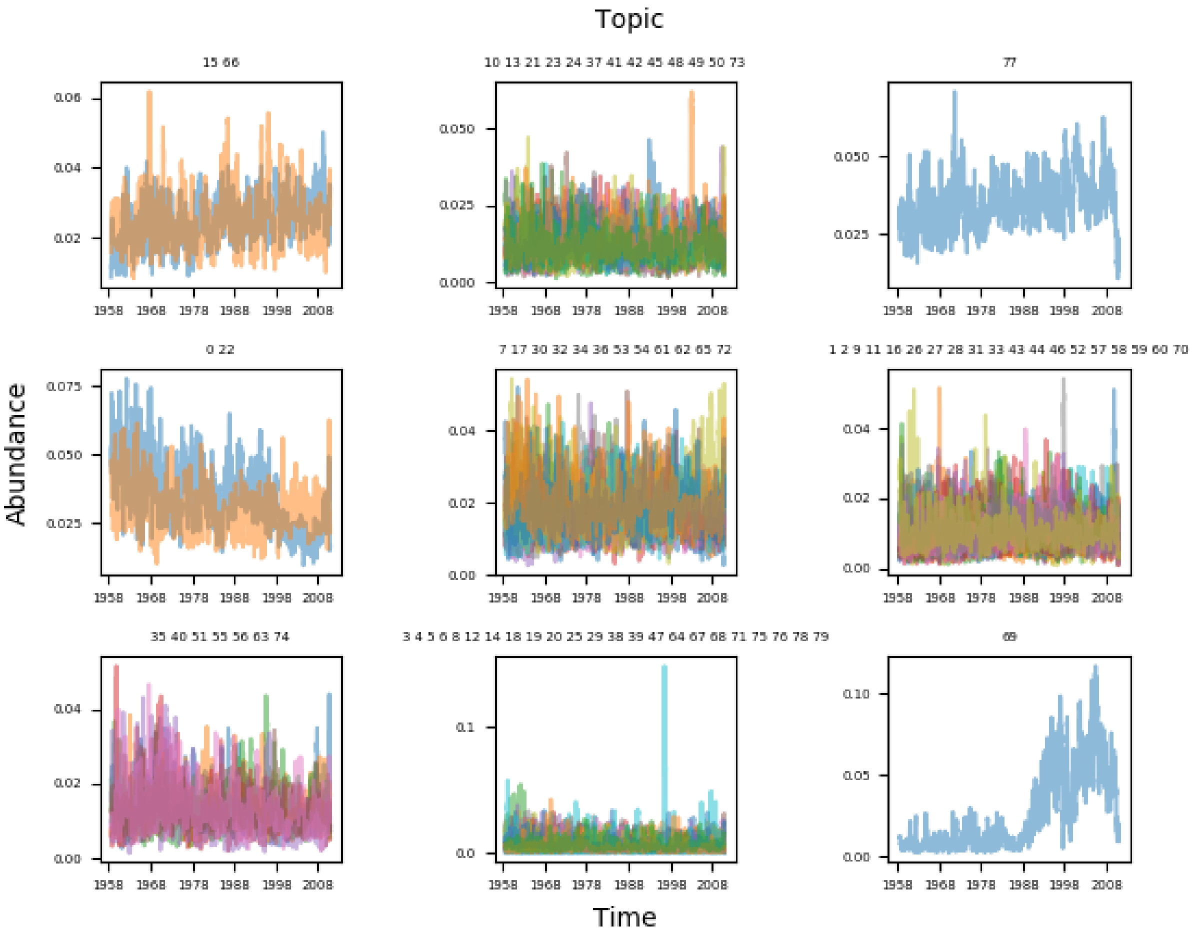


Figure 7.4: Clustering of 80 topic mixtures from LDA trained on complete data lyrics corpus using k-means.

time series in the topic mixtures though it could also be applied to average time series signals for each cluster. As we are looking for any significant results among the topics, we use the Bonferroni correction [6] to adjust our testing for multiple comparisons, so a p-value of 0.05 becomes 0.0016 as there are 80 time series being tested. We use a maximum lag of 52 weeks and use AIC to select the most likely auto-regressive model. We find three time series that cannot confirm the null hypothesis; Figure 7.5 shows them. All three time series can be characterized by at least two separate periods of behaviour. It may seem odd that we were able to reject the null hypothesis for some time series with sharp peaks. This is because over the 2,734 weeks of data, having one sharp peak is not sufficient evidence to reject the null hypothesis; however, in principle, the time series with sharp peaks should not be considered stationary.

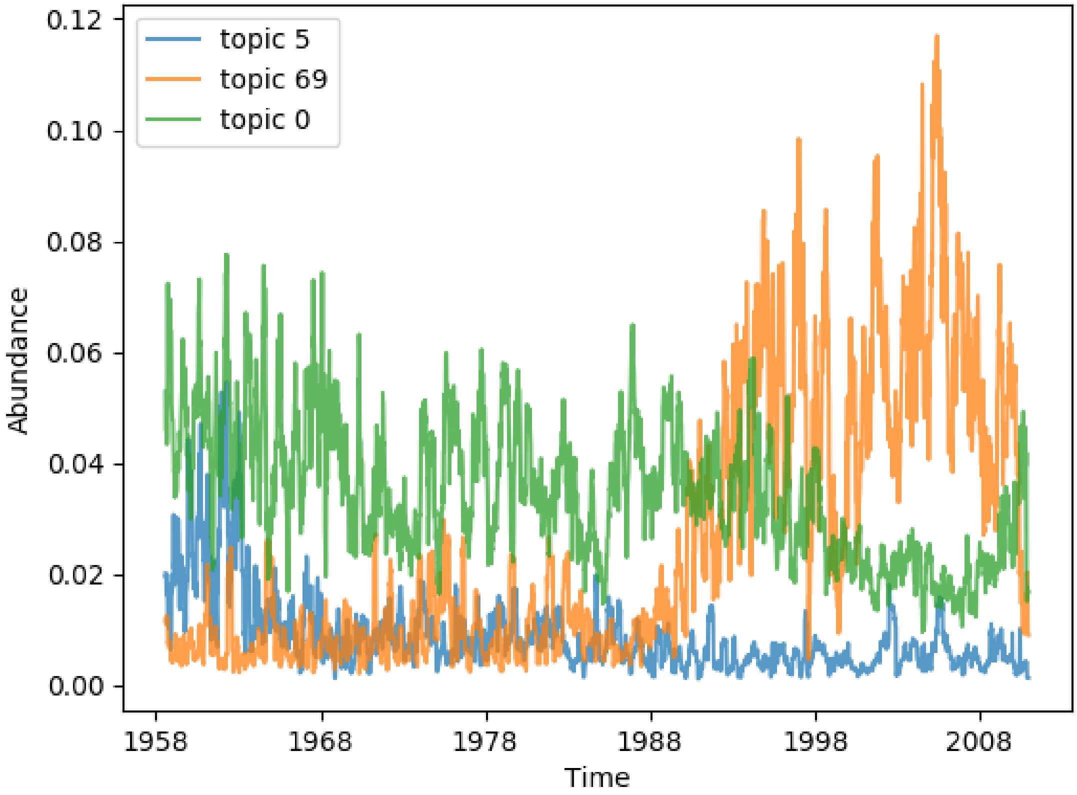


Figure 7.5: Non-stationary topic mixture time series from LDA with 80 topics trained on complete dataset lyrics corpus that failed the adjusted Dicky-Fuller Test.

Since we have established stationarity in 77 of the 80 topic mixtures, they can then be tested for things like time lags, where one looks at how one time series lags behind another based on some time interval and seasonality, where one looks at how recurrent changes in a time series follow a periodic trend. We leave these topics for future researchers, but we hope that this chapter provided some ideas for uses of our data outside of modelling.

7.3 Summary

In this chapter, we touched on a few time series analysis techniques of varying complexity from manual inspection to unit root tests to highlight ways that our data can be used for exploratory analysis.

Chapter 8

Conclusion

In this thesis, we set out to learn what makes a song successful and how different song attributes are related to song performance. We proposed two new target variables for modelling song performance that represent a song's position through time on the charts, which correspond to how long a song is in the minds of many listeners and how much reach a song has over a population of listeners. The target variables were constructed from chart data from the weekly Billboard Hot 100 chart using binning and alignment methods.

We attempted to model these target variables using a feature set composed of high level audio, lyric, and metadata features comprised of The Echo Nest features, LDA applied to lyrics from the MSD, and the weekly Billboard Hot 100 chart data. For many of these features, we followed the same derivation procedures as outlined by Askin & Mauskapf in their study on optimal differentiation [1]. While they were most interested in answering the question of how a song performs relative to its neighbors based on the level of distinctiveness in the song's features as represented by a typicality measure, our analysis goals were more broad; we wanted to justify the use of our target variables and find relationships between specific chart trajectory patterns and our song features. We were able to show that modelling songs based on higher level target variables that incorporate both time and magnitude is necessary because when modelled alone, these aspects of chart performance convey different messages about the importance of certain song features. We were also able to find some relationships between trajectory patterns and song features, but these were often with metadata features, which are baseline song attributes. We had hoped to find more links between trajectories and audio and lyric features as these features represent parts of a song that an artist can influence, but these links were not observed with the same strength as some metadata links. This does not mean that the metadata links had no value. One metadata feature, the time block, controlling for a song's release date, had a consistently strong and significant effect across all of our models where it was included, indicating that the song chart trajectories that defined success and failure in one era were not necessarily the same as those found in another.

There are some limitations with our work, which we hope can be resolved by future researchers who decide to take on this problem. We already outlined a few recommendations in chapter 7 where we considered how features vary through time and what time series analysis approaches could be taken to explore these variations. The most basic limitations were that our data's coverage was too low and our features were too high level. These realities affected our ability to draw wider conclusions and develop models with more predictive power. A more fundamental limitation with our experimental design was that by using so many features, we were not able to compare more niche chart trajectory patterns like the oscillatory trajectory in cluster 13 from Figure 4.10 because the models would not converge due to their small sample sizes. We had to limit our analysis to large or aggregate classes. Developing different control sets for different subsets or performing feature selection might be worthwhile to analyze these smaller datasets in the future.

Concerns could also be raised about how our data is drawn from a wider population. All of the songs we analyzed made an appearance on the Billboard Hot 100, which is one of the most competitive music charts; a song must have sales in the United States in the top 100 in order to make it on this chart; thus, out of the set of all songs created and released commercially, our models and findings were only based on a small subset. With that said, it is important to reiterate that our intent was not to build a universal classifier but a model that could identify historical differences between hits and flops on the Billboard Hot 100, which would be useful for musicologists interested in studying the evolution of popular music.

