A study of Music recommendation

1st Binh Van Duong
University of Information Technology,
VNU-HCM
18520505@gm.uit.edu.vn

2nd Chien Nhu Ha
University of Information Technology,
VNU-HCM
18520527@gm.uit.edu.vn

***Abstract*—Music has been becoming an essential part of one's life. Thousands of songs are uploaded to the Internet every day. From time to time, music tracks can be represent one's personality. Therefore, this is a potential researching field. Consequently, this paper introduces a proposed music recommendation system for Vietnamese users listening to music on Spotify. The research not only focuses on developing a recommendation system, but also does concentrate on the results of different experiments of different techniques. The approaches in natural language processing (i.e. word tokenization, word2vec) and recommendation system approaches (i.e. content-based, hybrid) are used to enhance the results. Machine learning models such as Light gradient boosting machine (LightGBM), Support vector machine (SVM), Logistic regression, Boosting decision tree are also used to build the best music recommendation system. The source code of this research could be found at the Github repository.**

***Index Terms*—Music recommendation, Music recommendation system, Content-based method, Models-based method.**

I. Introduction

Lately, it has shown the great explosion of entertainment market, especially in music. There are over 60,000 tracks are now uploaded to Spotify every day. It means Spotify is seeing a new track uploaded to its platform every 1.4 seconds [1]. Moreover, previous research [2] has figured out the users preferred listening to music to doing any of other activities as watching television, reading books, and watching movies. This leads to a need of developing a convenient, efficient and user-personalized recommendation system.

To the best of our knowledge, this is the first research of building a music recommendation system for Vietnamese listeners having accounts on Spotify. In this research, we introduce a dataset of Vietnamese tracks collected from Spotify and propose a music recommendation system based on Content-based method and Model-based method. Besides, we also run the experiments of the users' data so as to evaluate the methodologies. The research aims to develop a straightforward and authorized method that is easy to implement. So that, we keep far away from users' personal information. We only employ the information of users' history of listened tracks (i.e. track name, artist names, acoustic features). The study gives the experiments on track names, artist names, acoustic feature with different approaches (i.e. word tokenization, word2vec, hybrid).

Subsequently, we also apply the techniques of natural language processing (NLP) (i.e. word2vec) with the purpose to gain the better performance of the methodologies. We develop 2 main methodologies: Content-based method and Model-based method (LightGBM, SVM, Logistic regression, Boosting decision tree). Besides, we experiment on 2 types of word tokenization: morpheme-based and phrase-based. The result of our experiments are judged by the users from whom we collected data. Consequently, our results are subjective and trustworthy for a recommendation system. In section II, some previous research in the field are introduced. The data set with its details is made clear in section III. The following section IV shows how the features are extracted from the data set. The methodologies applied in the research are laid in Section V. Section VI and VII give the information of proposed recommendation system and experimental results, respectively. Finally, in section VIII, the conclusion are made and some future directions are discussed.

II. Literature survey

In the research [3], the authors built a music genre-based application by applying convolutional recurrent neural network (CRNN) for discovering the most suitable tracks for the target user. The authors used the signal processing techniques to obtain features extracted from the music audio. The study experimented on the data set of 7 genres (i.e. Classical, Electronic, Folk, Hip-Hop, Instrumental, Jazz, Rock) with 700 tracks (1000 to each genre). The authors were successful in building a music recommendation system based on the signal processing and deep learning.

The study [4] indicated that music track has shorter duration than that of a movie or others. And repeated music track recommendations are sometimes appreciated by the listener. The authors pointed out that Content-based and Hybrid approaches have been noticed by researchers when researching music recommendation system.

III. Data set

We leverage the Web API [1] provided by Spotify for collecting the track information from this music platform. With the API, we collect a set of 1781 Vietnamese tracks as the database for our proposed recommendation system. For the experiments, we continue to collect tracks from 3 Spotify users.

The attributes of a track are classified into 2 types: basis information and acoustic features.

