Hit Song Prediction based on Gradient Boosting Decision Tree

Bang-Dang Pham[1,2], Minh-Triet Tran2[1,2] and Hoang-Long Pham[1,2]
[1]Advanced Program in Computer Science
Faculty of Information Technology
University of Science, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
{pbdang18, tmtriet18, phlong18}@apcs.vn

2020 7th NAFOSTED Conference on Information and Computer Science (NICS) | 978-0-7381-0553-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/NICS51282.2020.9335886

***Abstract*—Record companies invest billions of dollars in new talent around the globe each year. Gaining insight into what actually makes a hit song would provide tremendous benefits for the music industry. In this research, we tackle this question by focusing on predicting rank of hit songs in the next 6 months. Our dataset is used in ZALO AI CHALLENGE 2019 in Hit Song Prediction problem including not only songs but also its information such as composer, artist name, released date, etc. Because of that, while most previous work formulates hit song prediction as a regression or classification problem, we present in this paper how to apply Gradient Boosting technique to treat it as a ranking problem. The resulting best model has a good performance when predicting whether a song is a top 10 dance hit versus a lower listed position with 1.48815 Root Mean Square Error - our result dominates most of the solution in this competition (better than *3rd* ranked solution of 87 in total). Moreover, it is possible to further improve by extracting chords, tones and more information from each song to obtain the highlights of songs and by using linguistics model to offer high-level features of metadata.**

I. Introduction

As the music industry is growing as it is today, music research is getting more attention and investment. Hit song prediction becomes one of the most interesting and beneficial problems for the music industry. The popularity of the song can be evaluated on various aspects such as the number of the ranking in Billboard Chart, digital downloads and the number of listeners. For music streaming service providers, it would be more interesting if song popularity can be predicted so that it can help them identify emerging trends, listen and understand their audience better. Another possible application is an automatic composition model that can help music producers generate new hits or improve existing ones.

In academia, researchers are also interested in understanding the factors that make a song popular. Hit song can be defined as a pattern recognition problem, where the work is to find a connection between popularity measurements and feature representation in a song [1]. Hit song prediction task can be approached in two different ways: using internal factors directly related to the content of the song, including different aspects of audio properties, tune and song lyrics; external factors, factors related to the metadata of songs such as composer name, artist name and release date.

Within internal factors are concerned with the audio properties of music. Our work is an extension of previous research in using deep learning for hit song prediction from audio. Many previous work viewed hit song prediction as a classification problem such as Support Vector Machine (SVM) [2] classifiers based on latent topic features. Besides, some researches also use Convolution Neural Network (CNN) [3] in this problem such as Siamese network [4] or Inception CNN [5].

On the other hand, on external factors side, Salganik [6] proved that the song itself has a relatively minor impact than the social influences for deciding whether a song can be a hit. Beside, Zangerla [7] used Twitter posts in predicting future charts and showed that Twitter posts are important when the music charts of the recent past are available.

In this paper, we represent another way to exploit metadata of a song such as artist name, composer name by applying the Gradient Boosting technique [8]. Although the technique we used is not a breakthrough, our method can potentially leverage it as vital factor to get exposed to higher accuracy with analyzing several parts of each song, influence of composer or artist creating that song, and suitable for releasing song as well.

In what follows, we firstly present some previous proposed works for this problem in Section II. Then, we show you our main approach in III. After that, we explain the reason and how to apply our method by analyzing the dataset feature in Section IV, and finally the experimental setup and results in Section V. We conclude in Section VII. In addition, we provide some potential extracted features and approaches, although they do not work well with our problem's condition.

II. Related Work

A. Internal factor

With the traditional approach, the prediction problem is mainly solved by methods of using CNN in deep learning. The audio segments are converted into the image format of Mel-spectrogram [9]. After that, these images will be passed through CNN layers for feature extraction and are finally passed through softmax layers for the classification task, with the label being the song's rank. CNN classes for this feature extraction process often use different pre-trained models such as ResNet-101 [10] or InceptionV2 [5] to provide quality

features for classification. Some works have significantly improved the use of deep learning models, such as Lang-Chi-Yu using Siamese CNN [4], [11] to measure ranking loss. Besides using deep learning models [4], [5], [10], [11], there are also some approaches to the above prediction problem with many traditional Machine Learning algorithms. Dhanaraj and Logan [1] used Support Vector Machine (SVM) [2] technique to classify whether a song will appear in music charts based on latent topic features computed from audio Mel-frequency cepstral coefficients (MFCC) [12] and song lyrics. Following this work, Ni [13] took a more optimistic stand, showing that certain audio features such as tempo, duration, loudness and harmonic simplicity correlate well with the evolution of musical trends. However, their work analyzes the evolution of hit songs [14]–[16], rather than discriminates hits from non-hits.

