Le Recommandeur

Abdullah Palaz [1] **Mete Mert Birdal** [1] **Gökalp Özer** [1]

Abstract

The vast amount of available data creates the problem of data overloading. In this environment, finding relevant/important/needed data points (i.e. websites, products, videos, or songs as in our case) is hard and is a problem that needs to be tackled. It is being tackled using recommendation systems. Music is an interesting and needed domain to be worked on because we listen to music all day and every day. Not just every day but everywhere, doing everything too. We work and listen to music, we are at the gym, working out, we listen to music, we study and we listen to music, we travel and we listen to music and this list can go way longer. But listening to that much music creates a problem where this particular person that listens to too much music becomes new musicless, this is the aforementioned problem projected in the music domain. To tackle this problem, we suggest a content-based music recommendation system. We extract audio features from songs' audio and lyrics to recommend songs.

1. Introduction

The amount of data creation increased vastly, with the digitalization of everything. As much as 64.2 Zettabytes of data is estimated to be created in 2020 and 181 Zettabytes of data creation is estimated for 2025 (1 Zettabyte = 1e+12 Gigabyte). Even though only 2% of the data created in 2020 was stored and retained into 2021 (Holst, 2021), and this vast amount of available data creates the problem of data overloading. In this environment, finding relevant/important/needed data points (i.e. websites, products, videos, or songs as in our case) is hard and is a problem that needs to be tackled. This problem is being tackled using recommendation systems. These recommendation systems are being used in various domains by various companies as Amazon uses a recommendation system to recommend products, Spotify recommends songs in different ways and YouTube recommends videos and all of these companies enlarges their profits using these systems.

Music in this way is an interesting and needed domain to be worked on because we listen to music all day and every day. Not just every day but everywhere, while doing everything too. We work and listen to music, we are at the gym, working out, we listen to music, we study and we listen to music, we travel and we listen to music and this list can go way longer. But listening to that much music creates a problem where this particular person that listens to too much music becomes new musicless. This is the aforementioned problem projected in the music domain. So, it is an urge to find new music to listen to. This project's goal is to recommend you music that you would like. As known, music is just audio and, audio is feature-rich data. Our approach is content-based. We extract audio features from songs' audio and lyrics to recommend songs. This paper structured as follows: Section 2 discusses work on recommendation systems and music recommendation systems, Section 3 introduces our dataset, Section 4 explains our approach, Section 5 shows and discusses our results and Section 6 summarizes our work.

2. Related Work

Since recommendation systems are being used in various domains, there are a few approaches to them: collaborative filtering, content-based, utility-based, demographic-based, knowledge-based and hybrid-based (Fayyaz et al., 2020). In the music domain, collaborative filtering and content-based approaches are widely used as well as hybrid models.

Recently, researches on this domain try to include multi-modalities mostly using emotion extracted from different sources. (Chang et al., 2017) try to recommend songs for stress relief and to do that, use electroencephalography feedback as an enhancement to the classical methods.

In 2018 (Ayata et al., 2018) enhanced music recommendation algorithms by using the "mood" of a person (extracted by Wearable Physiological Sensors). In 2019, (Chang et al., 2017) extracts emotion from the music's audio itself. In 2020, (Nath et al., 2020) tried to find the "type" of music

[1]Department of Computer Engineering, Hacettepe University, Ankara, Turkey. Correspondence to: Abdullah Palaz <abdullahpalaz@hacettepe.edu.tr>, Mete Mert Birdal <metebirdal@hacettepe.edu.tr>, Gökalp Özer <gokalpozer@hacettepe.edu.tr>.

Proceedings of the 38^{th} *International Conference on Machine Learning*, PMLR 139, 2021.

based on their lyrics.

Also, ACM carries out a challenge annually, RecSys (Corona et al., 2021), to get both main approach models as well as creative models.