Another problem with our dataset is that we only had a few control variables to use for distinguishing songs, and as a result, it is possible that our models could lock in bias against songs based on a confounding variable. For example, if a lead single and an album filler were both released by popular artists and had all the same baseline attributes, then our models would assume that they are similar even though their marketing strategies would likely be quite different. Consider that they have one song feature difference, which is their acousticness value; the single has a high value, while the filler has a low value. The single gets focused support from its artist's label and lasts a long time, while the filler only lasts a few weeks. Suddenly, our model learns that high acousticness values correspond to hits and low values correspond to flops even if acousticness does not have any true relationship to song performance. The model will be biased against future songs with low acousticness values even if they are lead singles. In general, minimizing this bias means developing more control variables.

We chose to represent the question of finding relationships between song attributes and song performance as a classification problem, while Askin & Mauskapf represented their related question as a regression problem [1]. When we tried to recreate their study, we found several areas of concern, which included the robustness and sensitivity of their results. In our own study, we also had some concerns over interpreting our models. We considered both two-class and multiclass models for tasks designed to answer specific questions. While the two-class model results were strong, the multiclass models were fragile. Unfortunately, this is not what we wanted to observe as part of the argument for considering more complex target variables was to move past the hit-flop paradigm. With that said, considering a more complex target variable may still have its uses. As we are able to represent more distinct chart behaviours, we can distinguish more niche groups of songs. Also, we observed that some features had contrasting effects when modelled along separate dimensions of song performance. What this indicates is that song performance is a complex concept that can be better represented by a target variable that explicitly takes into account multiple aspects of a song's performance, which is important to better understand past music consumption patterns for the cultural study of music and the forecasting of future musical trends.

References

[1] Noah Askin and Michael Mauskapf. What makes popular culture popular? product features and optimal differentiation in music. *American Sociological Review*, 82(10):910–944, 2017.

[2] Jack Atherton and Blair Kaneshiro. I said it first: Topological analysis of lyrical influence networks. In *ISMIR*, pages 654–660, 2016.

[3] Jonah Berger and Grant Packard. Are atypical things more popular? *Psychological Science*, 29(7):1178–1184, 2018.

[4] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, pages 591–596, 2011.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(1):993–1022, 2003.

[6] Carlo Emilio Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

[7] Ruth Dhanaraj and Beth Logan. Automatic prediction of hit songs. In *ISMIR*, pages 488–491, 2005.

[8] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *COLING*, pages 620–631, 2014.

[9] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. 12. Springer Series in Statistics, 2 edition, 2009.

[10] Toni Giorgino. Computing and visualizing dynamic time warping alignments in *R*: the *dtw* package. *Journal of Statistical Software*, 31(7):1–24, 2009.

[11] Bianca Gracie. We need to stop whitewashing dancehall music in 2016. *Fuse TV*, 2016.

[12] Allen Guo. Billboard.py. https://github.com/guoguo12/billboard-charts, 2018.

[13] Xiao Hu, J Stephen Downie, and Andreas F Ehmann. Lyric text mining in music mood classification. In *ISMIR*, pages 411–416, 2009.

[14] Xiao Hu, J Stephen Downie, Cyril Laurier, Mert Bay, and Andreas F Ehmann. The 2007 MIREX audio mood classification task: Lessons learned. In *ISMIR*, pages 462–467, 2008.

[15] Myra Interiano, Kamyar Kazemi, Lijia Wang, Jienian Yang, Zhaoxia Yu, and Natalia L Komarova. Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society Open Science*, 5(5), 2018.

[16] Molly E Ireland and James W Pennebaker. Language style matching in writing: Synchrony in essays, correspondence, and poetry. *Journal of Personality and Social Psychology*, 99(3):549, 2010.

[17] Cora Johnson-Roberson and Matthew Johnson-Roberson. Temporal and regional variation in rap lyrics. In *NIPS Workshop on Topic Models: Computation, Application and Evaluation*, 2013.

[18] Corey Kereliuk, Bob L Sturm, and Jan Larsen. Deep learning and music adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, 2015.

[19] Cyril Laurier, Jens Grivolla, and Perfecto Herrera. Multimodal music mood classification using audio and lyrics. In *ICMLA*, pages 688–693. IEEE, 2008.

[20] Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. In *ISMIR*, 2003.

[21] Beth Logan, Andrew Kositsky, and Pedro Moreno. Semantic analysis of song lyrics. In *ICME*, pages 827–830. IEEE, 2004.

[22] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[23] Babak Naimi, Nicholas AS Hamm, Thomas A Groen, Andrew K Skidmore, and Albertus G Toxopeus. Where is positional uncertainty a problem for species distribution modelling? *Ecography*, 37(2):191–203, 2014.

[24] François Pachet and Pierre Roy. Hit song science is not yet a science. In *ISMIR*, pages 355–360, 2008.

[25] Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

[26] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[27] Joan Serrà, Álvaro Corral, Marián Boguñá, Martín Haro, and Josep Ll Arcos. Measuring the evolution of contemporary western popular music. *Scientific Reports*, 2:521–526, 2012.

[28] Uri Shalit, Daphna Weinshall, and Gal Chechik. Modeling musical influence with topic models. In *International Conference on Machine Learning*, pages 244–252, 2013.

[29] Abhishek Singhi. Lyrics matter: Using lyrics to solve music information retrieval tasks. Master's thesis, University of Waterloo, 2015.

[30] Abhishek Singhi and Daniel G Brown. On cultural, textual and experiential aspects of music mood. In *ISMIR*, pages 3–8, 2014.

[31] Ben Sisario. Lil Nas X added Billy Ray Cyrus to 'Old Town Road.' is it country enough for Billboard now? *The New York Times*, page 1A, April 6, 2019.

[32] Lucas Sterckx, Thomas Demeester, Johannes Deleu, Laurent Mertens, and Chris Develder. Assessing quality of unsupervised topics in song lyrics. In *European Conference on Information Retrieval*, pages 547–552, 2014.

[33] Bob L Sturm. An analysis of the GTZAN music genre dataset. In *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, pages 7–12, 2012.

[34] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[35] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.

[36] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002.

[37] Todd Williams. How Run-D.M.C's 'Raising Hell' launched hip-hop's golden age. *The Boombox*, 2016.

[38] Yunqing Xia, Linlin Wang, and Kam-Fai Wong. Sentiment vector space model for lyric-based song sentiment classification. *International Journal of Computer Processing of Languages*, 21(4):309–330, 2008.

[39] Yi-Hsuan Yang, Yu-Ching Lin, Heng-Tze Cheng, I-Bin Liao, Yeh-Chin Ho, and Homer Chen. Toward multi-modal music emotion classification. *Advances in Multimedia Information Processing-PCM 2008*, pages 70–79, 2008.

APPENDICES

Appendix A

Optimal differentiation figures and tables

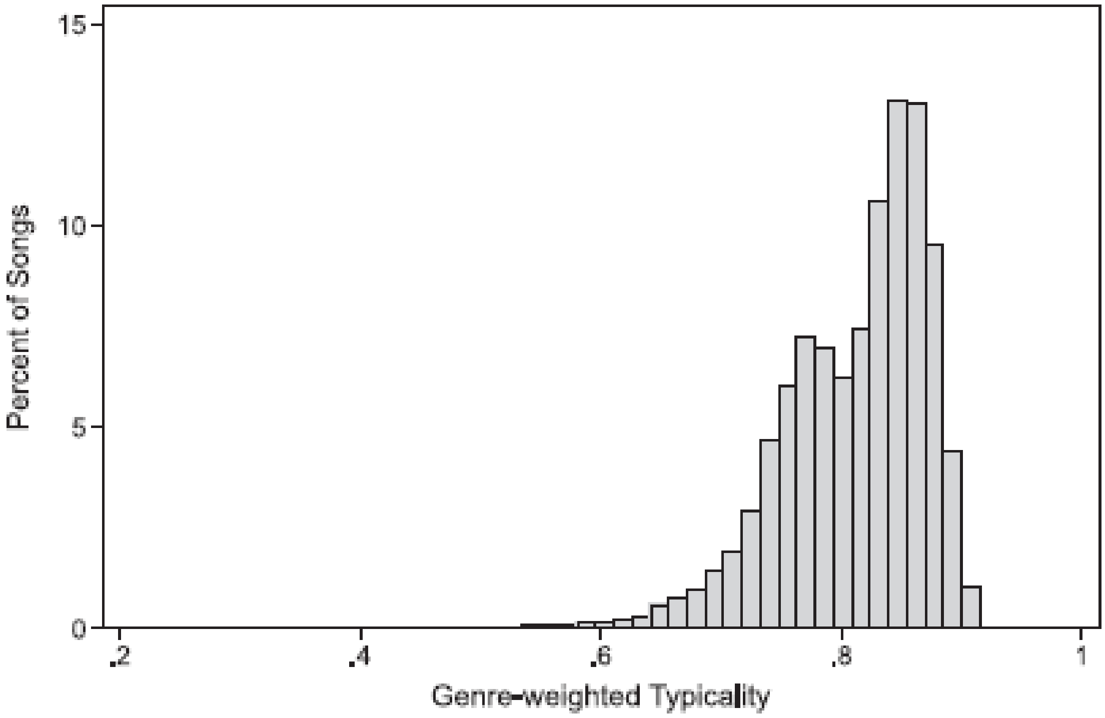


Figure A.1: Distribution of genre-weighted song typicality (yearly), taken from [1].