B. External factor

In addition to exploiting information inside songs, some works [6], [7] also perform analysis to exploit the metadata of the songs. The extraction of metadata shows how strong correlation between additional information such as singer name, song title, date of production and the popularity of the song. This leads to a linear approach to the hit song prediction problem. Besides, there are several approaches using ensemble learning methods that combine internal features and external features to enhance the effectiveness of the model.

In this paper, we introduce a method that can leverage the external factor optimally. Its learning does not depend on the feature extracted from songs, it yields more information from the song's influence.

III. METHOD

In this section, the LightGBM algorithm is introduced in detail as our main approach in this challenge. LightGBM is a novel Gradient Boosting Decision Tree algorithm (GBDT) (Ke , [17]) which has been used in many different kinds of data mining tasks, such as classification, regression and ordering. The LightGBM algorithm contains two novel techniques, which are the gradient-based one-side sampling and the exclusive feature bundling, respectively.

LightGBM can also handle categorical features by taking the input of feature names. It does not convert to one-hot coding, and is much faster than onpipe-hot coding. LightGBM uses a special algorithm to find the split value of categorical features.

The computational time of traditional GBDT algorithm is often consumed in the construction of a decision tree. The construction of a decision tree needs to find the optimal segmentation point. The general method is to sort feature values and then enumerate all possible feature points. This method wastes time and needs lots of memory. LightGBM algorithm uses an improved histogram algorithm. It divides the continuous eigenvalues into k intervals, and the division points are selected among the k values. So, it is better in training speed and space efficiency than GBDT algorithm. At the same

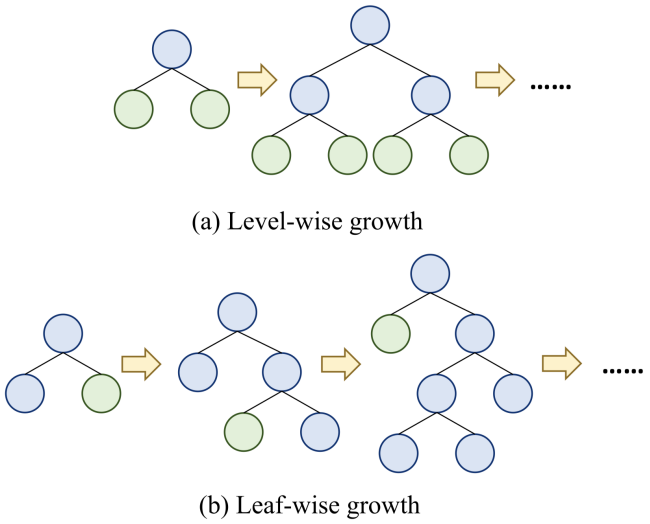

Fig. 1. The generation strategy of tree in LightGBM

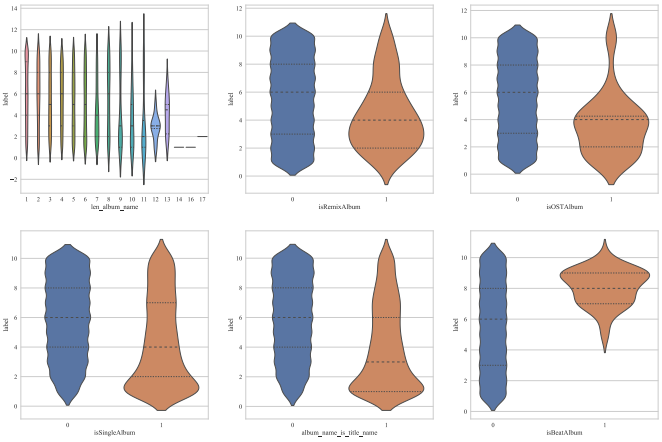

Fig. 2. **Album generated feature**: The vertical axis shows labels, while the horizontal axis shows the album's characteristics from the album's name length to album's genre. The width of each graph stands for the distribution of each unit in the horizontal axis