As can be seen from the recent developments, recommendation models mostly are being enhanced using information from sources other than audios. The feasibilities of these approaches are open to debate as no one probably would use an electroencephalography device for music recommendation but it is easy to see that a regular, classic recommendation model is not enough these days.

3. Data

Our dataset consists of about 1.6K music with 19 genres which we collected using spotDL (Malhotra & Gerchak, 2018), a command-line tool for music downloading. This genre number is not exactly known because Spotify (where the music tags are retrieved) does not hold this genre information in classical genres (e.g. pop, rock, etc.), instead, there are Australian pop, Canadian pop, dance-pop, etc.

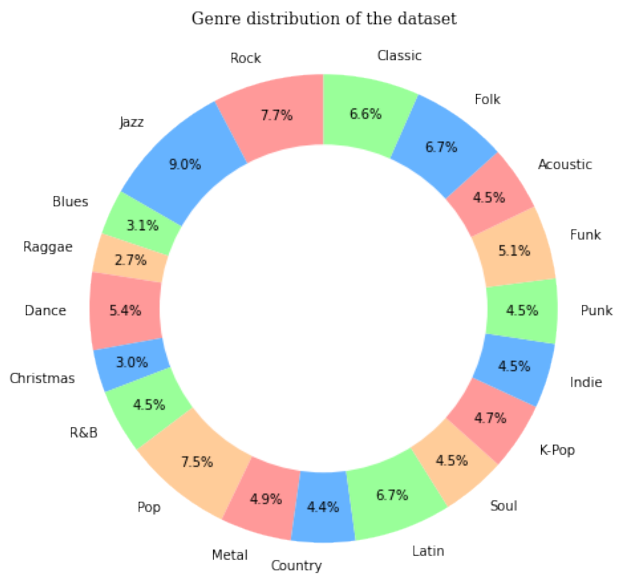

Figure 1. Music genres and their distribution in the dataset.

Song lyrics are also collected to be used in the extraction of the "mood" information of the song for better representation of each song sample. We used Genius API to collect lyrics. There are some problems with the lyrics collection. Unintentionally, we collected songs in languages other than English. This creates a problem because the NLP model we used was trained in English so, with non-English lyrics, the model will not work properly and in the intended way. Also, due to Genius API, some songs' lyrics are meaningless meaning that they have been collected in a wrong way. These problems are hard to tackle even in a small dataset like ours so we had to let them be.

Our dataset also consists of duplicate songs but this is no problem because the most harm this will do is recommending the same song twice. This is not a big problem. Also, this problem should not happen in a real-life application because the provided data (thinking about Spotify dataset), will most probably be clean and have no problems such as us.

One problem with dataset collection is bias. In a general sense, a dataset should not be biased towards an orientation. We too were careful on that matter. However, since we wouldn't be able to know about all the songs we had collected, we observed that our dataset is a bit biased towards the Jazz genre.

3.1. Audio Features

Audio features are the description of an audio signal. There are 4 audio features that we are using where there are 26 statistical variables that define these features. All features represent unique and useful information. These features are:

- Chroma STFT
- Spectral:
  - Spectral Centroid
  - Spectral Bandwidth
  - Spectral Roll Off
- Zero Crossing Rate
- MFFC: 20 mean scores

4. The Approach

Our approach consists of 2 parts: feature extraction and recommendation. For the feature extraction part we initially wanted to extract features automatically, instead of by hand because we have little domain knowledge, and even though we would have enough domain knowledge, we probably wouldn't perform better than a feature extractor. Our candidate for that job was 1D CNN. But due to computation power and time limitations, we couldn't manage to achieve this, so we extracted audio features by hand using librosa (McFee et al., 2015), a Python library, to create a dataset. As mentioned in Section 3.1, we have 26 audio feature columns.

Mood extraction from song lyrics has been done using NLTK's Sentiment Intensity Analyzer (Hutto & Gilbert, 2014). This is a trained model for calculating sentiment intensities to sentences and outputs two scores for a sentence: one is for positive orientation and one is for negative.

