UDC 004.04:004.942 https://doi.org/10.15407/pp2021.02.085

A.A. Triantafillu, M.A. Mateshko, V.L. Shevchenko, I.P. Sinitsyn

ALGORITHM AND SOFTWARE FOR DETERMNING A MUSICAL GENRE BY LYRICS TO CREATE A SONG HIT

One of the needs of music business is a quick classification of the song genre by means of widely available tools. This work focuses on improving the accuracy of the song genre determination based on its lyrics through the development of software that uses new factors, namely the rhythm of the text and its morpho-syntactic structure. In the research Bayes Classifier and Logistic Regression were used to classify song genres, a systematic approach and principles of invention theory were used to summarize and analyze the results. New features were proposed in the paper to improve the accuracy of the classification, namely the features to indicate rhythm and parts of speech in the song.

Keywords: genre, rhythm, song, text, classification.

Introduction

The role of music in the modern world is difficult to overestimate - we hear it everywhere. It is hard to determine what affects the listener's consciousness more - the musical component of the song or its lyrics. In the era of streaming services that allow you to listen to music legally and unlimitedly for a small fee, music is a large and profitable business. Spotify, Apple Music and other companies are constantly trying to improve their recommendation algorithms. But how will new to the streaming service users find music they like? They will choose a selection of songs by genre, which they already enjoy. Besides, it is not always clear to young songwriters who to offer their songs, it is hard to understand in which genre they would be most successful potentially. Therefore, it was decided to develop theoretical approaches and corresponding software for mobile and embedded digital systems that would help songwriters and could be used to improve music recommendation algorithms.

Analysis of existing studies and task statement

To date, there are many algorithms of analyzing the genre of the song by musical component, but no product allows you to predict the success of song in a specific genre not only by the content of lyrics or by melody, but also by its rhythm.

In 2019, researchers from the University of South Carolina developed a system using artificial intelligence that recognizes the song genre by lyrics and chords, and their model was trained on more than 5,500 songs that were typical for their genres. The research was closely related to song chords, but the researchers did not perform rhythm analysis [1].

In addition, numerous studies use the «bag-of-words» method to train a model to predict the genre of a song, which allows you to present each song as a set of «important» words. Important words are the words that have a meaning and do not serve just to bind other words. Adam Sadovsky and Xing Chen use approximately 150 songs of 4 genres (rap, hip-hop, rock and country) and they classify the genre of a song by using the «bag-of-words» technique and a special function to determine the weight of the lyrics [2].

The disadvantage of these studies, although the methods of creating features for models and the learning models themselves differ, is that the general essence remains the same: the prediction is carried out only on the basis of the words from the lyrics or its melody and the ways of processing are repeated. The difference of our research is that in addition to words importance metrics, we use features based on the song lyrics rhythm and morpho-syntactic structure, in order to increase the accuracy of the prediction result.

From the analysis of existing theoretical methods and researches it follows that there is a need for new methods to increase the accuracy of song genre classification. The aim of the scientific work is to increase the accuracy

© A.A. Triantafillu, M.A. Mateshko, V.L. Shevchenko, I.P. Sinitsyn, 2021

of determining the song genre by its lyrics by developing software that uses new features, namely the rhythm of the lyrics and its morpho-syntactic structure.

The aim of the study is to analyze researches on classifying song genre by different factors, create a program to determine the rhythm of the lyrics, create a program to determine the number of different parts of speech in the text, train several models to determine the most effective one and compare the results for different models and features.

The scientific novelty of the work lies in the usage of a new method in conjunction with well-known ones to predict the genre in which the song is most likely to become popular by lyrics component.

Description of theoretical methods used

Our work on the analysis of the lyrics consisted of two parts: feature engineering and training of the model on created features. Creating features is the transition from a text representation of lyrics to a numerical representation since text cannot be the input of a machine learning model.

The following metrics were selected as features:

1) ratio of stressed, unstressed syllables to the number of all syllables in the song, the number of undefined words in a song;

2) TF-IDF (term frequency-inverse document frequency);

3) ratio of the number of different parts of speech to the number of all words in a song.

We selected two models, the Bayes classifier and Logistic Regression, and compared their results. The Bayes classifier is a classic model for solving such problems, but since we use features of different types, we assumed that Logistic Regression will show better results.