time, the decision tree is a weak classifier. The use of the histogram algorithm will have a regularization effect and can effectively prevent overfitting. In terms of reducing training data, LightGBM algorithm uses a leaf-wise generation strategy. Compared with the traditional method like level(depth)-wise, the leaf-wise (as illustrated in Figure 1) can reduce more losses when growing the same leaf. Furthermore, the extra parameter is also used to limit the depth of the decision tree and can avoid overfitting. In terms of reducing the number of features, the traditional and mainly used method is PCA. This method is established in the case where the features are redundant, so this method has some limitations. The Exclusive Feature Bundling algorithm (EFB) used by LightGBM puts many features of high-dimensional data together in a sparse feature space to avoid calculation of redundant features.

After researching, the "quality" of songs depends on not

only their significant features or melody, but also its influences - name of artists or singers perform that song, correlation between songs in the same albums; and deep technique could be suffered from this ones (as discussed in Section IV). And machine learning mechanism should be considered for overcoming this task. Within that hyperparameter, we can yield more information from metadata - which is given in familiar datatype in data science problem - tablular. And it can be easily done in tree-based model. Moreover, *leaf-wise tree growth* organization in LightGBM could help this task, without its limitation about convergence speed or resources. With the internal signal from the songs, we extract more knowledge about chord, tonal, melody feature by converting .mp3 file to .wav to use it optimally.

$$CrossEntropy = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{1}$$

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{N}} \tag{2}$$

Due to optimizing problem solution, we do not use Cross-Entropy Loss [1] for this method, it cannot calculate exactly rank prediction and valid ate our model suitably. Then we use Root Mean Square Error (RMSE) [2], even though it is more sensitive to the presence of false data, we could reach to optimal zone for true results.

IV. DATA STATISTIC

Fig. 3. **Top word feature**. The figure illustrates the popularity of words that used to name the song in top 7 (its label < 7) **(Left)** and in top 3 (its label < 3) **(Right)**

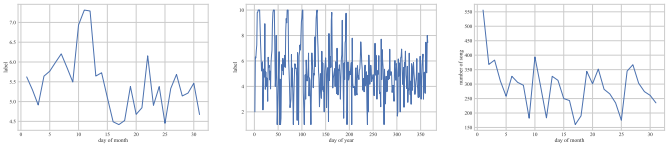

Fig. 4. **Released time feature**

In this problem, we use dataset from ZALO AI CHALLENGE 2019 (challenge.zalo.ai) (ZAC), which is extracted from ZingMP3 - one of the biggest music channels in Viet-Nam. Based on this dataset, we can optimize our model to serve for Vietnamese people and Vietnam music community. This dataset consists of 2 parts:

1) Audio data:
   - Data: Audio file in MP3 format
   - Label: The true rank of each song after 6 months
   - Training set: 9078 tracks
   - Test set: 1118 tracks
2) Metadata: The information of each song including ID, Title, Artist, Composer, Released Time (*.csv file*).

Because our method, Gradient Boosting, focuses on exploiting metadata, besides the ZAC dataset, in addition to providing soundtracks, the data set also provides external data such as the artist's name, song title, song release time. Therefore, in this section, we made statistics about the characteristics of a song as well as the metadata to put the necessary factors into our model.

A. Album generated feature

We first analyze the album information of the dataset. Most features extracted from an album have a strong correlation with the songs in that album. For example, for albums with the length of 8 - 10 songs, the songs in albums usually have a high rank. Besides, we also analyze the properties of albums such as Album Remix, Album OST, Single Album; based on these numbers, we can see that with albums have the type of music like Remix have Rank higher than the Beat album (as illustrated in Figure 2).