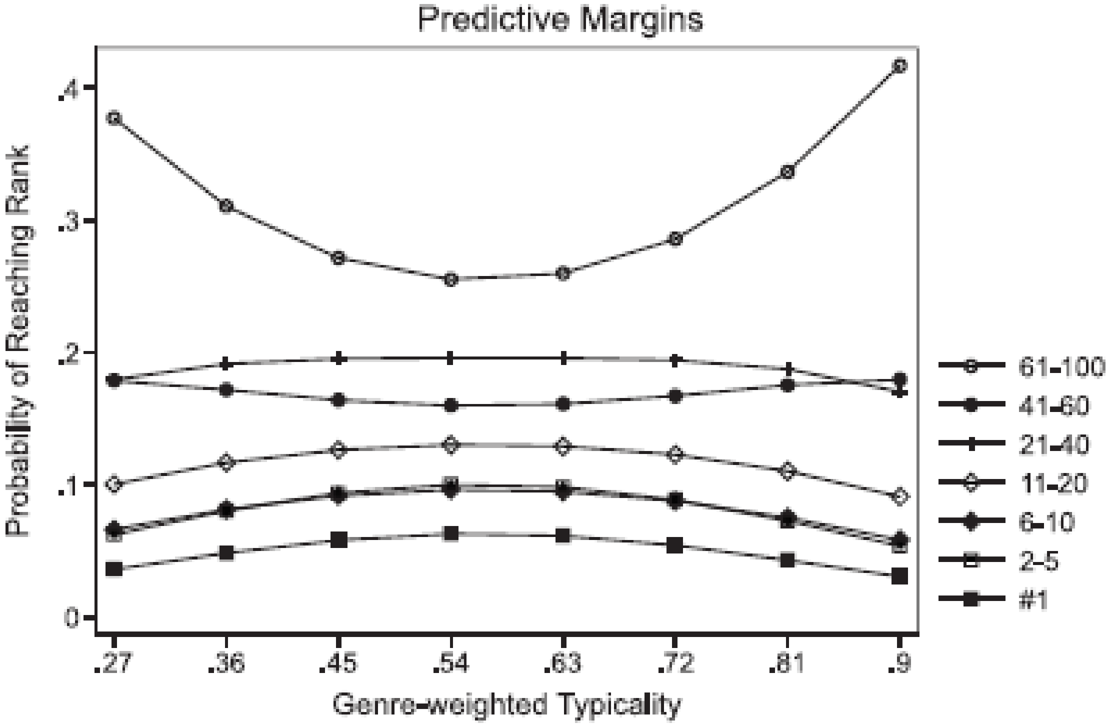
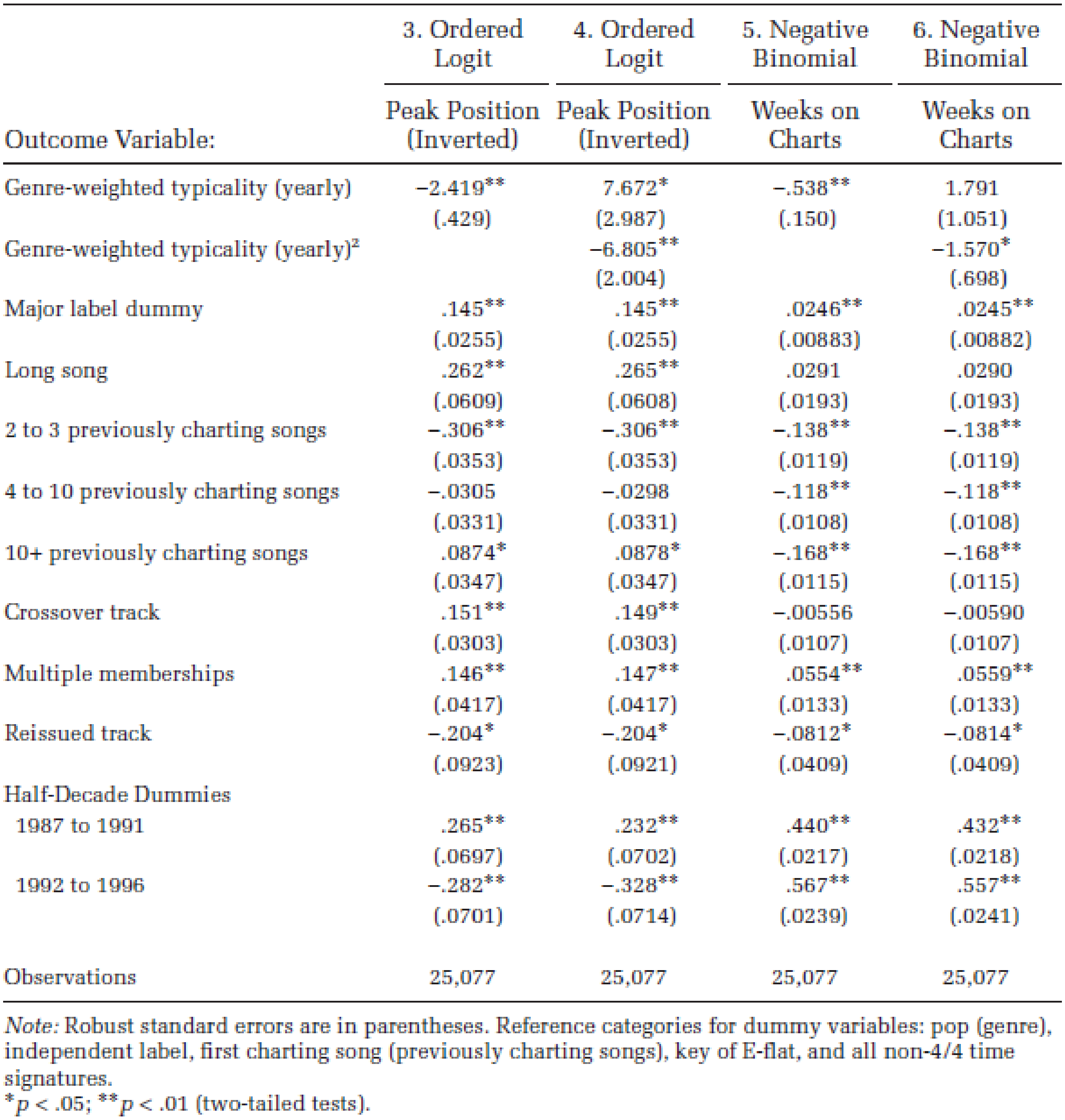


Figure A.2: Predicted marginal probability of songs achieving selected peak positions (by typicality) from ordered logit model (model 4), taken from [1].

Table A.1: Select variables from pooled, cross-sectional ordered logit and negative binomial models predicting *Billboard* Hot 100 peak chart position and longevity, 1958 to 2016, taken from [1].



	3. Ordered Logit	4. Ordered Logit	5. Negative Binomial	6. Negative Binomial
Outcome Variable:	Peak Position (Inverted)	Peak Position (Inverted)	Weeks on Charts	Weeks on Charts
Genre-weighted typicality (yearly)	–2.419**	7.672*	–.538**	1.791
	(.429)	(2.987)	(.150)	(1.051)
Genre-weighted typicality (yearly)^2		–6.805**		–1.570*
		(2.004)		(.698)
Major label dummy	.145**	.145**	.0246**	.0245**
	(.0255)	(.0255)	(.00883)	(.00882)
Long song	.262**	.265**	.0291	.0290
	(.0609)	(.0608)	(.0193)	(.0193)
2 to 3 previously charting songs	–.306**	–.306**	–.138**	–.138**
	(.0353)	(.0353)	(.0119)	(.0119)
4 to 10 previously charting songs	–.0305	–.0298	–.118**	–.118**
	(.0331)	(.0331)	(.0108)	(.0108)
10+ previously charting songs	.0874*	.0878*	–.168**	–.168**
	(.0347)	(.0347)	(.0115)	(.0115)
Crossover track	.151**	.149**	–.00556	–.00590
	(.0303)	(.0303)	(.0107)	(.0107)
Multiple memberships	.146**	.147**	.0554**	.0559**
	(.0417)	(.0417)	(.0133)	(.0133)
Reissued track	–.204*	–.204*	–.0812*	–.0814*
	(.0923)	(.0921)	(.0409)	(.0409)
Half-Decade Dummies				
1987 to 1991	.265**	.232**	.440**	.432**
	(.0697)	(.0702)	(.0217)	(.0218)
1992 to 1996	–.282**	–.328**	.567**	.557**
	(.0701)	(.0714)	(.0239)	(.0241)
Observations	25,077	25,077	25,077	25,077

Note: Robust standard errors are in parentheses. Reference categories for dummy variables: pop (genre), independent label, first charting song (previously charting songs), key of E-flat, and all non-4/4 time signatures.
$^{*}p < .05$; $^{**}p < .01$ (two-tailed tests).

Appendix B

Top five most probable words for each topic

Table B.1: Top five most probable words for each topic from each LDA model.

Topic #	10 topic	20 topic	40 topic	80 topic
0	light, rain, feel, like, fire	heart, cri, whi, say, tri	thing, still, chang, time, wrong	love, heart, true, abov, mine
1	love, need, give, heart, littl	said, littl, big, back, home	keep, gotta, run, nothin, goin	way, chang, ani, say, pay
2	hey, danc, rock, shake, boy	want, let, give, back, show	away, whi, wonder, save, cold	better, home, late, sorri, train
3	know, say, tell, think, never	hey, danc, rock, shake, move	wanna, littl, man, bit, gotta	lay, ich, die, wenn, echo
4	like, get, got, back, know	love, kiss, heart, sweet, true	chorus, two, step, one, vers	free, realli, angel, mine, set
5	babi, want, let, come, wanna	littl, bit, que, readi, cha	well, walk, yes, crazi, gone	new, citi, best, king, record
6	round, que, boogi, around, world	need, one, life, onli, hold	big, man, old, well, got	back, bring, come, push, seen
7	time, day, one, away, life	girl, like, bad, crazi, ladi	peopl, soul, god, lord, die	hold, kiss, touch, arm, tight
8	yeah, get, got, gonna, man	know, wanna, tell, feel, good	dream, find, fall, light, shine	alright, true, wish, hard, easi
9	littl, said, home, old, man	like, get, got, nigga, back	feel, make, real, touch, like	miss, much, noth, someth, anyth
10		world, around, turn, look, fall	new, ride, citi, street, beauti	believ, tri, hurt, wrong, strong
11		night, stop, play, song, music	need, hold, kiss, arm, lover	live, without, fire, burn, readi
12		time, take, tonight, wait, mind	like, turn, move, around, round	enough, old, diamond, hell, gun
13		babi, come, make, honey, take	come, back, home, bring, jump	keep, yes, pleas, stay, darl
14		man, woman, peopl, god, lord	time, mind, mine, goe, fine	peopl, parti, cool, fun, meet
15		away, walk, rain, run, sunshin	hey, que, hot, mari, cha	caus, befor, lose, say, done
16		gonna, way, day, say, make	let, rock, roll, parti, beat	get, readi, clap, outta, knock
17		yeah, get, got, right, gotta	would, could, stop, wait, wish	babi, honey, come, caus, cmon
18		light, sky, sun, blue, come	get, bodi, readi, put, enough	boy, blue, ride, wild, white
19		would, never, could, ever, gone	say, call, talk, show, hear	sweet, woman, hot, lovin, sugar

