Music Popularity Prediction with Neural Networks:

A Composition-Based Approach

April 14, 2023

Word Count: 5019

Context

For as long as can be determined, humans have been composing music, an art form that serves as a critical vector for the communication of ideas, stories, and emotions. Because of the uniquely human nature of these purposes, it is easy to suppose that the composition and analysis of music are uniquely human processes. In recent years, however, this alluring supposition has been challenged by the development of computer and statistical systems known as machine learning models. The more recent introduction of increasingly complex, multi-layered machine learning models known as neural networks or “deep learning” models has only accelerated this progress, with one researcher noting, “The proposal and application of the concept of deep learning is the biggest breakthrough in the 21st century” (Lin, 2022, p. 1).

These models can complete elaborate tasks that were long believed to be beyond the capabilities of computers. They can generate original photorealistic images, craft poetry, analyze human speech, and even write complicated computer programs and design websites. While many forms of software are deterministically limited to the capabilities hardcoded by their programmers, neural networks sidestep typical limitations through a structure designed to loosely mimic the human brain. Through an iterative training process, many interconnected layers of statistical units known as “neurons” are calibrated to associate given inputs of some kind with given outputs. This structure allows these models to establish complex correlations between inputs (such as captions) and outputs (such as original photorealistic images) that no mere computer program or statistical model could achieve. This unique structure has allowed neural networks to revolutionize various computing fields.

One such field is music. Though there is much uncertainty and philosophical debate surrounding the potential for neural networks to demonstrate true creativity, there is no question that they excel at tasks often considered creative such as generating original compositions (Carnovalini & Rodà, 2020). The continued and expanded application of neural networks to the field of music is valuable for several reasons. Computationally speaking, this pursuit will further our understanding of the creative nature - or lack thereof - of neural networks. More practically, neural networks that are capable of both generating and analyzing various aspects of music are valuable to musicians (Jabade et al., 2019).

Absent any consideration of the real, practical relationship between computational systems and human musicians, research into the cross-section of the two fields would remain ungrounded and meaningless. Fortunately, ample literature exists on this relationship and its nature. Sturm et al. (2019) performed a study of the potential application of neural networks to composition in which they begin with the premise that such work can only be truly meaningful if it is genuinely useful to real-world practitioners. They developed four machine learning tools with unique skill sets. Then, they invited several composers to use these models to write original music for performance. The researchers performed qualitative analysis on in-depth descriptions of the experience from the perspectives of the composers, musicians, and audience members. Generally, the composers felt that their work benefitted from collaboration with a machine learning model. The audience also left the experience with a generally favorable impression, with one member reflecting, “What matters is how good the music is - not the origin” (Sturm et al., 2019, p. 52). Some, however, expressed concerns that the decreased role of human creativity in compositional processes would eliminate some fundamental emotional component of the music. Carnovalini & Rodà (2020) suggest, in their literature review concerning the potential for creativity in computer systems, that a rigorous definition of creativity is required to draw conclusions about the extent to which computers are capable of demonstrating it. They propose that creativity exists when the creator, the methods by which the creation happens, the result of the creation, and the response of culture as a whole align in the proper manner (Carnovalini & Rodà, 2020, p. 3). By such a definition, computer systems are not inherently precluded from consideration for creativity.

Computers work in conjunction as tools of humans in the music industry. Jabade et al. (2019) explored the potential of machine learning for music generation, analysis, and success prediction through the lens of human use. They stated that computational and statistical tools for melody composition and popularity prediction, when paired with creative artists, have immense potential for success in the music industry. Furthermore, machine learning methods can be used to augment the careers of musicians in a variety of other ways. The research of Arakelyan et al. (2018) explored the potential of computer models to predict entire career arcs of music artists, and found that computers can give valuable insights into the outcomes of choices like performing in certain venues. In short, an abundance of evidence suggests that there is a place for expanded computational capabilities in the music field.