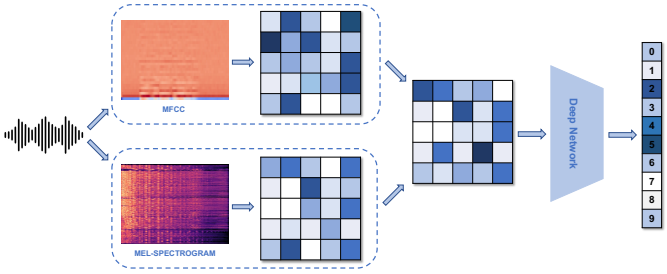

Fig. 5. Pipeline of deep network approach

B. Top five album

Besides, we also created features related to the top 5 albums to analyze data. Unlike rhythm and tonal information, these metadata have a significant influence on the results in the dataset. The albums 8, and 774 have very low variance in rank, which shows that the songs belong to these albums may have the same score. Some top popular album artist have low variance in ranks too. In Figure 6, we shows that some artist like "Hoang Minh Thang" has all songs with low ranks.

C. Top words

Another interesting information we can see is the number of words that appear in the name of songs. Thereby we see, songs of low rank often appear more from the Beat, while songs of higher rank often appear words like Remix, Cover. There are also some words like "Anh" and "Em" and "Tinh Yeu" (as illustrated in Figure 3) which is appear in both high and low-rank songs.

D. Number of artist and composer performs the song per rank

The number of artists and number of composers performs the song/ranks is also a piece of meaningful information. As you can see in the Figure 7, most artists only having a single song, in which artists with 5 songs often have a high rank. In the right statistic, the songs created by 3 composers are likely to have high rank.

E. Time

One critical information we can see is when the song is released. For example, on the tenth and thirteenth of a month, the songs are usually low-rank, probably due to black Friday. Meanwhile, in the middle of the year, many songs have higher ranks, possibly because it is close to holidays of Vietnam or Christmas (as illustrated in Figure 4). Besides, based on the figure the people tend to release the songs in the first day of the month.

V. EXPERIMENT & RESULT

A. Experiments

1) Deep Network Approach: Before choosing LightGBM as the main method in this task, we tried to apply deep technique model into this solution. Firstly, we convert all tracks into Mel-spectrogram [9] and MFCC [12] images, which are representations of the short-term power spectrum of a sound and take advantage from sound feature. Then, these ones would be passed through Neural Network Architecture to learn its visual semantic features. Finally, we solve this task as classification problem with 10 output class probability corresponding to 10 ranks in label (more details would be showed in Figure 5).

But, at that moment, this approach is hard to demonstrate with our condition. Because a song takes approximately 5 minutes to convert into an image. If we want transfer all tracks in *train-set* and *test-set* (10196 tracks in total), we have to spend more than 849 hours just only for pre-processing our data. Moreover, the metric of this problem is RMSE, that is not suitable loss function for classification task (because non-convex function in classification as RMSE is not guaranteed to minimize the cost).

2) LightGBM: Because of using tree-based algorithm, our model is easily overfitting with tons of data and limitation parameters. Then we decide to use average of predictions across 10 lightgbm models from 10 Cross Validation (CV) folds. We tried the maybe more conventional approach of regularising and training on 100% of the training data and found the model significantly fitted compared to 10 CV fold method. Besides that, splitting dataset by album instead of by label (ranks) then weight loss by rank to account for small class imbalance. Finally, as mentioned before, we use RMSE metric for evaluating the final result.

After experiments with a variety of hyperparameter, we finally choose an optimal set for our problem:

1) These are important parameters which control overfitting:
   - *learning_rate* = 0.001
   - *max_depth* = -1 (≤ 0 means no limit)
   - *num_leaves* = 50
   - *min_data_in_leaf* = 5
2) Parameters for categorical values:
   - *categorical_feature* = 0.09
3) Essential part for controlling speed:
   - *feature_fraction* = 0.1
   - *bagging_fraction* = 0.95
   - *num_iterations* = 0.09

Moreover, after recording in our model, we consider the best number of iteration in each fold is 151885 ± 35495.

B. Results

After training in 10 folds of 100% dataset, our best model in *test-set, val-set* is 1.49423 and 1.48815 respectively with *RMSE* metric.

Model	Result
1st place	**1.47030**
2nd place	1.48740
3rd place	1.50260
4th place	1.51050
Our Model	1.48815

TABLE I
COMPARISON WITH OTHER SOLUTIONS

As in the public leaderboard, our model outperforms other solutions except for 1^{st} and 2^{nd} solution (as mentioned in Table I). In this competition, the first-rank and second-rank solutions do not public their method, we cannot explain or analyze clearly why theirs outperform ours.