- Editorial features: *date_added, artists, track_name, id, uri, track_href, analysis_url.*

[1] https://developer.spotify.com/documentation/web-api/

- Acoustic features: *popularity, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_ms, time_signature.*

A. Content-based method.

The user provides a list of favorite tracks to which he or she listened in Spotify in the last month. Consequently, we have the users' personal list of tracks as following:

- User 1: 13 tracks.
- User 2: 32 tracks.
- User 3: 20 tracks.

B. Model-based method.

With the purpose of collecting the data set for training machine learning model-based, we add attribute *label* to the data set. Subsequently, each user manually labels a set of tracks as *like* (1) or *dislike* (0).

Each user are provided with the data set of nearly threefold larger than the that of Content-based method. We add tracks relevant to the list of tracks to which user listened in the last month. After being labeled manually, it reveals that most of added tracks are labeled 0. This makes the data sets are significantly imbalanced. As a result, we obtain the labeled data set as following and the distribution of the data set of each user showed in Fig. 1.

- User 1: 69 tracks.
- User 2: 126 tracks.
- User 3: 90 tracks.

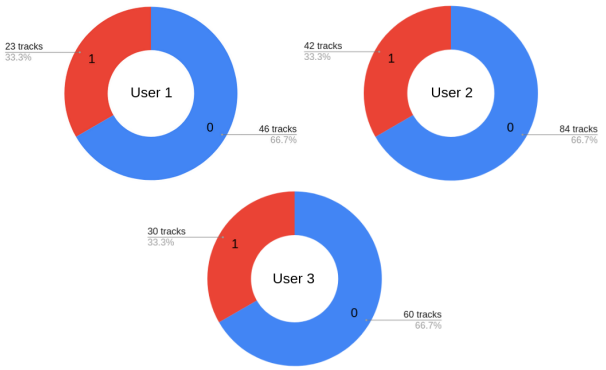

Fig. 1. Label distribution of each users' data.

IV. Feature Engineering

A. Data cleaning and Feature extraction

1) Content-based: We experiment on the *track_name* feature collected from the users' frequently listened track names and acoustic features separately for computing the similarity. To the *track_name* feature, we experiment 2 word tokenization techniques: morpheme-based and phrase-based.

Compute similarity based on the *track_name*. We extract the feature information by the pipeline of two steps: text preprocessing + embedding, visualized in Fig. 2.

- Text preprocessing
  - Do text lowercase ('2 cô TiÊn!!.' -> '2 cô tiên!!.').
  - Remove punctuation ('2 cô tiên!!.' -> '2 cô tiên').
  - Remove duplicate white space ('2 cô tiên' -> '2 cô tiên')
  - Change numbers to words ('2 cô tiên' -> 'hai cô tiên', '2018' -> 'hai nghìn không trăm mười tám').
  - Normalize diacritics ('em cuả ngaỳ hôm qua' -> 'em của ngày hôm qua').
- Embedding
  - Tokenize track name:
    + morpheme-based ('em của ngày hôm qua' -> ['em', 'của', 'ngày', 'hôm', 'qua']).
    + phrase-based [5] ('em của ngày hôm qua' -> ['em', 'của', 'ngày', 'hôm_qua']
  - Vectorize the word tokenization list (['em', 'của', 'ngày', 'hôm', 'qua'] -> [1, 45, 2, 98, 14])
  - Pad each track name to the same length of 17 word ([1, 45, 2, 98, 14] -> [1, 45, 2, 98, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]).
  - Apply embedding word2vecVN [6] ([1, 45, 2, 98, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> [[400 dims]], [400 dims], [400 dims], [400 dims], [400 dims], 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]).
  - Compute mean value for each embedding vectors ([[400 dims]], [400 dims], [400 dims], [400 dims], [400 dims], 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> [0.87, 0.1, 0.8772, 0.98, 0.324, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]).