Research Questions

The initial purpose of the research that will be discussed in this paper was to explore the general question: What lies at the cross-section of neural network research and music? To justify such research, it is necessary to understand how neural networks can be used practically in the music field. Furthermore, since such a cross-section depends on the application of neural networks to music in some capacity, it is necessary to ask: How are neural networks applied to music? Analytical applications of neural networks to music have been explored to an almost exhaustive degree. Similarly, though generative applications still invite both improvements in quality and philosophical contemplations, they have been fairly well-explored. Predictive approaches, however, are not as clearly understood. Though many studies have leveraged indirect information such as artist information and audio metadata[1] to construct predictive neural networks, few have incorporated raw audio data. No studies to date have examined formats such as MIDI that are more closely tied to the middle phases of the composition process, when an understanding of potential popularity would be a valuable asset to a composer. Artists seeking neural network feedback on their work throughout the compositional process are therefore without access to it. Furthermore, the extent to which a direct correlation between audio features and popularity exists independent of external factors is not clearly established. The vast majority of existing literature focuses on the analysis of finished recordings rather than partial compositions. While this is clearly valuable, it is not particularly helpful to artists throughout their compositional processes. Ultimately, this paper attempts to answer the question: Can a neural network be trained to accurately predict the popularity of a musical composition based on MIDI data?

---

[1] Metadata are pieces of information associated with audio files, such as composer, year of release, and tempo. These can also include analyses, such as the “danceability” metric assigned to each audio file on Spotify.

Professional Conversation

Due to the complex nature of the relationship between a neural network's structure and its purpose, studies in the field vary in methodology based on their objectives. Fortunately, it is possible to broadly classify most of these studies by assigning one of three common purposes to each one: They seek to analyze music, generate music, or make predictions about music.

Analytic Neural Networks

One of the primary reasons for which neural networks are applied to music is to analyze it in some way. For example, the work of Lin (2022) demonstrated that neural networks can be used to identify and label the instrumental components of a given audio sample. The paper incorporated a dense overview of the uncommonly complex deep learning model on which the researcher relied and commented that a more elaborate network has been found to correspond to more accurate identification of musical instruments.

In addition to identifying elements of a piece of music, neural networks can be used to classify it relative to other pieces of music. Zhang et al. (2016), for instance, found that convolutional neural networks are incredibly good at sorting music by genre because of their emphasis on spatial or time-based relationships between nearby elements. The researchers proposed a two-part system that leverages machine learning to reliably and accurately complete a task on a scale beyond human capacity. It was their conclusion that neural networks constitute a superior approach over existing methods. The research of Novello et al. (2013) showed that neural networks are also capable of internalizing human opinions to complete the similar task of inter-song similarity determination. The researchers trained a model to assess the degree to which two pieces are related. Such capabilities of neural networks were put to good use in the research of Huang & Song (2022), in which the authors lamented the lack of a centralized, comprehensive, and accessible internet repository of traditional Chinese folk music. They developed a neural network and paired it with an internet crawling program to search for and identify such music.

Perhaps most impressive is the work of Panwar et al. (2018), in which the researchers leveraged analytic neural networks to indirectly assess the collective emotional states of entire cities. They first identified a scheme by which the emotion in various works could be quantified and then trained a neural network to apply this scheme to musical works. Finally, they accessed radio broadcast data from several major cities and assessed the average emotion conveyed in the music listened to by different regions.

Generative Neural Networks

Generative neural networks often attract more attention than analytic neural networks because of their impressive, realistic compositions. The research of Sturm et al. (2019) encapsulates several good examples of machine learning models that are capable of generating very specific individual components of musical compositions when given very specific human-defined parameters. One of their models could only generate Irish and United Kingdom folk melodies, another could write vocal harmonizations, but couldn't generate melodies of its own, and another was able to transform existing compositions through methods like pitch shifting (Sturm et al., 2019, p. 39). Similarly, Wu et al. (2020) assessed the ability of a neural network to generate melodies for popular music. They developed such a model through the use of both a convolutional neural network and a recurrent neural network[2], into which they incorporated understanding of critical concepts such as repetition, structure, and chord progression. The researchers then compared their model's compositions to real, human-generated melodies and found similarities. Another model with similar characteristics is that proposed by Shi et al. (2021), which was designed to write accompaniment tracks for given melodies. The authors of this paper acknowledged that methods to accomplish this task exist independent of machine learning-based approaches, but they demonstrated through comparative analysis that machine learning yields far more accurate results.

Between the models above, there exists potential for combined productivity in pursuit of an end-to-end machine learning composition process. However, none of them are individually capable of generating complete compositions. It is this distinction that separates them from the model introduced by Dhariwal et al. (2020). Their model, Jukebox, uses deep learning neural network technology to generate high-quality audio music that incorporates voices.