Preprocessing. The prediction is done using data from the MetroLyrics Dataset, which is a collection of songs-related data, including genre and lyrics from www.metrolyrics.com site. This data allows to analyze the lyrics and find its distinct features. Unfortunately, this file is currently not freely available, so it was taken from a similar project of researchers from the University of California [3]. First of all, all songs written in non-English were removed from the dataset using the DETECTLANGUAGE feature from Google Sheets service. MetroLyrics dataset consists of more than 360 thousand songs in different languages. Since the dataset contains very few entries of Other, R&B, Indie and Folk genres, it was decided not to use them in model training. In addition, all songs without genre or without lyrics, songs labeled «instrumental», corrupted recordings (with a set of characters instead of lyrics) were removed. The distribution after data cleaning I displayed on Fig. 1.

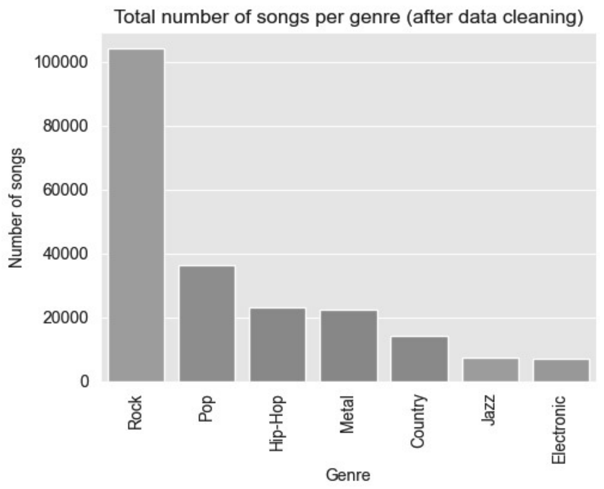

Fig. 1. Total number of songs per genre (after data cleaning)

Determining the rhythm of a song. The method consists in presenting lyrics in the form of a binary sequence, where 1 is the stressed syllable, 0 is the unstressed syllable. The text is pre-cleaned of punctuation and reduced to a unified form - all the letters in the words text are in lowercase.

Let the set L be the set of rows l_1, l_2, ... , l_n: $L = \{l_1, l_2, \dots , l_n\}$. Then the set W – is a set of words w_1, w_2, ... , w_n in a row: $W = \{w_1, w_2, \dots , w_n\}$.

From a phonetic point of view, each word consists of sounds that in turn can be vowel and consonant. Research is conducted using songs in English, and the sounds in words are identified with the help of Carnegie Mellon University Pronouncing Dictionary [4], which consists of more than 134,000 words and their pronunciation. The set of vowels V, according to the Carnegie Mellon University Pronouncing Dictionary, looks like this: $V = \{$'*AA*', '*AE*', '*AH*', '*AO*', '*AW*', '*AY*', '*EH*', '*ER*', '*EY*', '*IH*', '*IY*', '*OW*', '*OY*', '*UH*', '*UW*', '*Y*'$\}$

According to the CMU Dictionary, if there is a number one at the end of the sound designation (for example, "AH1"), then the stress in the word is primarily placed on this sound. Let S be a sentence where the words *a, b, c* are stop-words (words which serve only as a connection for other words): S = "A d, b $e - F$, c a e, g."

Since the number of syllables corresponds to the number of vowel sounds and a vowel sound is present in each syllable, the algorithm of creating a binary rhythm sequence, which was proposed in the work, looks like this:

```
Algorithm 1
1. Break the text into lines;
2. Tokenize (break into elements) the line: S = [“A», «d», «,», «b», «e», « – «, «F», «,», « c», «a», «e», «,», «g», «.»]
3. Convert all the letters to lowercase: S = [«a», «d», «,», «b», «e», « – «, «f», «,», « c», «a», «e», «,», «g», «.»]
4. Clean the text from punctuation: S = [«a», «d», «b», «e», «f», « c», «a», «e», «g»]
5. For each word in a string:
  5.1 If the word is in the dictionary:
    5.1.1 For each sound in the word:
      5.1.1.1 If the sound is vowel:
        5.1.1.1.1 If the sound is stressed:
          5.1.1.1.1.1 Add 1 to rhythm;
        5.1.1.1.2 Otherwise:
          5.1.1.1.2.1 Add 0 to rhythm;
  5.2 Otherwise:
    5.2.1 Add «_» to the rhythm (notation of unknown words (abbreviations, neologisms, profanity, etc.)).
```