VI. DISCUSSION

In this section, we introduce some experiments that we experienced. Even though it do not work out our task significantly, its potential can brings lots of interesting things to dive into.

A. Tonal feature

With tonal features, we analyze the dataset on two aspects: continuous and category. While the categories tonal feature is spread evenly, the continuous tonal features give us a clear picture of the difference between ranks. For example, songs with tonal scale equal to 1 or 2 have a spread of rank, while songs with tonal scale equal to zero focus on high-ranking songs (as illustrated in Figure 8).

B. Rhythm feature

For rhythm and tonal information, we use the Music Extractor library for producing features. As you can see in the Figure 9, high ranked songs usually have a large rhythm beat count. However, most features like loudness, bpm, and dance ability do not affect too much the results; each metric numbers are spread evenly from low to high.

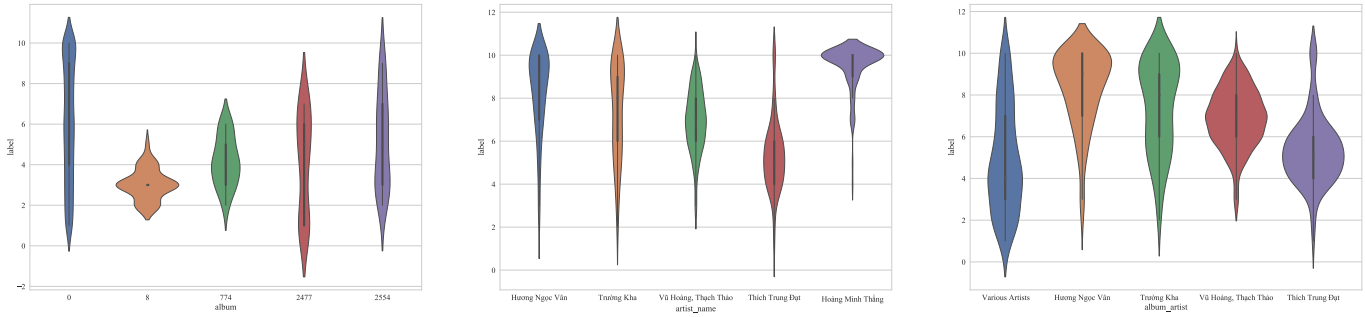

Fig. 6. **Top 5 album**. From left to right, the horizontal axis illustrates top 5 album has the most amount defined by ID, artist name and the most popular artist album

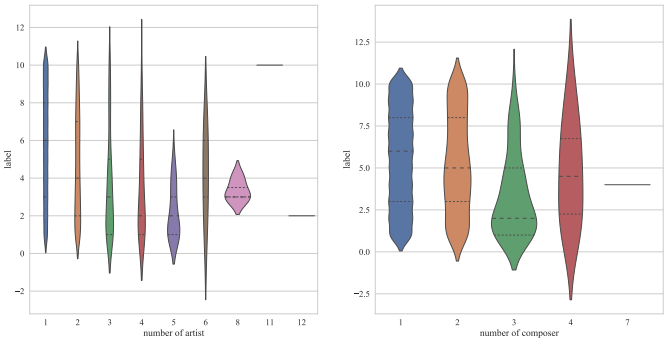

Fig. 7. **Number of artist and composer**. From left to right, the horizontal axis illustrates the number of artist and the number of composer on each label (rank)

C. Lyrics feature

We made some models training exclusively on the lyrics (Hashed) and got RMSE around 3.5, then tried adding this model's rank prediction as a feature. Sometimes saw small improvement, but sometimes not, so ran out of time to incorporate it to final model.

VII. CONCLUSION & FUTURE WORK

In this paper, we present a new approach for Hit Song Prediction problem using Gradient Boosting Decision Tree. Although it is not the best solution with dataset of ZAC, it can leverage more information from external features of songs to predict optimal rank, not only depend on internal features of ones. By using specific dataset for Vietnam community, our model can easily be applied in industrial domain to serve Vietnamese people and contribute to the set of method for solving Hit Song Prediction problem.