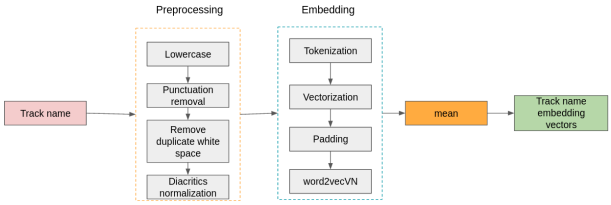

Fig. 2. Feature vector extraction pipeline.

Compute similarity based on the acoustic features. We simply use 13 collected features of a track. So that we have a vector of (1, 13) for each track. This vector will be used for computing the similarity amongst different tracks directly.

2) Model-based: In this part, we add artist names to track names and deliver it to Text preprocessing before embedding phase to minimize noises that negatively affect the process of training.

These are several artist names and track names after Text preprocessing: "jayki", "thu_minh", "bằng_kiều", "chút nắng mùa_đông", "duyên_phận" etc.

We use up Gensim library [7] to apply word2vec model. Thank to this, our artist vectors and track name vectors are able to be compared in terms of similarity based on their embedding space. The word2vec model is trained on the aforementioned preprocessed artist names and track names and

return 1-d vector has the length of 150 for each artist name and track name respectively.

Embedding: we leverage the benefit of embedding matrices that they they can explore the similarity between words in a dataset. Take an artist name "bằng_kiều" as an example. Model is even able to provide the most similar words as well as their corresponding probabilities in Table. I:

TABLE I
THE MOST SIMILAR WORDS TO ARTIST NAME "BẰNG_KIỀU".

Words	Similar rates
"biết"	40.12%
"quên"	39.44%
"xuân"	39.32%
"remix"	38.36%
...	...

Thanks to this acknowledged aspect which has been proved in many NLP researches, we process to embed and afterwards, the returned track name vector and artist name vector are concatenated into a representative vector.

Acoustic features: 13 features to be utilized and to capture more information to the general vector. Eventually, the general vector contains the information of track name, artist name and other useful acoustics features Fig. 3.

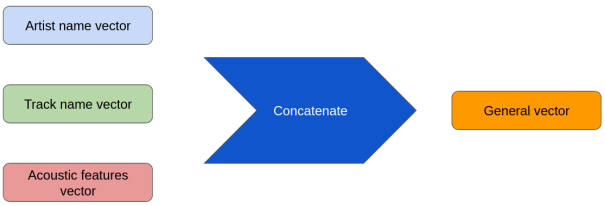

Fig. 3. Feature processing for Model-based method.

V. METHODOLOGIES

A. Content-based method

We compute the similarity score based on 3 types of input vector: track name based feature with morpheme tokenization, track name based feature with phrase-based tokenization and acoustic feature.

Similarity metrics.

- Cosine similarity: the metric bases on the degree between two vectors. The smaller the degree the more similar the two vector are. The metric is popularly used in recommendation research such as [8, 9]

$$sim(A, B) = \cos\varphi = \frac{A \cdot B}{|A| \cdot |B|}, \tag{1}$$

- Pearson similarity: similar to Cosine similarity, Pearson similarity are popularly used in recommendation research [9, 10].

$$sim(A, B) = \frac{\sum_{i=1}^{N}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{N}(A_i - \bar{A})^2}\sqrt{\sum_{i=1}^{N}(B_i - \bar{B})^2}}, \tag{2}$$

While A and B are the feature vector of two tracks generated from the pipelines Fig. 2 or the vector of acoustic features and $\bar{A}$ is the mean value of vector A.

B. Model-based method

We utilize several authorized classifiers in machine learning to find out whether a music user likes or dislikes a song. These models is then directly able to help us predict user's attitude for a new track or unprecedented songs in their lists.

Support Vector Machine. [11] SVM creates a decision boundary to separate data domain into distinct areas. In this context, we have a binary classification problem; therefore, that kind of mentioned boundary assists our system to distinguish *like* and *dislike* label.