Text representation in a form of «Bag of Words». The Bag of Words model considers only the frequency of words appearing in the text and converts the document into a vector of numbers. Each word is assigned a unique number and the number of occurrences in the document. Our work uses an existing CountVectorizer model that uses this principle. Let S be the sentence used in the previous example. Then algorithm of CountVectorizer looks like this:

```
Algorithm 2
1. Tokenize sentences (Algorithm 1, p.2);
2. Unify the words (Algorithm 1, p.3);
3. Remove punctuation (Algorithm 1, p. 4);
4. Remove stop-words: S = [«d”, «e», «f», «e», «g»]
5. Assign an index and a number of occurrences to each word: S = [«d», «e», «f», «e», «g»]; Index = [«d»:0, «e»:1, «f»:2, «g»:3];
   Result = [«d»:1, «e»:2, «f»:1, «g»:1]
```

Statistical metric TF-IDF. TF-IDF (TF - term frequency, IDF - inverse document frequency) is a metric used to assess the importance of a word in a document that is part of a collection of documents. A weight of a word i in a document j is calculated this way:

$$\omega_{i,j} = tf_{i,j} * idf_{i,j} = tf_{i,j} * log\left(\frac{N}{df_i}\right) \quad (1)$$

Where $tf_{i,j}$ is the number of occurrences of the word i in the document j; df_i – the number of documents in which there is a word i; N – the number of documents.

Let S be the sentence used in previous examples. In our work we used an existing TfidfVectorizer model with our own modification of the algorithm. By default, it works according to the following algorithm:

```
Algorithm 3
1. Tokenize sentences (Algorithm 1, p.2);
2. Unify the words (Algorithm 1, p.3);
3. Remove punctuation (Algorithm 1, p. 4);
4. Find TF of the words: S = [“a”, “d”, “b”, “e”, “f”, “ c”, “a”, “e”, “g”]; TF = [“a” : 1, “d” : 1, “b” : 1, “e” : 2, “f” : 1, “ c” : 1, “a” : 1, “g” : 1]
5. Find IDF of the words (depends on the frequency of words in all documents);
6. Vectorize a normalized result.
```

To improve accuracy, we modified the algorithm by creating our own function for text preprocessing. Let S be the sentence used in previous examples. Then the algorithm looks like this:

```
Algorithm 4
1. Tokenize sentences (Algorithm 1, p.2);
2. Unify the words (Algorithm 1, p.3);
3. Remove punctuation (Algorithm 1, p. 4);
4. Remove stop words (Algorithm 2, p.4);
5. Lemmatize words (treat word forms as one word): S = ["d", "e", "f", "e", "g"]
6. Find TF of the words: S = [“d”, “e”, “f”, “e”, “g”]; TF = [“d” : 1, “e” : 2, “f” : 1, “g” : 1]
7. Find IDF of the words (depends on the frequency of words in all documents);
8. Vectorize a normalized result.
```

Determining parts of speech in songs. To determine the entry of different parts of speech in the song, the pre-trained model «en_core_web_sm» from the SpaCy library was used. We used this model on songs that were pre-tokenized and lemmatized. The library functionality returns a tag to indicate a part of speech for each word. The feature is created by counting the number of different parts of speech and foreign words (not English) and their ratio to the number of all words in the song.

Bayes classifier. Since the analysis of text data involves long and multidimensional feature vectors, the learning algorithm should be effective both in terms of classification and in terms of computational speed. These qualities are present in the Bayes training model (Naive Bayes). The method involves dividing the feature into several independent variables and finding an estimate for each of them.