Predictive Neural Networks

The models of greatest interest in this research process are those that make predictions about the future performance of media. Instead of generating or assessing the current state of music, these make predictions about its future condition, usually its popularity. The use of deep learning to predict popularity is not exclusive to the music field. Cai & Zheng (2022) demonstrated that such a model can be employed to accurately predict

[2] RNNs are neural networks capable of feeding layer outputs back into previous layers, used in tasks like predicting future events based on ever-changing data.

the popularity of online news articles before they are ever published. The researchers concluded that such an approach constitutes a significant improvement over existing methods. Similarly, Chen et al. (2018) found that deep learning neural networks are capable of predicting the popularity of social media posts before they become public, providing that the model is supplied with a sufficient amount of contextual metadata concerning the author of the post. This distinction between pure content analysis and metadata analysis also exists in the context of music.

Indirect predictions concerning the direction of the music industry are easily within reach of neural networks. Xu et al. (2022) trained a model using a data mining technique that pulls vast quantities of user metrics from streaming platforms. They found that neural networks are capable of accurately predicting the general popularity of artists over a two-month period. Direct popularity prediction is a slightly more complicated enterprise, however, because audio does not inherently lend itself to analysis in the way that text does. Many researchers therefore choose to make use of publicly available metadata published by companies such as Spotify that describe the basic audio attributes of music in only a few simple numbers. One such research endeavor was that of Gao (2021), in which various statistical models, including a neural network, used these data to make predictions about future popularity. The researcher summarized the difficulty of the task, saying, “There is hardly a linear trend between each feature and popularity since very popular music may have largely diverse audio features” (Gao, 2021, p. 9). Martín-Gutiérrez et al. (2020) proposed a more elaborate deep learning neural network capable of accomplishing similar tasks based on similar data. They, however, incorporated an essential analysis of raw audio data and lyrics, which is not seen in Gao's work. For this reason, the process of Martín-Gutiérrez et al. is uniquely effective.

This research explored a logical extension of these studies, especially that of Martín-Gutiérrez et al., seeking to develop a neural network capable of accurately predicting the popularity of a composition based solely on notes to be of maximum utility to real-world practitioners.

Methods

Due to the computationally intensive nature of neural networks, an immense data set was required for the training process, and a significant amount of time was spent collecting and then manipulating relevant training and test data. Because this research sought to fill a methodological gap, the work done directly with the model utilized a necessarily recursive methodology. Both data preparation and model refinement approaches are explored in the following sections.

Data Collection

This research targeted musical artists at partial stages of their composition processes. Therefore, a general data format that describes pitches and durations, like MIDI, was desirable for training and testing. However, since no large MIDI dataset was readily available, one was created. Trpkovska et al. (2019) outlined an approach in which data mining scripts are devised that are capable of querying the publicly accessible Spotify API[3]. This API allows access to audio samples, song metadata, and useful popularity statistics. Gao (2021) and Martín-Gutiérrez et al. (2020) also employed similar methodology in data collection. To

[3] An API is an Application Programming Interface, a utility that gives software developers access to computational resources.

collect the data, access to the Spotify API was obtained, and software[4] was created that is capable of querying it. The API offers a search function, which was crawled by the software in a pseudo-random manner based on searches consisting of single alphanumeric characters. The software downloaded the thirty-second sample of each song as MP3 waveform audio, as well as a number indicating its popularity from 1 to 100[5]. Initially, approximately 5,000 samples were retrieved, but this was later expanded to over 12,000.

Data Manipulation

The novelty of the approach presented here lies specifically in its potential to be applied to data that might arise at a partial stage of the music composition process. Therefore, the finished audio collected from the Spotify API had to be reduced to MIDI, a less dense, more general format that describes pitches rather than waveform. Bittner et al. (2022) introduced a process by which a machine learning model converts MP3 audio to MIDI, and they created a software library which implements this process. This software, dubbed “Basic Pitch” by its developers, was applied to the collected data, converting it from MP3 to MIDI.

Following this conversion, the data was tokenized for machine learning. Fradet et al. (2021) introduced software called “MidiTok” specifically suited to tokenizing MIDI data for machine learning. Their software was applied to the data set, using the MIDI-like tokenization process[6], the simplest of many offered by the software and commonly used in

[4] Complete code and data are published here: https://github.com/NLenzini/Neural_Network_Music

[5] Spotify assigns these numbers with an internal algorithm. They are a composite of the number of plays and the recency of plays, to some extent.