Algorithm 1 Polarity to Mood

Input: lyrics x
$positive = SentimentAnalyzerPositiveScore(x)$
$negative = SentimentAnalyzerNegativeScore(x)$.
if x is empty **then**
  $mood = 0$
else
  if $positive - negative > 0.05$ **then**
    $mood = 1$
  else
    $mood = 0$
  end if
end if

We used an algorithm 1 to convert these scores to one of the ternary values (i.e. -1, 0, 1) to be used as the mood information in the dataset.

So, in total, we have 27 features for a song that we believe represents the song truly.

For recommendation, we feed the data into a neural network we created. This neural network Dense layer with 20 neurons and an output layer.

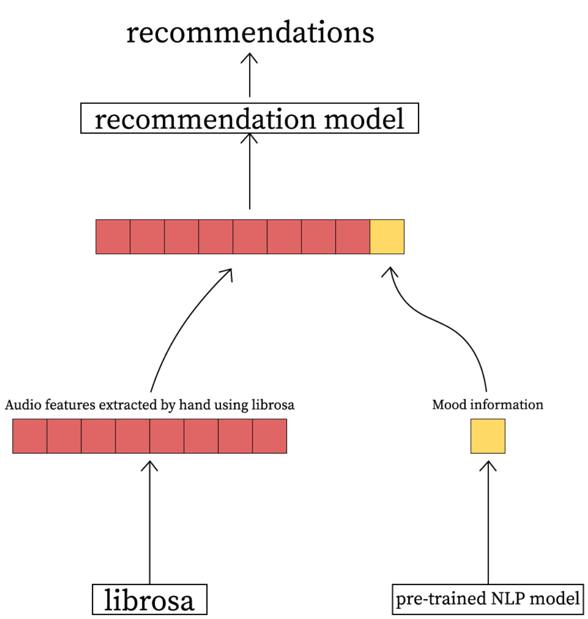

Figure 2. Figure demonstrating our approach.

5. Experimental Results

For evaluation, we used accuracy as the metric. Since the recommendation is highly human-dependent, the scores we got are open to argue. A general score for such a model may be not set globally since ”liking” differs from person to person. But to provide insight, for RecSys Challenge 2018, the highest performing team in main track performed 0.2241 R-precision (Chen et al., 2018). Our evaluation is the same with r-precision so we can compare our results with the results of this challenge. We have conducted our experiments with 9 people, including 3 authors of this paper. Users are asked to label 50 songs as 0 and 1, didn’t like - neutral and liked respectively. Then, we fed these songs to our model to get recommendations. We generally get the 10 highest-scored songs.

Scores we got our debatable, in many ways. Our testing team is too small, only 9 people are not enough for evaluation of a model, there should be more people just like in RecSys Challenge. Also, scores vary too much, this is due to the human intervention of the scores, which again reduces the reliability of these scores. Other metrics may have been used, but as mentioned, time limitations stopped us here.

There are some limitations to our approach. Firstly, we extracted audio features by hand and this process requires domain knowledge to select which features to extract. We believe that our knowledge on this domain is a bit less than needed so automatic feature extraction models probably will do better than our model.

One other limitation of our model is that it requires lyrics in English only. We tackle the problem of not having lyrics (instrumental songs) but tackling having non-English lyrics is a hard problem. Our Sentiment Intensity Analyzer model could perform better, and we believe that this may be due to the following: we may misuse the model, the model may not be presented for this topic, the model may perform too poorly than the state-of-the-art model on this matter. Hence, there can be an improvement here.

Also, the mood of the song does not solely depend on the lyrics. Music plays an important role in this job so there can be an improvement here too.

Dataset is a crucial part of this domain and we believe that a wider (including more genres), larger (including more songs) dataset with the same model will perform better than our current solution. This can be fixed easily by companies such as Spotify since they have a huge collection of music which is a better dataset than ours.