Naive Bayes classifier is based on Bayes theorem, which is an equation that describes the relationship between conditional probabilities. For our question, the Bayes classifier calculates the probability of a particular genre for a given feature. According to Bayes theorem, this probability is calculated as follows [5]:

$$P(G|f) = \frac{P(f|G)P(G)}{P(f)} \quad (2)$$

Where G is the genre, and f is a feature. The feature is divided into independent metrics from the metric (feature) vector, so:

$$P(f|G) = P(f_1|G) * P(f_2|G) * \dots * P(f_n|G) \quad (3)$$

The Naive Bayes classifier has certain drawbacks, namely the fact that the absence of a word in the document (in our case, song) has the same weight as its presence. Obviously, this affects the accuracy of the results because the song is defined by the words that are present in it, not absent. In addition, this classifier does not take into account the frequency of words, which, of course, is extremely important for the song analysis.

To solve these problems, we used a modification of the Naive Bayes classifier – the Multinomial Bayes classifier. It works as follows: the following formula is used to divide the feature into independent metrics:

$$P(f|G) = N! * \prod_{i=1}^{k} \frac{p_i^{n_i}}{n_i!} \quad (4)$$

Where p_i is the probability of the word *i* appearing in all songs of genre *G*; n_i – the number of occurrences of this word in this song; *N* – the number of all words in the song.

Typically, the features used by the classifier are only features to indicate the frequency of occurrence of words in songs (TF-IDF features, for example), but we have added our own features to them to indicate the rhythms of songs and their morpho-syntactic structures. For example, let X be a set of input features, and Y is a set of labels to indicate genres. x_w – classical features to indicate the frequency of words in songs (BoW, TF-IDF), x_r – our features to indicate the rhythm, x_s – our features to indicate the number of different parts of speech in a song, *y* – genres. In the classical solution of such problem, the sets *X* and *Y* look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn} \quad (5)$$

$$Y = y_1, y_2, \dots, y_n \quad (6)$$

After we added our features, the sets began to look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn}, x_{r1}, x_{r1}, x_{r2}, \dots, x_{rn},$$
$$x_{s1}, x_{s1}, x_{s2}, \dots, x_{sn} \quad (7)$$

$$Y = y_1, y_2, \dots, y_n \quad (8)$$

Logistic Regression. By definition, Logistic Regression is intended for the classification of binary classes. Since we use 7 genres for classification, we have chosen the type of logistic regression that can be used to predict more than two classes, namely Multinomial Logistic Regression.

The difference between a Multinomial Logistic Regression and a classic one is that instead of a standard logistic function [6] or a sigmoid function that predicts the probability of a binary event by comparing it with a logistic curve (sigmoid), a softmax function or a normalized exponential function is used, which compresses all values to the range [0,1] and the sum of all elements is 1. The normalized exponential function gives the answer in the form of the probability of the event, which can take more than 2 values. The classifier, which is based on a standard logistic function, calculates the probability of a result *Y* for a given feature *X* as follows [7]:

$$P(Y|X) = \frac{e^{f(x)}}{e^{f(x)}+1} = \frac{1}{1+e^{-f(x)}} \quad (9)$$

Where *f(x)* is a function that consists of the features *x*, and their weight β, which is assigned to the features by the classifier:

$$f(x) = x_0 + x_1\beta_1 + \cdots + x_n\beta_n + \varepsilon \quad (10)$$

The normalized exponential function, in turn, looks like this:

$$softmax(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \quad (11)$$

As is the case with the Bayes classifier, the features, which are commonly used for such studies, were combined with our own ones. For example, let X be a set of input features, and Y is a set of labels to indicate genres. x_w – classical features to indicate the frequency of words in songs (BoW, TF-IDF), x_r – our features to indicate the rhythm, x_s – our features to indicate the number of different parts of speech in a song, y – genres. In the classical solution of such problem, the sets *X* and *Y* look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn} \quad (12)$$

$$Y = y_1, y_2, \dots, y_n \quad (13)$$

After we added our features, the sets began to look like this:

$$X = x_{w1}, x_{w1}, x_{w2}, \dots, x_{wn}, x_{r1}, x_{r1}, x_{r2}, \dots, x_{rn,}$$
$$x_{s1}, x_{s1}, x_{s2}, \dots, x_{sn} \quad (14)$$

$$Y = y_1, y_2, \dots, y_n \quad (15)$$

Analysis of results

To analyze the results, we created a structural and logical scheme of the research (Fig.3-8). Each scheme block has the following structure (Fig. 2):