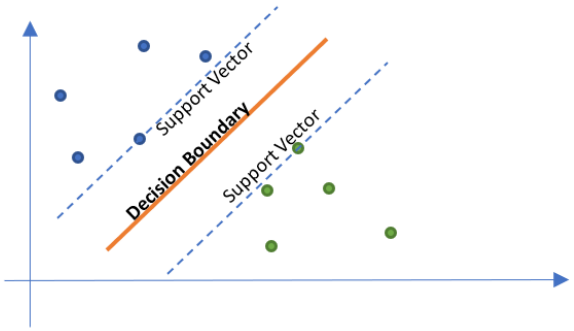

Fig. 4. SVM in 2D space.

Logistic regression. [12] is a classification model using a *Sigmoid* function. This function returns values in range of 0 and 1. Then, researcher has to decide the *Threshold* (usually 0.5) to separate data into 2 different classes (for binary classification). However, our data is going through *a linear function* prior to mapping in *Sigmoid*.

$$f(x) = b0 + b1x, \tag{3}$$

$$y(f(x)) = \frac{1}{1 + e^{-f(x)}}. \tag{4}$$

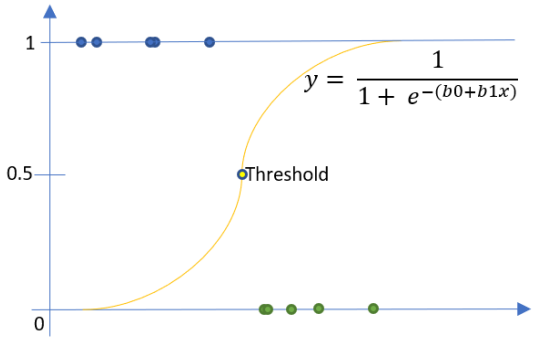

Fig. 5. Logistic regression.

Boosting decision tree. Regarding to Boosting decision tree, we use *DecisionTreeClassifier* [13] which is run under the control of *AdaBoostClassifier* [14] from *sklearn*. Adaptive boosting uses weight to focus on incorrect classification and fix them in the next iteration and boost the result.

- Step 1: Initialize the sample weights:

$$sample\_weight = \frac{1}{number\_of\_samples}$$

- Step 2: Make a decision for each feature, classify data and get the results.
- Step 3: Compute the significance of each tree for its classification based on total_error(sum of error made by a tree):

$$significance = \frac{1}{2} log(\frac{1 - total\_error}{total\_error})$$

- Step 4: Update the sample weights:
  If a sample is incorrectly classified:

  $$new\_sample\_weight = sample\_weight \times e^{significance}$$

  If a sample is correctly classified:

  $$new\_sample\_weight = sample\_weight \times e^{-significance}$$

  New sample weights are then normalized. Step 2 to step 4 are repeated by number of iterations equivalent to number of estimators.
- Step 5: Use the forest of decision trees to make predictions:
  Trees after making predictions will be separated into groups according to their decisions.
  Get the sum of trees' significance in each groups and decide the final prediction by which is larger. Take 3 trees each group as an instance:

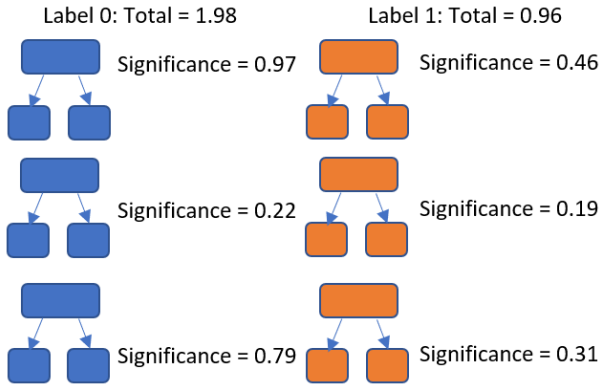

Fig. 6. Step 5: Prediction.

In above example, the answer prediction belongs to label 0.