For future work, we plan to re-implement deep network approach (in V-A1) for boosting our result as ensemble model. After classifying the label, we draw a statistic of results to define the label that model can predict with high precision score [18] and Cross-entropy loss function. It can help LightGBM approach to reinforce its result for getting smaller RMSE score. Besides that, we plan to use linguistics model [19]–[22] leverage more information from metadata - lyrics, name of songs, artists, order of songs, etc.

ACKNOWLEDGEMENT

This research is supported by research funding from Advanced Program in Computer Science, University of Science, Vietnam National University - Ho Chi Minh City.

REFERENCES

[1] R. Dhanaraj and B. Logan, "Automatic prediction of hit songs." in *ISMIR*, 2005, pp. 488–491.
[2] L. Wang, *Support vector machines: theory and applications*. Springer Science & Business Media, 2005, vol. 177.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[4] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," 2015.
[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.4842
[6] M. J. Salganik, P. S. Dodds, and D. J. Watts, "Experimental study of inequality and unpredictability in an artificial cultural market," *science*, vol. 311, no. 5762, pp. 854–856, 2006.
[7] E. Zangerle, M. Pichl, B. Hupfauf, and G. Specht, "Can microblogs predict music charts? an analysis of the relationship between# nowplaying tweets and music charts." in *ISMIR*, 2016, pp. 365–371.
[8] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.
[9] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.
[11] L.-C. Yu, Y.-H. Yang, Y.-N. Hung, and Y.-A. Chen, "Hit song prediction for pop music by siamese cnn with ranking loss," *arXiv preprint arXiv:1710.10814*, 2017.
[12] W. Majewski, "Mel frequency cepstral coefficients (mfcc) of original speakers and their imitators," *Archives of Acoustics*, vol. 31, 01 2006.
[13] Y. Ni, R. Santos-Rodriguez, M. Mcvicar, and T. De Bie, "Hit song science once again a science," in *4th International Workshop on Machine Learning and Music: Learning from Musical Structure, Sierra Nevada, Spain*. Citeseer, 2011.
[14] R. M. MacCallum, M. Mauch, A. Burt, and A. M. Leroi, "Evolution of music by public choice," *Proceedings of the National Academy of Sciences*, vol. 109, no. 30, pp. 12 081–12 086, 2012.
[15] M. Mauch, R. M. MacCallum, M. Levy, and A. M. Leroi, "The evolution of popular music: Usa 1960–2010," *Royal Society open science*, vol. 2, no. 5, p. 150081, 2015.