Our approach is user-specific, so there is no one universal model that can recommend songs to anyone, so this might be a drawback also. We need to train a model for each person. But the training of the model is fast, so there should not be a scalability problem.

We see no enhancement when using mood information as can be seen from Table 2 and this could be due to a few things: our sample space is too small, only 3 people are not enough to demonstrate the effect of this feature, this feature is not extracted well enough so it actually decreases the overall performance but due to low sampling we didn't observe that good enough, the proportion of mood feature to audio features is very low (only 1 feature in 27 is mood) so the effect of this mood may not be strong enough to change the results of recommendations.

We also provided the Figure 5 to show the effect of each input feature. This shows that mood features affect the model's recommendations about average.

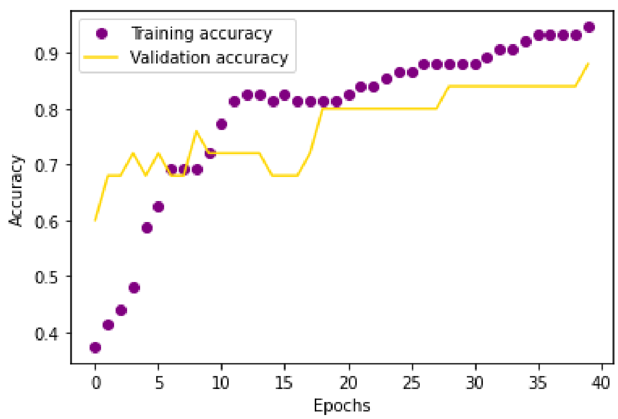

Figure 3. Accuracy of our model while training.

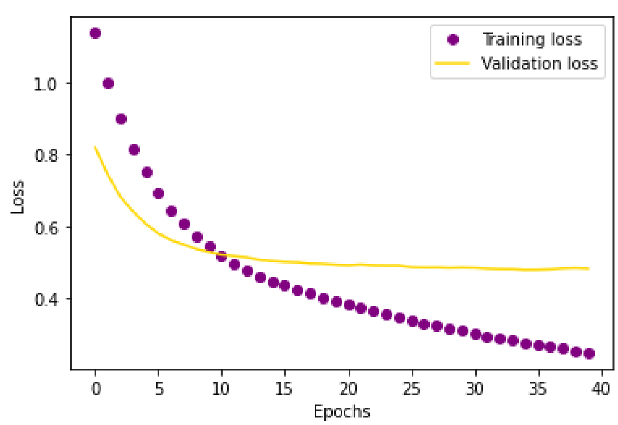

Figure 4. Loss of our model while training.

6. Discussions

We collected data and extracted audio features by hand as well as mood information from lyrics to enhance our model. We believe that our work can be enhanced in many ways

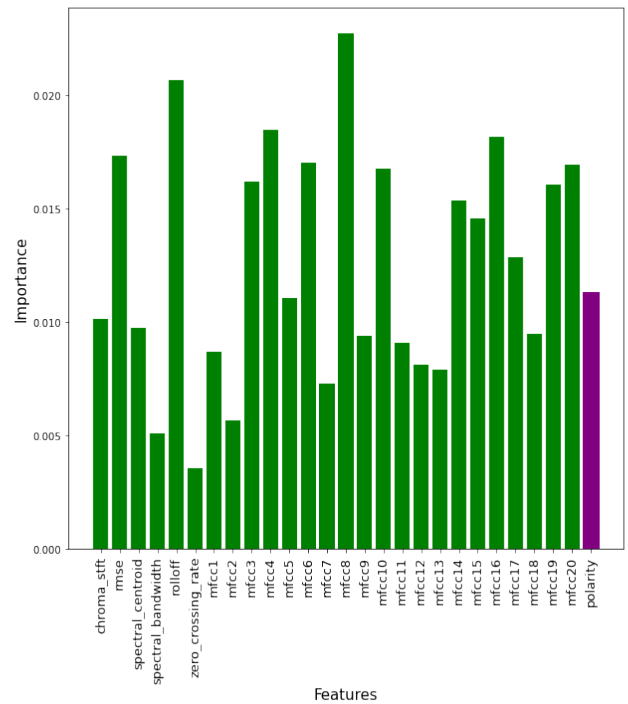

Figure 5. The effect of each input feature.