LightGBM. [15] is a decision tree-based algorithm. It grows tree leaf-wise instead of level-wise and reduces loss more efficiently. This algorithm shows more durable performance in terms of speed when the size of data increases due to GPU support. LightGBM also consumes less memory in processing. Especially, it concentrates on accuracy and becomes a popular model to be chosen in many machine learning tasks.

VI. Recommendation System

In this research, we proposed a music recommendation system for Spotify users listening to Vietnamese music. We build a Web API based on the recommendation models we developed with a user friendly interface. In the web interface, user gives a name of a track to which they have recently listened. Then, the API returns 10 most suitable recommended tracks for the user.

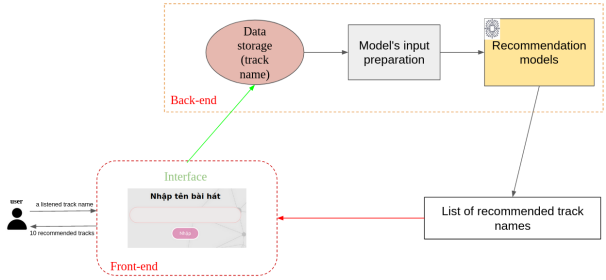

Fig. 7. Web API recommendation system.

VII. Experimental Results

A. Evaluation metrics

Human evaluation. The list of top 10 recommended track names is given to the user in order to get the feedback. Then, number of accepted track names are aggregated to have the final evaluation score. The metric pipeline are visualized in Fig. 8.

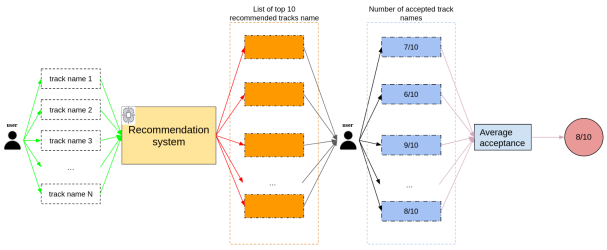

Fig. 8. Human evaluation pipeline.

k-fold Cross-validation. As the data sets for training machine learning models are small in size. We take advantage of k-fold Cross-validation [16] in order to train machine learning model.

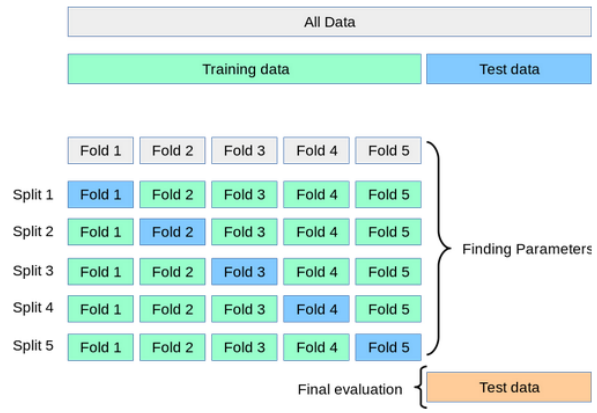

Fig. 9. k-fold Cross-validation procedure [16].

B. Model configuring

SVM. We set up the parameters with kernel='linear', C=1, random_state=10.

Logistic Regression. is configured with C=1.0, fit_intercept=True, intercept_scaling=1, max_iter=100

Boosting decision tree. We set the parameters with min_samples_split=10, max_depth=4), n_estimators=10, learning_rate=0.6)

LightGBM. has boosting_type='gbdt', num_leaves=31, max_depth=- 1, learning_rate=0.1, n_estimators=10000

C. Results

Content-based method. It can be seen that Content-based method with acoustic features gives the highest accuracy on Human evaluation metrics. Besides, phrase-based tokenization's performance is expectedly higher than morpheme-based tokenization.

TABLE II
HUMAN EVALUATION ON 3 TYPES OF FEATURE SELECTION.

	Morpheme-based tokenization	Phrase-based tokenization	Acoustic features
User 1	6/10	7/10	8/10
User 2	5/10	**8/10**	8/10
User 3	**8/10**	7/10	**9/10**

Model-based method. LightGBM model outperforms the other models. Most of models do not have a good result on User 3.