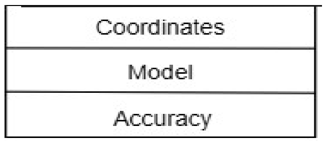

Fig. 2. Structure of structural and logical scheme

To understand the scheme, we need to enter some notations:

- rhythm 1 – 3 features: ratio of the number of stressed syllables, unstressed syllables, unrecognized words to the number of all syllables;
- rhythm 2 – 10 features: 3 of rhythm 1, percentage of unfamiliar words, number of lines, average number of syllables per line, number of words, average number of syllables in a word, average number of letters in a word, average number of letters in a line;
- parts of speech 1 – the ratio of the number of parts of speech to the number of all words, without removing stop-words;
- parts of speech 2 – the ratio of the number of parts of speech to the number of all words, with the removal of stop words.

The scheme is divided into parts according to the model and the main feature: CountVectorizer (Bag of Words) or TF-IDF (Term Frequency – Inversed Document Frequency). Italics highlight the best accuracy result from the part. The result of accuracy was calculated under the following conditions:

- 70% of the data was used to train the model, and to verify it the remaining 30% was used.
- the random state parameter was set to the same value for each experiment. This is necessary so that the same songs always fall into the test set of songs.

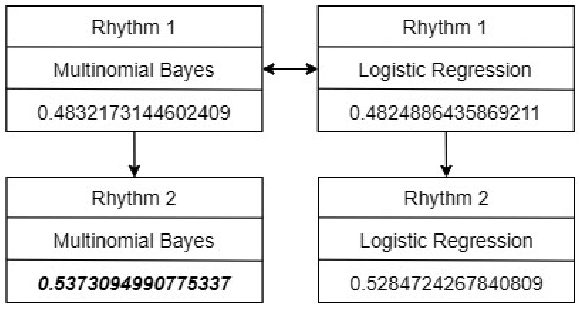

Fig. 3. Structural and logical scheme, p.1

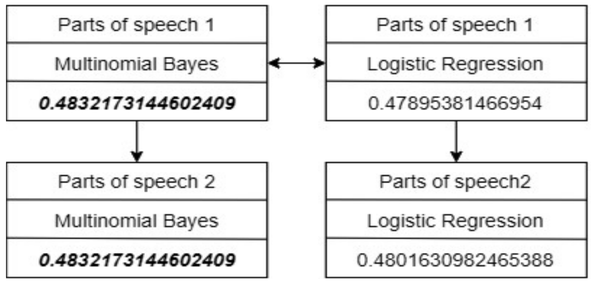

Fig. 4. Structural and logical scheme, p.2

From the results on the scheme it became obvious that the assumption that the Logistic Regression would give a better result was confirmed. To assess the depth of innovation we created the following table (Table 1):