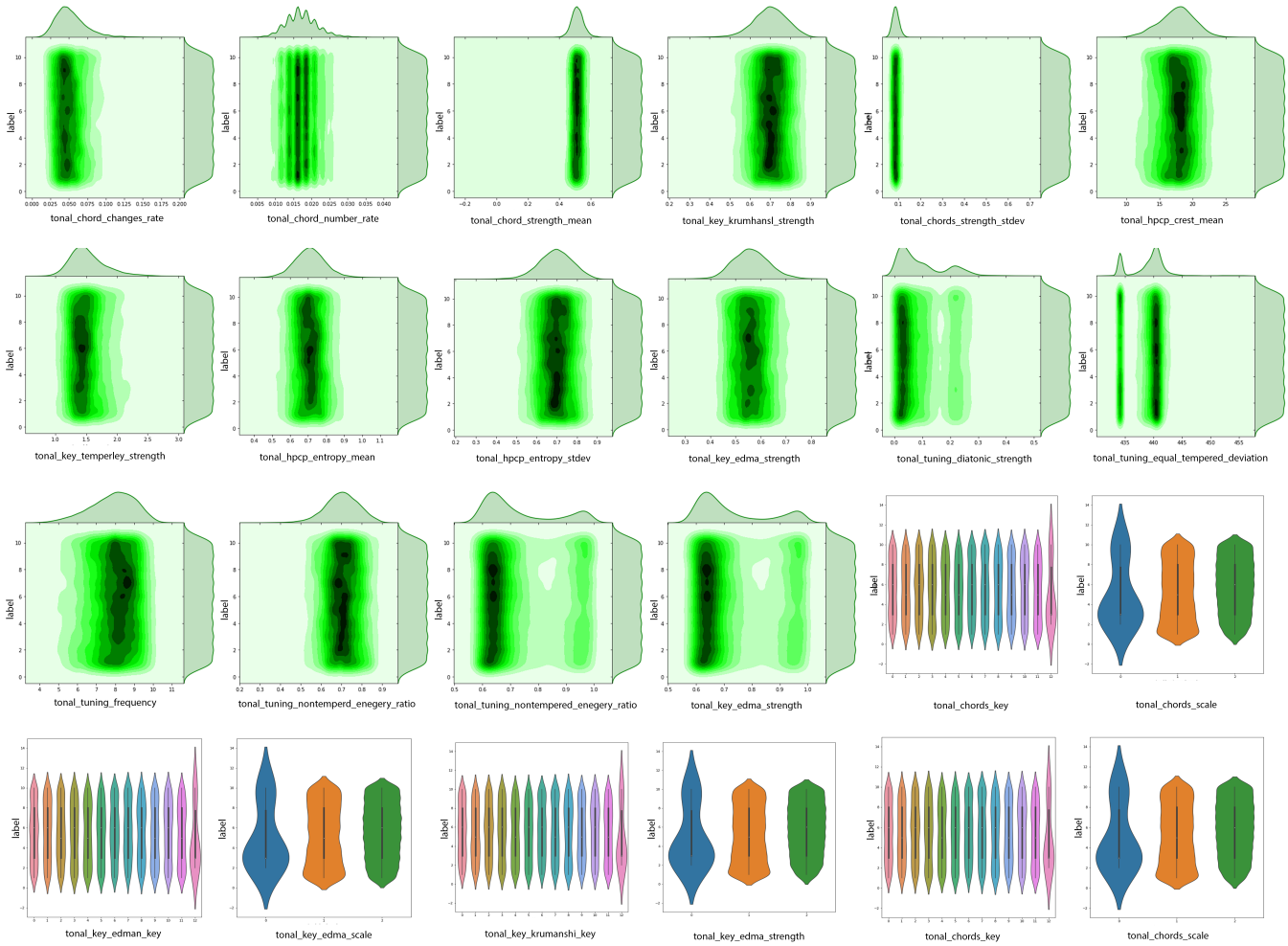

Fig. 8. **Tonal features**. From left to right the horizontal axis shows tonal category feature and tonal continuous feature

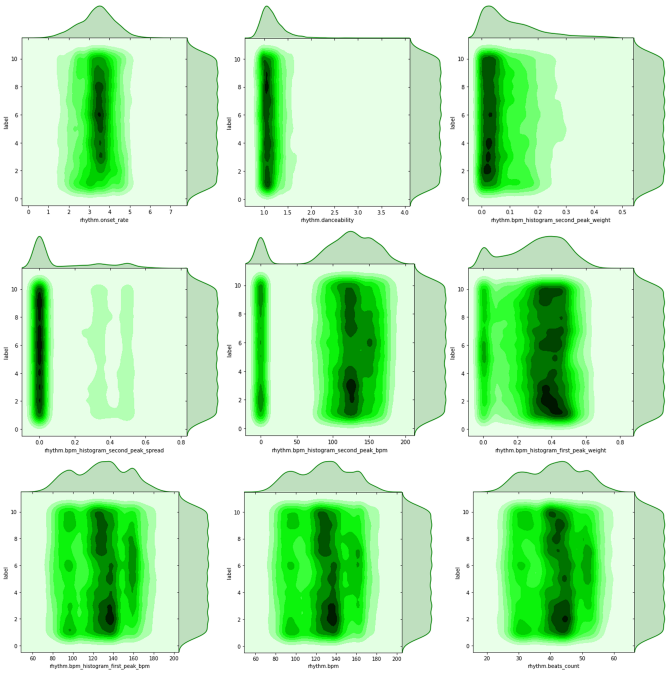

Fig. 9. **Rhythm features**. Each graph represents for each feature related to rhythm feature

[16] J. Serrà, Á. Corral, M. Boguñá, M. Haro, and J. L. Arcos, "Measuring the evolution of contemporary western popular music," *Scientific reports*, vol. 2, p. 521, 2012.

[17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[18] P. Flach and M. Kull, "Precision-recall-gain curves: Pr analysis done right," in *Advances in neural information processing systems*, 2015, pp. 838–846.

[19] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *arXiv preprint arXiv:1808.03314*, 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.