such as automatic feature extraction can be done, better Sentiment Analyzer can be used, a larger dataset can be collected. Due to time and computation power limitations, we couldn't manage to do these. Since this is only an undergraduate student project, it is not expected to propose a state-of-the-art model. This model, though, can be improved in certain aspects as mentioned in Section 5. Even though the scores we got are debatable, we believe the proposed model does a good job in these conditions and if improved, it can be used in real-life applications.

References

Ayata, D., Yaslan, Y., and Kamasak, M. E. Emotion based music recommendation system using wearable physiological sensors. *IEEE Transactions on Consumer Electronics*, 64(2):196–203, 2018. doi: 10.1109/TCE.2018.2844736.

Chang, H.-Y., Huang, S.-C., and Wu, J.-H. A personalized music recommendation system based on electroencephalography feedback. *Multimedia Tools and Applications*, 76(19):19523–19542, Oct 2017. ISSN 1573-7721. doi: 10.1007/s11042-015-3202-4. URL https://doi.org/10.1007/s11042-015-3202-4.

Chen, C.-W., Lamere, P., Schedl, M., and Zamani, H. Recsys challenge 2018: Automatic music playlist continuation. In *Proceedings of the 12th ACM Conference*

Table 1. Accuracies we got for 9 people.

PERSON	ACCURACY
ABDULLAH	0.1
METE	0.4
GÖKALP	0.7
PERSON 1	0.4
PERSON 2	1.0
PERSON 3	0.8
PERSON 4	0.7
PERSON 5	0.8
PERSON 6	0.6

Table 2. Accuracies including/excluding mood information.

PERSON	ACCURACY W MOOD	ACCURACY W/O MOOD
ABDULLAH	0.1	0.2
METE	0.4	0.3
GÖKALP	0.7	0.7

on Recommender Systems, RecSys '18, pp. 527–528, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016. doi: 10.1145/3240323.3240342. URL https://doi.org/10.1145/3240323.3240342.

Corona, H., Larson, M., Willemsen, M., Pera, S., and Liem, C. Acm recommender systems, 2021. URL https://recsys.acm.org/recsys21/.

Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., and Kashef, R. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences*, 10(21), 2020. ISSN 2076-3417. doi: 10.3390/app10217748. URL https://www.mdpi.com/2076-3417/10/21/7748.

Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., Chiu, C., Zhan, J., Gewali, L., and Oh, P. T-recsys: A novel music recommendation system using deep learning. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, 2019. doi: 10.1109/ICCE.2019.8662028.

Holst, A. Total data volume worldwide 2010-2025, Jun 2021. URL https://www.statista.com/statistics/871513/worldwide-data-created/.

Hutto, C. and Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225, May 2014. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

Li, T., Chan, A., and Chun, A. Automatic musical pattern feature extraction using convolutional neural network. *Lecture Notes in Engineering and Computer Science*, 2180, 03 2010.

Malhotra, R. and Gerchak, E. Spotdl/spotify-downloader: Download your spotify playlists and songs along with album art and metadata (from youtube if a match is found)., 2018. URL https://github.com/spotDL/spotify-downloader.

McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. 2015.

Nath, D., Roy, A., Shaw, S. K., Ghorai, A., and Phani, S. Textual lyrics based emotion analysis of bengali songs. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pp. 39–44, 2020. doi: 10.1109/ICDMW51313.2020.00015.