20			fire, high, sky, fli, angel	wanna, rock, roll, rockin, night
21			one, onli, look, thing, come	daddi, kid, bye, school, mother
22			know, tell, think, realli, someth	said, would, rememb, knew, could
23			play, song, sing, music, hear	find, mayb, stand, sometim, somewher
24			said, would, could, never, made	want, bad, step, treat, need
25			yeah, good, hey, check, right	whoa, smile, hate, pray, laugh
26			better, tri, caus, nobodi, somebodi	take, tonight, slow, chanc, tomorrow
27			gonna, want, work, make, caus	leav, alon, left, home, behind
28			night, right, tonight, woman, shake	around, world, round, goe, town
29			got, danc, everybodi, walkin, girlfriend	walk, still, side, becaus, pretti
30			take, world, anoth, around, hand	time, mind, mine, lover, line
31			girl, bad, guy, boom, like	danc, play, music, beat, game
32			way, alway, long, ever, matter	life, everyth, happi, made, world
33			life, live, believ, everyth, without	could, even, bout, guess, talkin
34			never, give, leav, stay, pleas	feel, need, know, insid, real
35			love, sweet, heart, true, give	got, money, get, lot, honey
36			like, got, get, nigga, shit	one, onli, wait, two, minut
37			heart, cri, break, lone, tear	song, sing, hear, listen, play
38			day, boy, everi, ring, morn	fool, lost, found, sexi, blame
39			babi, honey, know, come, let	hey, ladi, sha, woah, mess
40				let, say, time, right, tell
41				high, fli, head, star, sky
42				gonna, well, work, might, midnight
43				girl, drop, like, guy, low
44				long, lone, somebodi, god, thank
45				heart, break, start, apart, broken
46				man, talk, bodi, understand, make
47				day, anoth, today, gimm, morn
48				yeah, jump, caus, right, yes
49				nothin, feelin, goin, doin, lookin
50				everi, say, doe, matter, word
51				tell, whi, wonder, ask, reason
52				thing, show, one, look, come
53				like, chorus, repeat, vers, look
54				night, dream, light, shine, sleep
55				lord, street, war, hair, children
56				rain, summer, wind, send, sunshin

57				forev, togeth, fall, last, magic
58				right, gotta, turn, loos, light
59				give, alway, promis, never, dot
60				someon, care, must, els, kind
61				big, car, shot, hous, boom
62				run, insid, cold, open, door
63				make, feel, sure, whatev, take
64				look, everybodi, boogi, sign, girlfriend
65				would, away, could, stay, day
66				hope, end, tear, year, learn
67				call, name, number, hear, mari
68				good, nobodi, fine, feel, like
69				like, get, got, nigga, shit
70				hand, put, face, place, young
71				stop, lie, friend, dont, troubl
72				never, ever, cri, goodby, made
73				come, shake, move, groov, jam
74				heaven, river, sea, water, mountain
75				que, amor, cuerpo, alegria, por
76				littl, bit, count, window, celebr
77				know, think, mean, realli, tell
78				soul, help, check, funki, dig
79				gone, crazi, real, drive, sinc

Appendix C

Summary model tables

Table C.1: Interval target variable logistic regression models for various feature sets.

	Dependent variable:					
	Interval target variable					
	(metadata)	(The Echo Nest)	(10 topic)	(20 topic)	(40 topic)	(80 topic)
constant	-0.322 (0.482)	-1.144 (1.051)	-1.854 (1.203)	-1.288 (1.294)	-0.391 (1.610)	-0.934 (1.546)
long song	-0.914^{***} (0.341)	-0.873^{**} (0.362)	-0.832^{**} (0.366)	-0.831^{**} (0.378)	-0.818^{**} (0.388)	-0.997^{**} (0.412)
2-3 hits	-0.864^{***} (0.167)	-0.760^{***} (0.211)	-0.845^{***} (0.197)	-0.805^{***} (0.220)	-0.792^{***} (0.222)	-0.724^{***} (0.232)
6-10 hits	-0.634^{***} (0.160)	-0.484^{**} (0.227)	-0.608^{***} (0.204)	-0.578^{**} (0.236)	-0.570^{**} (0.239)	-0.422^{*} (0.250)
10+ hits	-0.629^{***} (0.195)	-0.522^{**} (0.252)	-0.614^{***} (0.232)	-0.569^{**} (0.263)	-0.523^{*} (0.267)	-0.486^{*} (0.278)
crossover	0.191 (0.153)	0.139 (0.161)	0.149 (0.162)	0.194 (0.165)	0.183 (0.169)	0.153 (0.174)
reissue	6.729^{***} (1.349)	6.890^{***} (1.346)	6.777^{***} (1.367)	7.229^{***} (1.443)	7.066^{***} (1.452)	7.631^{***} (1.542)
genres						
rap	1.136^{**} (0.565)	1.282^{**} (0.597)	1.521^{**} (0.620)	1.795^{***} (0.643)	1.958^{***} (0.656)	1.928^{***} (0.669)
rock	0.540 (0.447)	1.398^{***} (0.508)	1.129^{**} (0.550)	0.830 (0.584)	0.837 (0.612)	1.392^{**} (0.625)
metal	0.171 (0.520)	1.278^{**} (0.594)	1.072^{*} (0.631)	0.876 (0.661)	0.790 (0.687)	1.291^{*} (0.713)
folk	-0.544 (0.557)	-0.267 (0.615)	-0.522 (0.653)	-0.860 (0.687)	-0.839 (0.712)	-0.167 (0.719)
country	-0.358 (0.500)	0.133 (0.563)	-0.027 (0.604)	-0.280 (0.633)	-0.434 (0.657)	0.029 (0.674)
blues	-1.324^{**} (0.675)	-0.601 (0.731)	-1.034 (0.777)	-1.297 (0.800)	-1.198 (0.804)	-0.802 (0.825)
R&B	1.839^{***} (0.487)	2.269^{***} (0.530)	1.956^{***} (0.563)	1.602^{***} (0.591)	1.589^{**} (0.619)	2.123^{***} (0.639)

soul	0.098	0.708	0.332	-0.193	0.049	0.497
	(0.660)	(0.724)	(0.761)	(0.794)	(0.818)	(0.853)
disco	0.701	1.177^{**}	0.777	0.433	0.466	0.896
	(0.470)	(0.522)	(0.563)	(0.596)	(0.621)	(0.632)
funk	0.075	0.035	-0.440	-0.666	-0.263	-0.109
	(0.849)	(0.883)	(0.942)	(0.978)	(1.030)	(1.036)
pop	1.088^{**}	1.773^{***}	1.459^{***}	1.084^{*}	1.104^{*}	1.611^{***}
	(0.444)	(0.502)	(0.545)	(0.579)	(0.609)	(0.620)
none	-0.069	0.479	0.185	-0.242	-0.165	0.340
	(0.476)	(0.530)	(0.573)	(0.608)	(0.634)	(0.646)
time blocks						
1963-1967	-3.148^{***}	-3.127^{***}	-3.171^{***}	-3.077^{***}	-3.162^{***}	-3.335^{***}
	(0.553)	(0.565)	(0.567)	(0.570)	(0.577)	(0.594)
1968-1972	-0.698^{**}	-0.424	-0.441	-0.272	-0.385	-0.286
	(0.298)	(0.328)	(0.334)	(0.342)	(0.348)	(0.361)
1973-1977	0.982^{***}	1.151^{***}	1.156^{***}	1.260^{***}	1.140^{***}	1.294^{***}
	(0.264)	(0.305)	(0.312)	(0.318)	(0.322)	(0.337)
1978-1982	1.162^{***}	1.243^{***}	1.224^{***}	1.300^{***}	1.291^{***}	1.214^{***}
	(0.280)	(0.333)	(0.342)	(0.347)	(0.351)	(0.367)
1983-1987	1.312^{***}	1.470^{***}	1.498^{***}	1.655^{***}	1.560^{***}	1.606^{***}
	(0.270)	(0.333)	(0.344)	(0.352)	(0.357)	(0.372)
1988-1992	1.078^{***}	1.140^{***}	1.101^{***}	1.217^{***}	1.228^{***}	1.213^{***}
	(0.278)	(0.343)	(0.354)	(0.359)	(0.366)	(0.381)
1993-1997	0.441	0.423	0.540	0.625	0.653	0.565
	(0.324)	(0.389)	(0.405)	(0.413)	(0.423)	(0.437)
1998-2002	0.981^{***}	0.810^{**}	0.878^{**}	0.983^{**}	1.047^{**}	1.159^{**}
	(0.334)	(0.401)	(0.412)	(0.421)	(0.429)	(0.454)
2003-2007	-0.464	-0.613	-0.520	-0.386	-0.427	-0.498
	(0.300)	(0.379)	(0.395)	(0.401)	(0.411)	(0.423)
2008-2012	-1.005^{***}	-1.170^{***}	-1.097^{***}	-0.944^{**}	-1.035^{**}	-1.197^{***}
	(0.318)	(0.394)	(0.406)	(0.414)	(0.422)	(0.436)
The Echo Nest similarity						
genre similarity		-1.530^{*}	-0.956	-0.751	-0.500	-1.259
		(0.925)	(1.013)	(1.022)	(1.039)	(1.047)
artist similarity		-0.278		-0.350	-0.186	-0.466
		(0.214)		(0.307)	(0.301)	(0.306)
The Echo Nest features						
tempo		0.089	0.028	-0.039	-0.018	0.225
		(0.527)	(0.533)	(0.541)	(0.548)	(0.569)
energy		-0.498	-0.588	-0.546	-0.534	-0.892^{*}
		(0.462)	(0.473)	(0.483)	(0.488)	(0.506)
speechiness		1.970	2.566^{*}	2.697^{*}	2.722^{*}	2.958^{*}
		(1.330)	(1.400)	(1.427)	(1.466)	(1.514)
acousticness		0.760^{**}	0.832^{**}	0.986^{***}	0.921^{***}	0.912^{**}
		(0.340)	(0.345)	(0.353)	(0.357)	(0.368)
danceability		3.593^{***}	3.646^{***}	3.748^{***}	3.772^{***}	4.050^{***}
		(0.588)	(0.606)	(0.616)	(0.630)	(0.656)
valence		-1.448^{***}	-1.531^{***}	-1.624^{***}	-1.538^{***}	-1.582^{***}
		(0.380)	(0.388)	(0.393)	(0.402)	(0.419)
instrumentalness		-2.385^{***}	-2.299^{***}	-2.438^{***}	-2.543^{***}	-2.894^{***}
		(0.697)	(0.703)	(0.731)	(0.746)	(0.788)
liveness		-1.104^{***}	-1.186^{***}	-1.245^{***}	-1.149^{***}	-1.183^{***}
		(0.403)	(0.412)	(0.416)	(0.431)	(0.445)