[6] The MIDI-like method “encodes MIDI messages (Note on, Note off, Time-shift...) as distinct tokens” (Fradet et al., 2021, p. 1). Other potential methods include structured MIDI encoding, REMI, compound word, octuple, and MuMIDI.

the field. The primary purpose of this step was to change the MIDI data from a file format intended for sound applications to a data structure easily interpreted by a neural network. Additionally, MIDI events that commonly occured in sequence were "learned" during the tokenization process and combined for optimization of the data.

The data were then split into a training set, which consisted of eighty percent of the total data and was used to accurately calibrate the network's parameters, and a test set, which consisted of the remaining twenty percent of the total data and was presented to the network as novel material to assess its efficacy.

Model Development

Yang et al. chronicled the effective application of a convolutional neural network to a predictive musical endeavor (2017). Hoping to replicate their success, this researcher also employed a convolutional neural network. Ideally, such a structure would emphasize the analysis of small pieces of each composition being examined in a manner that emulated the piecewise structure of many popular musical compositions. In accordance with the best practices outlined by Paper (2021), the neural network was initialized using the Python 3 programming language and the TensorFlow 2.x library. An initial loss function[7], sparse categorical cross-entropy[8], was selected. The purpose of this function is to quantify the model's error and allow it to improve its performance. Initial layers were selected, each of a predetermined shape and type. Three one-dimensional convolution layers, designed to emphasize the relationships between nearby notes in the song, were used, interspersed with

[7] Neural networks train themselves through a process called "gradient descent". The model's parameters and losses are represented as surfaces and vectors in higher dimensional space, and the most efficient path to the loss of the least magnitude is determined through the calculation of approximate partial derivatives.

[8] Sparse categorical cross-entropy computes loss through a logarithmic operation that can be performed on integers.

one-dimensional max-pooling layers, which condensed the inputted data into fewer individual points. Following the convolutions was a flattening layer, which reshaped the multi-dimensional convolutional data into a one-dimensional array. A dense layer, simply a traditional set of neurons with weights and biases, was then used. A dropout layer, which introduced a degree of randomness to the training process, followed. The final output layer gave a predicted popularity value from 1 to 100.

Model Refinement

Software was developed to evaluate the model's performance. After training, it was presented with each individual element of both the training data set and the testing data set. Predictions were compared against the actual popularity values associated with the tracks, and error was determined. Though the model primarily used sparse categorical cross-entropy to determine its own error throughout the training process, a mean absolute error calculation provided more intuitive results to the user. The evaluation software therefore calculated error simply by finding the average distance between the expected popularity value and the predicted popularity value. Each was represented on a percent scale, from 0 to 99. Furthermore, the model was pitted against a simple opponent that guessed the mean popularity value of the data set for every input. Both the network and the average-guessing opponent were compared against the actual value using mean absolute error. This provided a numerical benchmark for meaningful improvement over general common sense.

Evaluation against each dataset provided insights that were valuable to both the ultimate conclusions of the research and the furthering of the iterative process by which the model was refined. Testing the model's performance on the training data allowed the extent to which it had memorized to be determined, while evaluating its predictions on the testing data revealed the extent to which it had generalized. The ideal model would strike a balance, neither learning its data too specifically nor acquiring impractically general skills. Once a model had been trained, this method of evaluation allowed weak points to be identified. The next iteration of the model then implemented alterations designed to minimize these weak points. Possible alterations were many and varied, and included changes to the type, number, and size of layers, selection of a new loss function, modification of the data set, and alteration of the validation behavior, a process in which a neural network gradually incorporates more of the training data as it goes through the training in order to maintain a steady source of novel material.

This process, when repeated as many times as necessary, should naturally guide the model to an improved state in which conclusions regarding its capacity to aid musicians in their compositional processes can be reached.

Results

In the course of this research, five distinct iterations of the music popularity prediction neural network, including the aforementioned initial model, were trained. That model was trained twice, and proved largely ineffective both times. Each of its failures is attributable to a different suboptimal behavior, and those same behaviors plague all future iterations in some form or another. In that way, the successes and limitations of the initial model serve as a neat summary of the efficacy of the entire endeavor.