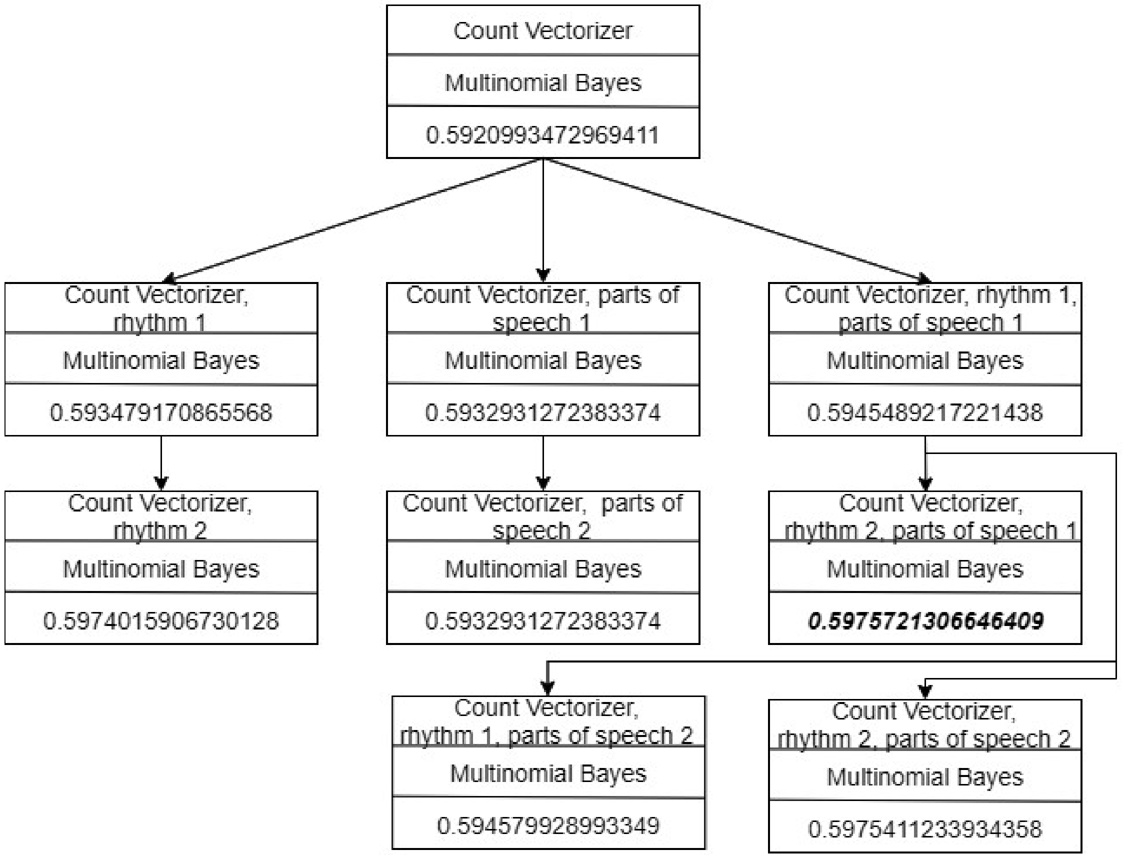

Fig. 5. Structural and logical scheme, p.3

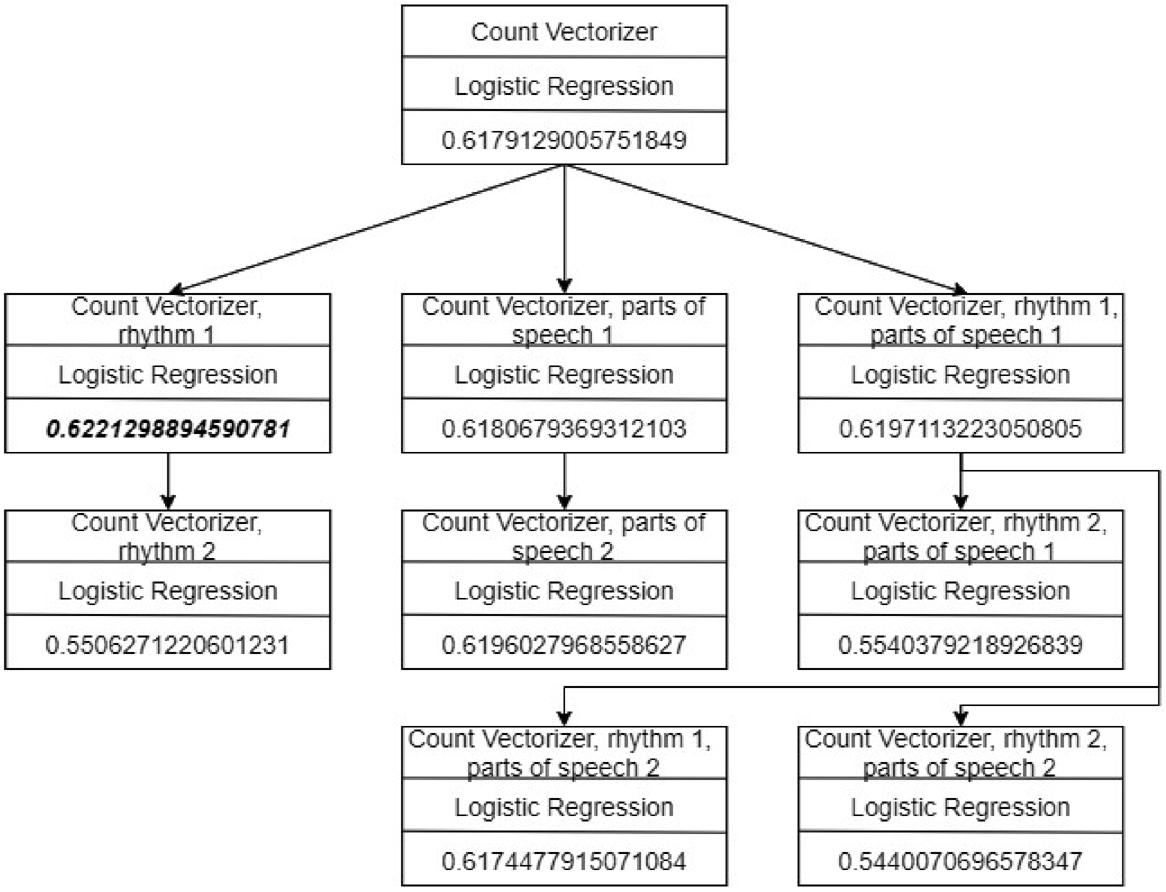