key = C#	0.081	0.055	0.134	0.028	−0.040
	(0.281)	(0.284)	(0.289)	(0.294)	(0.306)
key = D	−0.318	−0.293	−0.296	−0.286	−0.407
	(0.250)	(0.253)	(0.257)	(0.263)	(0.271)
key = E♭	0.537	0.556	0.565	0.596	0.553
	(0.362)	(0.370)	(0.373)	(0.382)	(0.386)
key = E	−0.042	−0.068	−0.060	−0.044	−0.247
	(0.285)	(0.289)	(0.293)	(0.299)	(0.312)
key = F	−0.468*	−0.437	−0.347	−0.413	−0.507*
	(0.265)	(0.268)	(0.273)	(0.276)	(0.294)
key = F#	0.163	0.138	0.141	0.131	0.101
	(0.337)	(0.341)	(0.345)	(0.353)	(0.358)
key = G	−0.197	−0.194	−0.177	−0.169	−0.257
	(0.253)	(0.256)	(0.260)	(0.265)	(0.279)
key = G#	−0.230	−0.194	−0.100	−0.078	−0.190
	(0.335)	(0.340)	(0.341)	(0.348)	(0.356)
key = A	−0.539**	−0.562**	−0.597**	−0.608**	−0.699**
	(0.262)	(0.266)	(0.269)	(0.274)	(0.282)
key = B♭	−0.218	−0.228	−0.192	−0.151	−0.281
	(0.288)	(0.291)	(0.295)	(0.299)	(0.312)
key = B	0.281	0.266	0.248	0.248	0.038
	(0.300)	(0.305)	(0.308)	(0.312)	(0.323)
mode	0.181	0.118	0.102	0.092	0.112
	(0.159)	(0.163)	(0.165)	(0.167)	(0.173)
time signature	0.472	0.423	0.485	0.475	0.502
	(0.305)	(0.313)	(0.317)	(0.320)	(0.329)
topic mixture similarity					
similarity		2.390**	1.662	0.671	−0.954
		(1.218)	(1.505)	(1.844)	(2.323)
genre similarity		−1.180	−2.690***	−3.074**	−1.399
		(0.814)	(1.007)	(1.209)	(1.401)
artist similarity		−0.279	0.095	−0.404	0.494
		(0.271)	(0.456)	(0.510)	(0.637)
topic mixtures					
X1		1.110*	−0.728	1.851	−0.463
		(0.662)	(0.870)	(1.876)	(3.055)
X2		1.780**	−0.030	−2.210	3.539
		(0.763)	(1.090)	(1.919)	(2.755)
X3		−0.172	1.776*	−2.334	2.847
		(0.564)	(1.018)	(1.917)	(3.788)
X4		−0.432	1.141	−5.323**	0.596
		(0.759)	(0.999)	(2.068)	(3.230)
X5		0.106	−1.131	0.075	4.818*
		(0.606)	(0.975)	(2.256)	(2.772)
X6		0.170	1.175	−1.809	−2.781
		(0.798)	(0.890)	(1.369)	(3.277)
X7		−0.110	1.412	−4.526***	−1.877
		(0.616)	(1.281)	(1.697)	(2.143)
X8		0.653	1.256	−0.669	−3.349
		(0.693)	(0.887)	(1.403)	(2.979)
X9		−0.502	−0.578	0.593	2.158
		(0.653)	(1.004)	(1.578)	(2.623)
X10			−0.340	0.434	−0.622

	(0.972)	(1.793)	(2.747)
X11	3.041^{**}	−1.060	5.031^{*}
	(1.208)	(1.628)	(2.934)
X12	0.770	1.839	2.257
	(1.104)	(2.068)	(2.802)
X13	0.858	−1.765	3.683
	(1.086)	(2.020)	(2.859)
X14	-1.969^{*}	0.675	2.041
	(1.088)	(1.976)	(3.336)
X15	1.487	−2.165	0.709
	(1.496)	(1.352)	(2.165)
X16	−0.575	−2.519	−0.518
	(0.952)	(1.782)	(2.729)
X17	0.471	−1.250	0.401
	(1.034)	(1.853)	(2.295)
X18	0.414	−0.824	5.671^{*}
	(0.891)	(1.861)	(2.982)
X19	1.059	0.161	3.280
	(0.925)	(1.482)	(2.792)
X20		1.215	1.631
		(1.550)	(2.569)
X21		−1.273	-8.536^{***}
		(1.775)	(2.584)
X22		−0.002	0.207
		(1.584)	(1.945)
X23		2.512	2.846
		(1.758)	(2.636)
X24		0.781	1.069
		(1.672)	(2.271)
X25		−3.028	10.427^{**}
		(1.980)	(4.328)
X26		−0.116	−0.201
		(1.945)	(2.815)
X27		−1.281	−3.226
		(1.947)	(2.878)
X28		0.530	2.323
		(1.768)	(2.845)
X29		1.387	−2.956
		(2.072)	(3.366)
X30		−3.020	1.325
		(1.989)	(2.418)
X31		−0.716	3.033
		(1.948)	(2.029)
X32		−1.020	2.814
		(1.764)	(2.400)
X33		2.837^{*}	−1.241
		(1.667)	(2.628)
X34		0.643	0.690
		(1.659)	(2.265)
X35		1.230	0.713
		(1.562)	(2.680)
X36		−0.633	−1.724

	(1.462)	(2.249)
X37	−1.509	−1.931
	(1.509)	(2.195)
X38	−0.263	−0.747
	(1.787)	(2.892)
X39	0.653	−1.081
	(1.658)	(3.237)
X40		0.212
		(2.480)
X41		2.260
		(2.609)
X42		−2.370
		(2.631)
X43		−0.021
		(2.641)
X44		−4.848
		(3.066)
X45		0.213
		(2.146)
X46		−2.957
		(2.361)
X47		−1.062
		(2.516)
X48		−0.231
		(2.390)
X49		1.983
		(2.686)
X50		−2.511
		(1.753)
X51		−2.342
		(2.544)
X52		0.566
		(2.854)
X53		−0.805
		(2.819)
X54		−0.142
		(1.996)
X55		-6.982^{***}
		(2.693)
X56		−0.978
		(2.118)
X57		−0.266
		(2.706)
X58		−0.501
		(3.226)
X59		5.275
		(3.424)
X60		−2.447
		(2.881)
X61		−2.826
		(2.088)
X62		−0.598

						(2.169)
X63						3.986
						(2.686)
X64						4.513
						(2.918)
X65						0.930
						(2.323)
X66						−0.472
						(1.983)
X67						6.061**
						(3.020)
X68						−2.959
						(3.161)
X70						−0.431
						(2.968)
X71						−4.173
						(2.817)
X72						2.660
						(2.163)
X73						2.527
						(2.638)
X74						−3.048
						(2.131)
X75						−1.708
						(1.361)
X76						−6.427**
						(2.772)
X77						1.721
						(1.955)
X78						−9.277**
						(4.022)
X79						−0.149
						(3.061)
Observations	1,638	1,638	1,638	1,638	1,638	1,638
Log Likelihood	-873.199	-817.253	-806.656	-791.428	-781.379	-752.163
Akaike Inf. Crit.	1,804.399	1,738.506	1,739.311	1,730.857	1,750.758	1,770.326
Note:						*p<0.1; **p<0.05; ***p<0.01

Table C.2: Alignment target variable logistic regression models for various feature sets.

	Dependent variable:					
	Alignment target variable					
	(metadata)	(The Echo Nest)	(10 topic)	(20 topic)	(40 topic)	(80 topic)
constant	1.206*	0.864	0.392	1.091	1.735	−0.533
	(0.715)	(1.347)	(1.502)	(1.599)	(1.982)	(1.887)
long song	0.123	0.219	0.273	0.243	0.455	0.075
	(0.391)	(0.435)	(0.450)	(0.463)	(0.460)	(0.494)
2-3 hits	−0.325	−0.117	−0.254	−0.255	−0.355	−0.324
	(0.204)	(0.262)	(0.244)	(0.274)	(0.289)	(0.295)
6-10 hits	0.165	0.400	0.289	0.284	0.187	0.287
	(0.187)	(0.280)	(0.242)	(0.292)	(0.304)	(0.318)
10+ hits	0.477**	0.723**	0.674**	0.646*	0.532	0.573