The first training of the initial iteration of the model yielded high memorization but low generalization. It showed only 6% error on the training data, but erred by 95% on average when applied to the testing data. The simulated, mean-guessing opponent achieved about 12% error. Anything lower on the testing data indicates some degree of effective generalization, and anything higher indicates failure. However, on its second round of training, the model changed course and guessed only a popularity of 66% for each input. This came closer, but failed to constitute meaningful improvement.

In the second iteration of the model, the sizes and number of layers were decreased, in hopes of fighting overfitting, but very little changed in terms of the model's behavior. In the third iteration, the layers of the initial iteration were restored, but validation behavior was changed. This, too, failed to meaningfully improve the model's behavior. The fourth iteration was trained on a larger data set and used mean absolute error as its loss function. This model made wildly erratic guesses, eventually demonstrating an accuracy level measurable in fractions of a percent. The final fifth iteration restored sparse categorical cross-entropy as the loss function and introduced larger layers. It too, however, primarily guesses 66% popularity. It displays an error of just over 12% on the testing data, while the benchmark for meaningful insight remains just below 12%.

Discussion

Though one can conclude fairly simply from the above results that this approach was ineffective, the reflexive and perhaps even introspective nature of this research means that this conclusion carries relevance for two different and generally distinct communities of professional practice. The conclusions that one can draw regarding the extent to which melodic elements impact the popularity of music and the role which computers play, can play, and may grow to play in composition belong to musicians and composers, those who use neural networks such as those developed in the course of this research as tools. However, the conclusions that one can draw regarding the efficacy of neural networks in isolating melodic features and generalizing to a task are vastly more relevant in the realm of computer science, to those who study neural networks and seek to expand their capabilities. In a reflection of the duality of the conclusions that will be drawn, the implications for each of these areas of practice will be discussed here.

Computer Science

The failure of the first iteration of the model is due to a phenomenon known as overfitting, in which the model simply memorizes its training data in order to most easily complete its task. An overfitted model has never been incentivized to learn certain features of songs that might render them popular. Therefore, when exposed to material that it has not had the opportunity to memorize, it is utterly helpless to draw any meaningful conclusions from the data. The causes of overfitting are many, but two possibilities are excessive training and a mismatch between network size and data set length. Approaches such as shrinking the network and later expanding the data set did resolve the problem of overfitting in this case. However, these later iterations usually guessed only a few values for popularity, typically 66. These models could never defeat their simulated, average-guessing opponent, because 66 wildly overshoots the average for all data sets involved; the typical average popularity of a song in a data set obtained from the Spotify API through the processes outlined above is about 58%. 66% is, however, the mode of the training data used. This suggests, interestingly, that the model iterations that did not overfit prioritized being exactly right sometimes over being nearly right more frequently. It would be logical to conclude from this that a change in the loss function, perhaps from sparse categorical cross-entropy to a simpler function like mean absolute error, would incentivize the training model more correctly, except that such a change resulted in a model devoid of any discernible behavioral pattern. The significance of the loss algorithm should be noted, as well as the poorly understood inefficacy of mean absolute error in training.

These results also indicate an inability of the network to generalize effectively. If the network architecture had the capacity to encode meaningful insights in its weights and biases, it would be able to discriminate, labeling some music as more popular and some music as less popular. Its inclination to default to the mode popularity suggests that there is little in the tokenized MIDI that corresponds to popularity.

The absence of evidence to suggest the efficacy of a convolutional neural network leaves open the possibility that a radically different architecture like a classical neural network or an RNN would be better suited to the task. In fact, a neural network may not be the best tool for this task at all.

Music

The apparent lack of a correlation between melody and popularity emphasized by these models is a significant consideration for music enthusiasts and professionals. Because this is firstly a study of neural networks and only secondarily an examination of musical phenomena, it is difficult to draw definite conclusions from these results regarding the extent to which melody is actually correlated to popularity. Past research deals very little with this issue, and it is not clearly understood at present. However, the extreme nature of the models' failures does suggest a possible absence of meaningful indicators of popularity in the MIDI.

In all likelihood, popularity is governed by lyrics, artist, and other contextual factors beyond composition to an extent that precludes popularity as an exclusive determinant.