TABLE III
ACCURACY AND F1-SCORE AFTER 10-FOLDS CROSS-VALIDATION.

	User 1		User 2		User 3	
	Acc	F1	Acc	F1	Acc	F1
SVM	0.68	0.09	0.68	0.17	0.68	0.00
Logistic regression	0.82	0.66	0.69	0.16	0.65	0.12
Boosting decision tree	0.97	0.95	0.93	0.88	0.60	0.27
LightGBM	1.00	1.00	0.97	0.95	0.74	0.32

TABLE IV
HUMAN EVALUATION ON 4 MODELS.

	SVM	Logistic regression	Boosting decision tree	LightGBM
User 1	**7/10**	7/10	**8/10**	**9/10**
User 2	**7/10**	**8/10**	**8/10**	8/10
User 3	5/10	6/10	7/10	7/10

VIII. CONCLUSION AND DISCUSSION

In this research, we contribute a data set of 1781 Vietnamese tracks collected from Spotify for developing music recommendation research. In deed, we also collect the labeled data from 3 users using for training machine learning models. A recommendation system are built so as to apply the methodologies developed from the data set. Moreover, our recommendations are judged by the users providing the data for model training. Subsequently, our methodologies at least satisfy the users in this research.

For future directions, we may increase the number of users in the research. We may even apply the signal processing to the track audio in order to train the big deep learning models. Besides, adding more features into our web API music recommendation system.

REFERENCES

[1] Tim Ingham. *Over 60,000 tracks are now uploaded to Spotify every day. That's nearly one per second.* Mar. 2021. URL: https://www.musicbusinessworldwide.com/over-60000-tracks-are-now-uploaded-to-spotify-daily-thats-nearly-one-per-second/.

[2] Peter J Rentfrow and Samuel D Gosling. "The do re mi's of everyday life: the structure and personality correlates of music preferences." In: *Journal of personality and social psychology* 84.6 (2003), p. 1236.

[3] Alexander AS Gunawan, Derwin Suhartono, et al. "Music recommender system based on genre using convolutional recurrent neural networks". In: *Procedia Computer Science* 157 (2019), pp. 99–109.

[4] Markus Schedl. "Deep learning in music recommendation systems". In: *Frontiers in Applied Mathematics and Statistics* 5 (2019), p. 44.

[5] Viet-Trung Tran. *Pyvi tokenizer*. https://pypi.org/project/pyvi/. version 2021.

[6] Xuan-Son Vu. *Pre-trained Word2Vec models for Vietnamese*. https : / / github . com / sonvx / word2vecVN. commit xxxxxxx. 2016.

[7] Radim Rehurek. *Gensim NLP library*. https://pypi.org/project/gensim/. version 2021.

[8] Badrish Chandramouli et al. "Streamrec: a real-time recommender system". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 2011, pp. 1243–1246.

[9] Hailong Wen et al. "Matrix factorization meets cosine similarity: addressing sparsity problem in collaborative filtering recommender system". In: *Asia-pacific web conference*. Springer. 2014, pp. 306–317.

[10] Leily Sheugh and Sasan H Alizadeh. “A note on pearson correlation coefficient as a metric of similarity in recommender system”. In: *2015 AI & Robotics (IRANOPEN)*. IEEE. 2015, pp. 1–6.

[11] *Support Vector Machine library*. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. version 2021.

[12] *Logistic Regression in sklearn*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. version 2021.

[13] *Decision Tree Classifier library*. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. version 2021.

[14] *AdaBoost Classifier library*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html. version 2021.

[15] *Light Gradient Boosted Machine(LightGBM) library*. https://lightgbm.readthedocs.io/en/latest/Python-Intro.html. version 2021.

[16] *3.1. Cross-validation: evaluating estimator performance*. URL: https://scikit-learn.org/stable/modules/cross_validation.html.