	(0.236)	(0.315)	(0.283)	(0.331)	(0.344)	(0.357)
crossover	−0.039	−0.090	−0.094	−0.063	−0.059	−0.104
	(0.174)	(0.181)	(0.183)	(0.188)	(0.191)	(0.199)
reissue	0.039	0.165	0.339	0.118	0.092	0.308
	(1.102)	(1.164)	(1.182)	(1.226)	(1.191)	(1.111)
genres						
rap	1.721**	2.128**	2.014**	2.119**	2.115**	2.312***
	(0.781)	(0.835)	(0.850)	(0.856)	(0.864)	(0.897)
rock	0.527	1.454*	0.693	0.335	0.028	0.331
	(0.671)	(0.780)	(0.825)	(0.866)	(0.907)	(0.953)
metal	−0.132	1.013	0.414	−0.115	−0.319	−0.239
	(0.733)	(0.844)	(0.882)	(0.931)	(0.963)	(1.021)
folk	0.587	1.462*	0.820	0.407	0.120	0.396
	(0.715)	(0.819)	(0.858)	(0.900)	(0.937)	(0.993)
country	−1.481**	−0.938	−1.677*	−1.931**	−2.385**	−2.286**
	(0.712)	(0.812)	(0.863)	(0.893)	(0.932)	(0.980)
blues	−1.476*	−0.543	−1.329	−1.668	−2.123*	−1.400
	(0.887)	(0.979)	(1.029)	(1.070)	(1.129)	(1.162)
R&B	2.042***	2.518***	1.628**	1.245	0.927	1.180
	(0.685)	(0.768)	(0.811)	(0.842)	(0.887)	(0.944)
soul	−0.489	0.319	−0.655	−1.001	−1.237	−1.306
	(0.752)	(0.847)	(0.899)	(0.936)	(0.970)	(1.032)
disco	0.390	1.128	0.281	−0.057	−0.468	−0.093
	(0.694)	(0.789)	(0.835)	(0.872)	(0.913)	(0.957)
pop	1.585**	2.434***	1.613**	1.255	1.049	1.247
	(0.665)	(0.770)	(0.819)	(0.860)	(0.901)	(0.951)
none	−0.138	0.583	−0.183	−0.587	−1.141	−0.728
	(0.702)	(0.797)	(0.851)	(0.892)	(0.943)	(0.988)
time blocks						
1963-1967	−0.241	−0.172	−0.202	−0.087	−0.018	0.048
	(0.321)	(0.334)	(0.335)	(0.342)	(0.354)	(0.369)
1968-1972	−0.433	−0.497	−0.479	−0.429	−0.389	−0.465
	(0.320)	(0.338)	(0.341)	(0.344)	(0.355)	(0.374)
1973-1977	−2.153***	−2.279***	−2.308***	−2.277***	−2.356***	−2.338***
	(0.361)	(0.386)	(0.392)	(0.397)	(0.405)	(0.431)
1978-1982	−4.235***	−4.489***	−4.624***	−4.530***	−4.474***	−4.719***
	(0.474)	(0.504)	(0.515)	(0.517)	(0.530)	(0.556)
1983-1987	−4.773***	−4.984***	−5.074***	−4.951***	−5.011***	−5.203***
	(0.550)	(0.584)	(0.595)	(0.592)	(0.613)	(0.646)
1988-1992	−3.095***	−3.415***	−3.467***	−3.422***	−3.401***	−3.551***
	(0.396)	(0.441)	(0.453)	(0.458)	(0.468)	(0.493)
1993-1997	−1.712***	−1.964***	−1.987***	−1.858***	−1.699***	−1.862***
	(0.366)	(0.416)	(0.434)	(0.444)	(0.453)	(0.472)
1998-2002	−1.451***	−1.623***	−1.690***	−1.607***	−1.462***	−1.482***
	(0.373)	(0.434)	(0.448)	(0.453)	(0.462)	(0.483)
2003-2007	−2.535***	−2.806***	−2.812***	−2.789***	−2.617***	−2.661***
	(0.376)	(0.448)	(0.466)	(0.470)	(0.485)	(0.500)
2008-2012	−2.996***	−3.121***	−3.221***	−3.069***	−2.854***	−3.154***
	(0.412)	(0.478)	(0.494)	(0.497)	(0.502)	(0.532)
The Echo Nest similarity						
genre similarity		−2.223*	−1.432	−1.229	−1.544	−1.628
		(1.152)	(1.202)	(1.215)	(1.280)	(1.327)
artist similarity		−0.332		−0.062	0.155	−0.365

	(0.276)		(0.386)	(0.385)	(0.393)
The Echo Nest features					
tempo	0.460	0.589	0.656	0.414	0.467
	(0.577)	(0.587)	(0.597)	(0.605)	(0.644)
energy	−1.576***	−1.728***	−1.763***	−1.885***	−2.034***
	(0.568)	(0.578)	(0.588)	(0.605)	(0.624)
speechiness	2.503	2.908*	2.601	3.094*	4.067**
	(1.564)	(1.668)	(1.697)	(1.740)	(1.800)
acousticness	−0.133	−0.179	−0.099	−0.026	−0.007
	(0.374)	(0.379)	(0.387)	(0.396)	(0.415)
danceability	2.266***	2.209***	2.283***	2.309***	2.105***
	(0.684)	(0.703)	(0.725)	(0.735)	(0.771)
valence	−0.823*	−1.076**	−1.176**	−1.025**	−0.975*
	(0.468)	(0.481)	(0.491)	(0.500)	(0.519)
instrumentalness	−1.332	−1.323	−1.171	−1.498	−1.058
	(0.964)	(1.000)	(1.018)	(1.057)	(1.114)
liveness	0.157	0.196	0.074	0.221	0.274
	(0.481)	(0.489)	(0.498)	(0.507)	(0.531)
key = C#	0.420	0.500	0.553	0.621*	0.721*
	(0.346)	(0.353)	(0.357)	(0.375)	(0.384)
key = D	0.609*	0.662**	0.561*	0.776**	0.708**
	(0.318)	(0.322)	(0.325)	(0.335)	(0.346)
key = E♭	0.943**	1.003**	0.897*	1.069**	1.045**
	(0.449)	(0.459)	(0.458)	(0.477)	(0.485)
key = E	0.294	0.303	0.231	0.370	0.301
	(0.335)	(0.340)	(0.344)	(0.354)	(0.365)
key = F	0.522	0.594*	0.638*	0.831**	0.481
	(0.335)	(0.340)	(0.352)	(0.358)	(0.370)
key = F#	0.066	0.089	0.018	0.317	0.178
	(0.365)	(0.374)	(0.376)	(0.383)	(0.401)
key = G	0.427	0.399	0.407	0.497	0.537
	(0.313)	(0.316)	(0.321)	(0.332)	(0.341)
key = G#	0.651*	0.657*	0.552	0.774**	0.725*
	(0.366)	(0.371)	(0.376)	(0.385)	(0.402)
key = A	0.076	0.061	−0.043	0.145	0.148
	(0.319)	(0.323)	(0.328)	(0.336)	(0.346)
key = B♭	0.216	0.313	0.186	0.259	0.275
	(0.365)	(0.369)	(0.374)	(0.390)	(0.400)
key = B	0.622*	0.666*	0.602*	0.703*	0.587
	(0.350)	(0.356)	(0.362)	(0.376)	(0.385)
mode	0.086	0.086	0.014	0.070	−0.058
	(0.197)	(0.202)	(0.204)	(0.208)	(0.220)
time signature	1.067***	1.066***	1.098***	1.088***	1.125***
	(0.329)	(0.333)	(0.341)	(0.345)	(0.359)
topic mixture similarity					
similarity		2.379*	2.612	0.646	1.172
		(1.403)	(1.703)	(2.191)	(2.728)
genre similarity		−2.096**	−3.039***	−2.886**	−2.872*
		(0.927)	(1.151)	(1.424)	(1.608)
artist similarity		−0.293	−0.477	−0.922	0.328
		(0.337)	(0.543)	(0.606)	(0.719)
topic mixtures					
X1		0.486	−0.208	2.640	−2.471

	(0.728)	(1.041)	(2.311)	(3.872)
X2	0.085	0.909	0.386	−2.448
	(0.949)	(1.326)	(2.462)	(2.745)
X3	0.922	−0.495	3.776	4.229
	(0.694)	(1.263)	(2.606)	(2.732)
X4	0.497	−0.297	1.063	6.651
	(0.921)	(1.120)	(2.524)	(4.054)
X5	2.430***	0.844	2.967	4.169
	(0.787)	(1.308)	(3.225)	(2.904)
X6	1.426	0.562	−0.563	4.437
	(1.038)	(1.205)	(1.699)	(2.961)
X7	0.450	2.788*	1.259	−0.364
	(0.763)	(1.451)	(1.943)	(2.914)
X8	0.123	1.297	−0.169	6.340*
	(0.819)	(1.106)	(1.711)	(3.409)
X9	0.360	0.036	−0.292	1.089
	(0.763)	(1.265)	(2.163)	(3.636)
X10		−0.073	0.426	2.929
		(1.202)	(2.010)	(2.996)
X11		−0.050	−1.182	2.220
		(1.328)	(2.073)	(4.025)
X12		−1.455	0.141	2.053
		(1.231)	(2.365)	(3.916)
X13		2.824**	1.170	3.399
		(1.369)	(2.228)	(3.660)
X14		0.433	−0.392	4.163
		(1.222)	(2.138)	(4.184)
X15		1.739	−0.349	6.387**
		(1.789)	(1.814)	(2.608)
X16		−0.233	−0.737	0.134
		(1.083)	(2.272)	(3.447)
X17		1.055	1.536	9.471***
		(1.240)	(2.254)	(2.756)
X18		−0.326	0.519	2.329
		(1.077)	(2.462)	(2.628)
X19		2.252**	−0.827	7.749**
		(1.137)	(1.917)	(3.813)
X20			0.250	6.978**
			(1.815)	(2.810)
X21			−0.855	4.638
			(2.310)	(2.827)
X22			1.829	5.971***
			(1.882)	(2.197)
X23			1.434	7.393**
			(2.042)	(3.303)
X24			4.596**	7.314**
			(1.943)	(2.869)
X25			−2.172	5.351
			(2.170)	(4.673)
X26			−0.951	1.819
			(2.360)	(3.425)
X27			−0.239	1.950

	(2.071)	(2.692)
X28	−0.355	5.393
	(2.358)	(3.760)
X29	−0.665	4.517
	(2.431)	(4.224)
X30	−0.543	−4.784*
	(2.659)	(2.837)
X31	6.006**	1.105
	(2.935)	(2.571)
X32	0.281	4.188
	(2.060)	(2.845)
X33	4.696**	−2.349
	(2.039)	(3.255)
X34	0.113	0.209
	(2.021)	(2.597)
X35	−0.170	−0.928
	(1.747)	(3.238)
X36	−0.769	0.931
	(1.875)	(3.086)
X37	0.152	3.687
	(1.851)	(2.546)
X38	−1.417	4.910
	(2.204)	(4.248)
X39	4.957**	−1.488
	(2.133)	(3.223)
X40		4.309
		(3.582)
X41		3.177
		(3.566)
X42		−3.802
		(3.052)
X43		9.283***
		(3.537)
X44		4.156
		(3.262)
X45		1.317
		(2.837)
X46		−0.549
		(3.140)
X47		−1.137
		(2.921)
X48		5.443*
		(3.017)
X49		6.082*
		(3.494)
X50		2.273
		(1.812)
X51		1.288
		(2.945)
X52		4.974
		(3.373)
X53		3.277

						(2.927)
X54						−1.113
						(2.645)
X55						3.430
						(2.633)
X56						1.795
						(2.329)
X57						9.659***
						(3.399)
X58						3.739
						(4.605)
X59						6.120*
						(3.408)
X60						5.764*
						(3.459)
X61						3.587
						(2.272)
X62						0.312
						(2.327)
X63						3.768
						(3.278)
X64						5.471*
						(3.167)
X65						3.759
						(2.848)
X66						2.752
						(2.395)
X67						−0.748
						(3.310)
X68						2.760
						(3.876)
X70						1.374
						(3.294)
X71						1.008
						(3.996)
X72						3.262
						(2.409)
X73						0.071
						(3.116)
X74						−0.237
						(2.702)
X75						2.916*
						(1.694)
X76						1.903
						(3.858)
X77						4.164*
						(2.168)
X78						1.727
						(2.544)
X79						3.089
						(4.190)
Observations	1,251	1,251	1,251	1,251	1,251	1,251

Log Likelihood	-610.513	-584.974	-574.620	-564.524	-554.912	-537.279
Akaike Inf. Crit.	1,277.025	1,271.947	1,273.241	1,275.047	1,295.824	1,338.558
Note:						*p<0.1; **p<0.05; ***p<0.01

Table C.3: Lasso logistic regression models for various time tasks.