Furthermore, the conversion from MP3 to MIDI data constituted a serious reduction in data. Lyrics and instrumentation were not preserved, and other factors thought to be deeply influential in this regard such as the artist's preexisting popularity were also not included. Though vocal melodies were sometimes misinterpreted by the conversion algorithm, bass lines were reproduced fairly faithfully in several random samples examined. These lines, which often encode chord progressions, tend to offer a fairly comprehensive overview of the melodic constraints within which a piece exists. Therefore, if one is to suggest the irrelevance of melody on the basis of these results, one might also question the relevance of chord progression.

In no way do neural networks "run backwards". Therefore, even if a more effective iteration on this initiative is developed and finds a popularity-causing formula, it cannot apply this formula compositionally or communicate it to human musicians. In this way, we are a long way from cracking the code of music popularity with neural networks. The failures of all iterations to provide meaningful insight into compositional data indicate that neural networks' role in the composition process is still one limited primarily to generative applications.

Limitations

Several factors limited the accuracy and scope of this research. Principally among these were small data sets and limited access to computational power. Generally, more training data leads to more effective neural networks. The structure of the Spotify API limited the number of samples that could reasonably be collected. Additionally, the computational limits of the machine[9] used for training and testing prevented rapid development of improved iterations, and limited the number of iterations possible overall.

Furthermore, the model was trained on imperfect data. The process by which MP3 was reduced to MIDI was imperfect and non-deterministic. The machine learning algorithm used made its best guesses as to which notes were being played at every timestamp examined, but sometimes failed to accurately perceive any notes for measures at a time. It is therefore possible that a neural network trained on “organic” MIDI, rather than MIDI that has been reverse-engineered from finished sound, would be more effective. No such data set is known to exist in a practically accessible way at this time. The tokenization process is another instance of data reduction that deepens this inaccuracy. Tokenization, though it most likely resulted in the loss of some data, is essential to rendering the data intelligible to the model.

Opaqueness surrounding the very popularity metric that pervades this research also limits the conclusions that can be drawn. The exact manner in which Spotify quantifies popularity is not understood and aims primarily to make recommendations to modern music consumers. As such, it is likely a poor representation of many works, such as classical compositions, that have endured for centuries but attract fewer plays from the modern listener. At the same time, because the goal of this research is to be relevant to musicians aiming to maximize plays by the modern consumer, this limitation is not considered serious within the scope of this paper.

---

[9] All training and testing was completed on a 2021 M1 iMac with 16 gigabytes of memory, running macOS Ventura 13.2.1. No GPU acceleration was leveraged. Anaconda Python 3 and Tensorflow 2 were used.

Conclusion & Future Research

The results of this process clearly indicate that, according to all present knowledge, a neural network is not an effective tool in predicting the popularity of a musical piece based solely on compositional data. This research also provides compelling insight into the little-researched field of composition-popularity correlation. Though no definite conclusion can be drawn, it seems unlikely that such a correlation exists to a significant degree. The most critical understanding to be drawn from this research is that neural networks are not, at present, destined to play an integral role in the compositional process, perhaps with the exception of generative applications.

Future research should investigate the efficacy of non-convolutional neural networks in this task. A more effective loss function must also be determined. Identifying the threshold at which a significant correlation between an input vector[10] and a popularity exists is essential. Going forward, researchers should expose neural networks to varying degrees of information, alternately providing and withholding, among other things, lyrics, waveform audio, metadata, and artist information. Once the determinants of popularity are identified more conclusively than they are at present, it will become easier to evaluate neural networks with an objectivity that stems from an understood, expected behavior.

[10] It is common practice to reduce model inputs to vectors in higher-dimensional spaces. These vectors, when they encode multiple features of an input item like a piece of music, are often called “feature vectors”.

References

Arakelyan, S., Morstatter, F., Martin, M., Ferrera, E., & Galstyan, A. (2018). Mining and forecasting career trajectories of music artists. *Proceedings of the 29th on Hypertext and Social Media*, 11-19. https://doi.org/10.1145/3209542.3209554

Bittner, M., Bosch, J., Rubinstein, D., Meseguer-Brocal, G., & Ewert, S. (2022). A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation. *arXiv*. https://doi.org/10.48550/arXiv.2203.09893.