	Dependent variable:					
	Interval target variable					
	(A)	(A control)	(B)	(B control)	(C)	(C control)
constant	−0.671	−0.428	−0.785	−1.516	1.41	−0.313
long song	0	.	0	.	0	.
2-3 hits	0	.	0	.	0	.
6-10 hits	0	.	−0.118	.	0	.
10+ hits	0	.	−0.557	.	0	.
crossover	0	.	0	.	−0.176	.
reissue	0.952	.	1.6	.	1.227	.
genres						
rap	0	.	0	.	0	.
rock	−0.081	.	−0.162	.	0	.
metal	0.204	.	0.343	.	0	.
folk	0	.	0	.	−0.614	.
country	0.489	.	0.725	.	0	.
blues	0	.	0	.	0	.
R&B	0.139	.	0.433	.	0.632	.
soul	0	.	0	.	−1.006	.
disco	0	.	0	.	0	.
funk	0	.	−0.682	.	0	.
pop	−0.338	.	0	.	0	.
none	0	.	0	.	0	.
time blocks						
1963-1967	−0.7	.	−1.728	.	−3.203	.
1968-1972	−0.28	.	−0.976	.	−2.609	.
1973-1977	0	.	−0.566	.	−0.137	.
1978-1982	0	.	0	.	1.258	.
1983-1987	0	.	0.314	.	1.229	.
1988-1992	0.156	.	0.837	.	0.636	.
1993-1997	1.363	.	3.38	.	0.831	.
1998-2002	1.389	.	2.487	.	0.732	.
2003-2007	0.536	.	2.683	.	1.062	.
2008-2012	0	.	1.549	.	0.484	.
The Echo Nest similarity						
genre similarity	0	0	0	0	0	0
artist similarity	−0.023	−0.18	−0.625	−0.847	0	−0.065
The Echo Nest features						
tempo	0	0	0	0	0	0
energy	0	0	0	1.438	0.204	1.125
speechiness	0	0	0	0	0	0.59
acousticness	−0.765	−1.375	−0.624	−1.469	−1.044	−1.669
danceability	0.123	1.319	0	2.693	0.009	3.522
valence	−0.331	−1.163	−0.327	−2.793	−0.236	−2.254
instrumentalness	0	0	0	0	0	0
liveness	−0.231	−0.392	0	−0.261	0	0
key = C#	0	0	0	0	0	0
key = D	0	0	0	0	0	0
key = E♭	0	0	0	0	0	0
key = E	0	0	0	0	0	0
key = F	0	0	0	0	0	0
key = F#	0.1	0.158	0	0.018	0	0
key = G	0	0	0	0	0	−0.038
key = G#	0	0	0	0.273	−0.655	−0.352
key = A	0	0	0	0	0	0
key = B♭	0	0	0	0	0	0
key = B	0	0	0	0	0.018	0
mode	0	0	0	0	0	0
time signature	0	0	0.232	0	0	0
topic mixture similarity						
similarity	0	0	0	0	0	0
genre similarity	0	0.425	0.705	3.481	0	0
artist similarity	0	0	0	0	0	0
topic mixtures						
X1	0	0	0	0	0	0
X2	0	0	0	0	0	0
X3	0	0	−0.524	0	0	0
X4	0	0	0	−0.577	0	0

X5	0	0	0	0.631	0	0
X6	0	0	0	0	0.338	1.87
X7	0	0	0	0	0	0
X8	0	0	0	0	0	0
X9	0	0	0	0	0.415	1.019
X10	0	0	0	0	0	0
X11	0	0	0	−0.136	0.115	0
X12	0	0	0	0	0	0
X13	0	0	0	0	0	0
X14	0	0	0	0	0	−1.721
X15	0	0	0	0	0	0
X16	0	0	0	0	−1.134	−0.082
X17	0	0	0	0	0	−0.524
X18	0	0	0	0	0	0
X19	0	0	0	0.295	0	0
Observations	1,167	1,167	1,222	1,222	342	342
DF	18	7	23	13	25	13
λ_{SE}	0.028	0.027	0.014	0.021	0.025	0.036
%Dev	0.204	0.087	0.541	0.24	0.63	0.253
Note:						*p<0.1; **p<0.05; ***p<0.01

Table C.4: Lasso logistic regression models for various peak tasks.

	Dependent variable:					
	Interval target variable					
	(A)	(A control)	(B)	(B control)	(C)	(C control)
constant	−1.147	−0.77	−3.052	−1.697	0.135	−0.286
long song	0	.	0	.	0.086	.
2-3 hits	0	.	0	.	0	.
6-10 hits	0.112	.	0	.	0	.
10+ hits	0	.	0	.	0.259	.
crossover	0	.	0.457	.	0.009	.
reissue	0	.	0	.	0	.
genres						
rap	0	.	0	.	0	.
rock	0	.	0.033	.	0.253	.
metal	−0.35	.	0	.	0	.
folk	−0.441	.	0.326	.	0.369	.
country	0	.	−0.169	.	−0.674	.
blues	0	.	0	.	0	.
R&B	0	.	−0.029	.	0	.
soul	0	.	0.922	.	0	.
disco	0.099	.	0	.	0	.
funk	0	.	0	.	0	.
pop	0	.	0.256	.	0.307	.
none	−0.295	.	0	.	0	.
time blocks						
1963-1967	0	.	3.198	.	1.799	.
1968-1972	0	.	2.366	.	0.445	.
1973-1977	0.058	.	1.918	.	−1.276	.
1978-1982	0.282	.	0	.	−2.224	.
1983-1987	0.536	.	−0.261	.	−3.015	.
1988-1992	0	.	0	.	−2.94	.
1993-1997	−0.107	.	−0.606	.	−2.781	.
1998-2002	0	.	−0.509	.	−2.664	.
2003-2007	−0.053	.	−0.755	.	−2.09	.
2008-2012	0	.	−0.321	.	−1.423	.
The Echo Nest similarity						
genre similarity	0	0	0	0	0	0
artist similarity	0	0	0	0	0	0.097
The Echo Nest features						
tempo	0	0	0	0	0	0
energy	0	0	0	−1.856	−1.075	−1.307
speechiness	0	0	0	0	0	0
acousticness	0.292	0	2.584	3.015	0.386	2.187
danceability	1.42	1.016	0	−0.672	0	−2.034
valence	0	0	0.727	2.658	1.078	2.9
instrumentalness	0	0	0	0	0	0
liveness	0	0	0	0	0	0
key = C#	0	0	0	0	0	0
key = D	0.115	0	0	0	0	0
key = E♭	0	0	0	0	0	0
key = E	0	0	0	0	0	0

key = F	0	0	0	0	0	0
key = F#	0	0	0	0	−0.152	−0.627
key = G	0	0	0	0	0	0
key = G#	0	0	0	0	0	0
key = A	0	0	0	0	0	0
key = B♭	0	0	0	0	0	0
key = B	0	0	0	0	0	0
mode	0	0	0	0	0	0
time signature	0	0	0	0	0	0
topic mixture similarity						
similarity	0	0	0	0	0	0
genre similarity	0	0	0	−0.565	0	−1.386
artist similarity	0	0	0	0	0.137	0.307
topic mixtures						
X1	0	0	0	0	0	0
X2	0	0	0	0	0	−0.39
X3	0	0	0	0	0	0
X4	0	0	0	0	0	0
X5	0	0	0	−0.154	0	0
X6	0	0	0	−0.483	0	0
X7	0	0	0	0	0	0
X8	0	0	0	0	0	0
X9	0	0	−0.448	−2.598	0	0
X10	0	0	0	0	0	0
X11	0	0	0	0	0.158	0
X12	0	0	0	0	0	0
X13	0	0	0.952	0.79	0	0
X14	−0.839	0	0	1.411	0	0
X15	0	0	0	0	0	0
X16	0	0	0.314	0	0	0
X17	0	0	0	1.761	0	0
X18	0	0	0	0	0	0
X19	0	0	0	0	0	0
Observations	454	454	358	358	438	438
DF	14	1	20	11	23	9
λ_{SE}	0.043	0.063	0.024	0.031	0.019	0.038
%Dev	0.088	0.018	0.615	0.352	0.67	0.266
Note:				*p<0.1; **p<0.05; ***p<0.01		

Table C.5: Multinomial lasso logistic regression model for modelling chart trajectories using full feature set.

	Dependent variable:				
	Alignment target variable				
	(A)	(B)	(C)	(D)	(E)
constant	0.16	0.275	0.536	−0.178	−0.792
long song	0	0	0	0	0
2-3 hits	0	0	0	−0.032	0
6-10 hits	0	0	0	0	0
10+ hits	0	0	0	0	0
crossover	0	0	0	0.055	0
reissue	0	0	0	0	0
genres					
rap	0	0	0	0	0
rock	0	0	0	0	0
metal	0	0	0	0	0
folk	0	0	0	0.422	0
country	0.159	0	0.23	−0.004	0
blues	0	0	0	0	0
R&B	0	0	0	0.713	0
soul	0	0	0	0	0
disco	0	0.145	0	0	0
funk	0	0	0	0	0
pop	0	0	−0.022	0.3	0
none	0	0	0	0	0
time blocks					
1963-1967	0	0.033	−1.456	0.11	−0.543
1968-1972	0	0.242	−1.018	0.653	−0.232
1973-1977	0	0.117	−0.717	0	0.747
1978-1982	0	0	0.24	0	3.179
1983-1987	0	0	1.231	0	2.53
1988-1992	0	−0.477	0.081	−0.445	1.115

1993-1997	0	0	0.219	0	0
1998-2002	0	−0.215	0	0.278	0
2003-2007	0.267	0	0	0	0
2008-2012	0.672	0	0	0	0
The Echo Nest similarity					
genre similarity	0	0	0	0	0
The Echo Nest features					
tempo	0	0	0	0	0
energy	0	0	0.169	0	0
speechiness	0	0	0	0	0
acousticness	0	0	−0.93	0.361	0
danceability	0	−0.166	0	0	0
valence	0	0	0	0	0
instrumentalness	0	0	0	0	0
liveness	0	0	0	0	0
key = C♯	0	0	0	0	0
key = D	0	0	0	0	0
key = E♭	0	0	0	0	0
key = E	0	0	0	0	0
key = F	0	0	0	0	0
key = F♯	0	0	0	0	0
key = G	0	0	0	0	0
key = G♯	0	0	0	0	0
key = A	0	0	0	0	0
key = B♭	0	0	0	0	0
key = B	0	0	0	0	0
mode	0	0	0	0	0
time signature	0	0	0	0	0
topic mixture similarity					
similarity	0	0	0	0	0
genre similarity	0	0	0	0	0
artist similarity	0	0	0	0	0
topic mixtures					
X1	−0.216	0	0	0	0
X2	0	0	0	0	0
X3	0	0	0	0	0
X4	0	0	0	0	0
X5	0	0	0	0	0
X6	0	0	0	0	0
X7	0	0	0	0	0
X8	0	0.003	0	0	0
X9	0	0	0	0	0
Observations	1,280				
DF	22				
λ_{SE}	0.025				
%Dev	0.209				
Note:					*p<0.1; **p<0.05; ***p<0.01

Table C.6: Multinomial lasso logistic regression model for modelling chart trajectories using feature set with metadata features removed.