Cai, Y., & Zheng, Z. (2022). Prediction of news popularity based on deep neural network. *Hindawi*, *2022*. https://doi.org/10.1155/2022/8280036

Carnovalini, F., & Rodà, A. (2020). Computational creativity and music generation systems: An introduction to the state of the art. *Frontiers in Artificial Intelligence*, *3*(14). https://doi.org/10.3389/frai.2020.00014

Chen, G., Kong, Q., Xu, N., & Mao, W. (2018). NPP: A neural popularity prediction model for social media content. *Neurocomputing*, *333*, 221-230. https://doi.org/10.1016/j.neucom.2018.12.039

Dhariwal, P., Jun, H., Payne, C., Kim, J., Radford, A., & Sutskever, I. (2020). Jukebox: A generative model for music. *arXiv*. https://doi.org/10.48550/arXiv.2005.00341

Fradet, N., Briot, J., Chhel, F., Seghrouhni, A., & Gutowski, N. (2021). MidiTok: A Python package for MIDI file tokenization. *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*. https://archives.ismir.net/ismir2021/latebreaking/000005.pdf

Gao, A. (2021). Catching the earworm: Understanding streaming music popularity using machine learning models. *International Conference on Environmental and Engineering Management, 253*. https://doi.org/10.1051/e3sconf/202125303024

Huang, L., & Song, Y. (2022). Intangible cultural heritage management using machine learning model: A case study of northwest folk song Huaer. *Hindawi, 2022*. https://doi.org/10.1155/2022/1383520

Jabade, V., Deshpande, V., & Kumar, K. (2019). Music generation and song popularity prediction using artificial intelligence - An overview. *International Journal of Computer Applications, 182*(50), 33-39. https://doi.org/10.5120/ijca2019918762

Lin, Q. (2022). Music score recognition method based on deep learning. *Hindawi, 2022*. https://doi.org/10.1155/2022/3022767

Martín-Gutiérrez, D., Peñaloza, G., Belmonte-Hernández, A., & García, F. (2020). A multimodal end-to-end deep learning architecture for music popularity prediction. *IEEE Access, 8*, 39361-39374. https://doi.org/10.1109/ACCESS.2020.2976033

Novello, A., Par, S., McKinney, M., & Kohlrausch, A. (2013). Algorithmic prediction of inter-song similarity in western popular music. *Journal of New Music Research, 42*(1), 27-45. https://doi.org/10.1080/09298215.2012.749919

Panwar, S., Rad, P., Choo, K. R., & Roopaei, M. (2019). Are you emotional or depressed? Learning about your emotional state from your music using machine learning. *The Journal of Supercomputing, 75*, 2986-3009. https://doi.org/10.1007/s11227-018-2499-y

Paper, D. (2021). *TensorFlow 2.x in the Colaboratory Cloud*. Apress Berkeley, CA. https://doi.org/10.1007/978-1-4842-6649-6

Shi, S., Xi, S., & Tsai, S. (2021). Research on autoarrangement system of accompaniment chords based on hidden Markov model with machine learning. *Hindawi, 2021*. https://doi.org/10.1155/2021/6551493

Sturm, B. L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., Hadjeres, G., Deruty, E., & Pachet, F. (2019). Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, *48*(1), 36-55. https://doi.org/10.1080/09298215.2018.1515233

Trpkovska, M., Kajtazi, A., Bexheti, L., & Kadriu, A. (2019) Applying data mining and data visualization within the scope of audio data using Spotify. *12th IADIS International Conference Information Systems*, 197-204. https://doi.org/10.33965/is2019_201905l025

Wu, J., Liu, X., Hu, X., & Zhu, J. (2020). PopMNet: Generating structured pop music melodies using neural networks. *Artificial Intelligence*, *286*. https://doi.org/10.1016/j.artint.2020.103303

Xu, Y., Wang, M., Chen, H., & Hu, F. (2022). Prediction model of music popular trend based on NNS and DM technology. *Hindawi Journal of Function Spaces*, *2022*. https://doi.org/10.1155/2022/6104056

Yang, L., Chou, S., Yang, Y., & Chen, Y. (2017). Revisiting the problem of audio-based hit song prediction using convolutional neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 621-625. https://doi.org/10.48550/arXiv.1704.01280

Zhang, W., Lei, W., Xu, X., & Xing, X. (2016). Improved music genre classification with convolutional neural networks. *Interspeech*, 3304-3308. http://dx.doi.org/10.21437/Interspeech.2016-1236