	Dependent variable:				
	Alignment target variable				
	(A)	(B)	(C)	(D)	(E)
constant	0.513	0.1	−0.184	−0.186	−0.243
The Echo Nest similarity					
genre similarity	0	0	0	0	0
The Echo Nest features					
tempo	0	0	0	0	0
energy	0	0	0.825	0	0
speechiness	0	0	0	0	−2.265
acousticness	0	0.419	−0.843	0.885	0
danceability	0	−0.243	0	0	0.821
valence	0	0	0	0	0
instrumentalness	0	0	0	0	0
liveness	0	0	0	0	0
key = C♯	0	0	0	0	0
key = D	0	0	0	0	0
key = E♭	0	0	0	0	0
key = E	0	0	0	0	0
key = F	0	0	0	0	0
key = F♯	0	0	0	0	0

key = G	0	0	0	0	0
key = G#	0	0	0	0	0
key = A	0	0	0	0	0
key = B♭	0	0	0	0	0
key = B	0	0	0	0	0
mode	0	0	0	0	0
time signature	−0.433	0	0	0	0
topic mixture similarity					
similarity	0	0	0	0	0
genre similarity	0	0	0	0	0
artist similarity	0	0	0	0	0
topic mixtures					
X1	0	0	0	0	0
X2	0	0	0	0	0
X3	0	0	0	0	0
X4	0.023	0	0	0	0
X5	0	0	0	0	0
X6	0	0	0	0	0
X7	0	0	0	0	0
X8	0	0.225	0	0	0
X9	0	0	0	0	0
Observations	1,280				
DF	7				
λ_{SE}	0.03				
%Dev	0.035				
Note:					*p<0.1; **p<0.05; ***p<0.01

Table C.7: Ordered logit regression models fit to complete dataset using Discogs genre in models #1 and #2 and Spotify genre in models #3 and #4.

	Dependent variable:			
	Peak position (inverted)			
	(1)	(2)	(3)	(4)
I(typˆ2)		−6.413*		−7.071*
		(3.482)		(3.917)
similarity	−2.020***	8.079	−1.175*	9.999
	(0.681)	(5.486)	(0.673)	(6.205)
long song	−0.308***	−0.306***	−0.302***	−0.294***
	(0.108)	(0.108)	(0.107)	(0.107)
2-3 hits	−0.319***	−0.321***	−0.364***	−0.364***
	(0.058)	(0.058)	(0.059)	(0.059)
6-10 hits	−0.112**	−0.112**	−0.178***	−0.178***
	(0.054)	(0.054)	(0.055)	(0.055)
10+ hits	−0.091	−0.092	−0.153**	−0.150**
	(0.063)	(0.063)	(0.064)	(0.064)
crossover	0.215**	0.215**	0.062	0.059
	(0.104)	(0.104)	(0.052)	(0.052)
reissue	0.553***	0.556***	0.490**	0.476**
	(0.212)	(0.212)	(0.212)	(0.212)
The Echo Nest features				
tempo	0.183	0.201	0.181	0.200
	(0.164)	(0.164)	(0.164)	(0.164)
energy	−0.013	0.010	−0.093	−0.068
	(0.148)	(0.149)	(0.148)	(0.149)
speechiness	−1.383***	−1.399***	−0.393	−0.407
	(0.418)	(0.418)	(0.441)	(0.441)
acousticness	0.251**	0.274***	0.218**	0.245**
	(0.105)	(0.105)	(0.105)	(0.105)
mode	0.202**	0.230**	0.119	0.140
	(0.092)	(0.094)	(0.091)	(0.092)

danceability	1.202^{***}	1.219^{***}	1.306^{***}	1.329^{***}
	(0.196)	(0.196)	(0.195)	(0.196)
valence	-0.245^{*}	-0.225^{*}	-0.289^{**}	-0.270^{**}
	(0.129)	(0.130)	(0.129)	(0.130)
instrumentalness	-1.069^{***}	-1.074^{***}	-0.878^{***}	-0.876^{***}
	(0.204)	(0.204)	(0.202)	(0.202)
liveness	-0.318^{**}	-0.322^{**}	-0.284^{**}	-0.285^{**}
	(0.131)	(0.131)	(0.130)	(0.130)
key = C#	0.158^{*}	0.156^{*}	0.193^{**}	0.190^{**}
	(0.092)	(0.092)	(0.092)	(0.092)
key = D	−0.040	−0.041	−0.006	−0.010
	(0.081)	(0.081)	(0.081)	(0.081)
key = E♭	0.315^{***}	0.316^{***}	0.329^{***}	0.330^{***}
	(0.119)	(0.119)	(0.119)	(0.119)
key = E	−0.032	−0.031	0.004	0.006
	(0.089)	(0.089)	(0.089)	(0.089)
key = F	0.090	0.092	0.099	0.097
	(0.087)	(0.087)	(0.087)	(0.087)
key = F#	0.025	0.023	0.042	0.044
	(0.101)	(0.101)	(0.101)	(0.101)
key = G	−0.010	−0.011	0.008	0.006
	(0.079)	(0.079)	(0.079)	(0.079)
key = G#	0.017	0.012	0.061	0.055
	(0.101)	(0.101)	(0.101)	(0.101)
key = A	−0.063	−0.062	−0.048	−0.051
	(0.082)	(0.082)	(0.082)	(0.082)
key = B♭	-0.166^{*}	-0.167^{*}	−0.146	−0.147
	(0.095)	(0.095)	(0.095)	(0.095)
key = B	−0.036	−0.038	0.011	0.010
	(0.094)	(0.094)	(0.094)	(0.094)
time signature	0.381^{***}	0.380^{***}	0.378^{***}	0.379^{***}
	(0.084)	(0.084)	(0.084)	(0.084)
time blocks				
1963-1967	−0.130	−0.134	−0.122	−0.123
	(0.099)	(0.099)	(0.100)	(0.100)
1968-1972	-0.427^{***}	-0.444^{***}	-0.478^{***}	-0.495^{***}
	(0.104)	(0.105)	(0.106)	(0.106)
1973-1977	-0.190^{*}	-0.222^{**}	-0.258^{**}	-0.294^{***}
	(0.110)	(0.112)	(0.112)	(0.114)
1978-1982	-0.313^{***}	-0.346^{***}	-0.349^{***}	-0.383^{***}
	(0.113)	(0.115)	(0.115)	(0.117)
1983-1987	-0.386^{***}	-0.422^{***}	-0.387^{***}	-0.421^{***}
	(0.118)	(0.120)	(0.118)	(0.120)
1988-1992	-0.371^{***}	-0.403^{***}	-0.360^{***}	-0.394^{***}
	(0.122)	(0.124)	(0.121)	(0.123)
1993-1997	-1.086^{***}	-1.134^{***}	-1.050^{***}	-1.100^{***}
	(0.133)	(0.136)	(0.133)	(0.136)
1998-2002	-0.734^{***}	-0.780^{***}	-0.751^{***}	-0.797^{***}
	(0.135)	(0.138)	(0.135)	(0.138)
2003-2007	-0.829^{***}	-0.868^{***}	-0.857^{***}	-0.898^{***}
	(0.131)	(0.133)	(0.133)	(0.135)
2008-2012	-1.120^{***}	-1.149^{***}	-1.240^{***}	-1.272^{***}

	(0.132)	(0.134)	(0.134)	(0.135)
Discogs genres				
blues	-0.427^{***}	-0.434^{***}		
	(0.165)	(0.166)		
child	2.183^{*}	2.147^{*}		
	(1.266)	(1.254)		
classical	-1.022	-0.989		
	(0.896)	(0.895)		
electronic	0.087	0.091		
	(0.097)	(0.097)		
folk, world, and country	-0.641^{***}	-0.648^{***}		
	(0.103)	(0.103)		
funk and soul	-0.172^{*}	-0.175^{*}		
	(0.092)	(0.092)		
hip hop	0.391^{***}	0.393^{***}		
	(0.113)	(0.113)		
jazz	-0.460^{***}	-0.452^{***}		
	(0.153)	(0.153)		
latin	-0.425^{*}	-0.429^{*}		
	(0.219)	(0.219)		
non-music	0.267	0.309		
	(0.506)	(0.509)		
pop	0.206^{**}	0.207^{**}		
	(0.092)	(0.092)		
reggae	-0.047	-0.041		
	(0.254)	(0.254)		
rock	0.032	0.031		
	(0.091)	(0.091)		
stage and screen	0.346^{*}	0.346^{*}		
	(0.190)	(0.190)		
Spotify genres				
blues			-0.093	-0.113
			(0.248)	(0.248)
country			0.071	0.053
			(0.190)	(0.190)
disco			0.520^{***}	0.506^{***}
			(0.184)	(0.184)
folk			0.525^{***}	0.513^{***}
			(0.198)	(0.198)
funk			0.231	0.229
			(0.326)	(0.326)
metal			0.451^{**}	0.440^{**}
			(0.210)	(0.210)
none			0.348^{*}	0.330^{*}
			(0.187)	(0.187)
pop			1.167^{***}	1.159^{***}
			(0.181)	(0.181)
rap			0.680^{***}	0.671^{***}
			(0.210)	(0.210)
R&B			1.204^{***}	1.193^{***}
			(0.185)	(0.185)
rock			0.772^{***}	0.760^{***}
			(0.182)	(0.182)

soul			0.275 (0.212)	0.264 (0.212)
Observations	7,683	7,683	7,683	7,683
Note:				*p<0.1; **p<0.05; ***p<0.01