Fig. 6. Structural and logical scheme, p.4

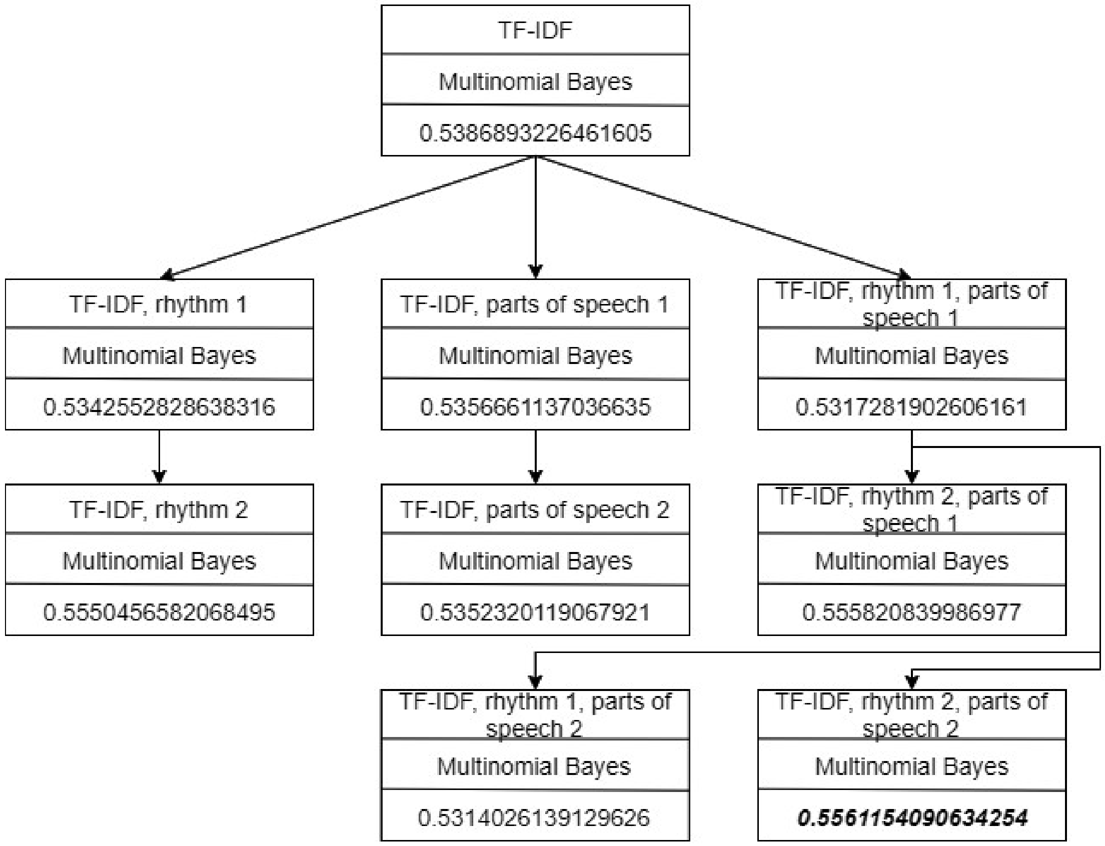

Fig. 7. Structural and logical scheme, p.5

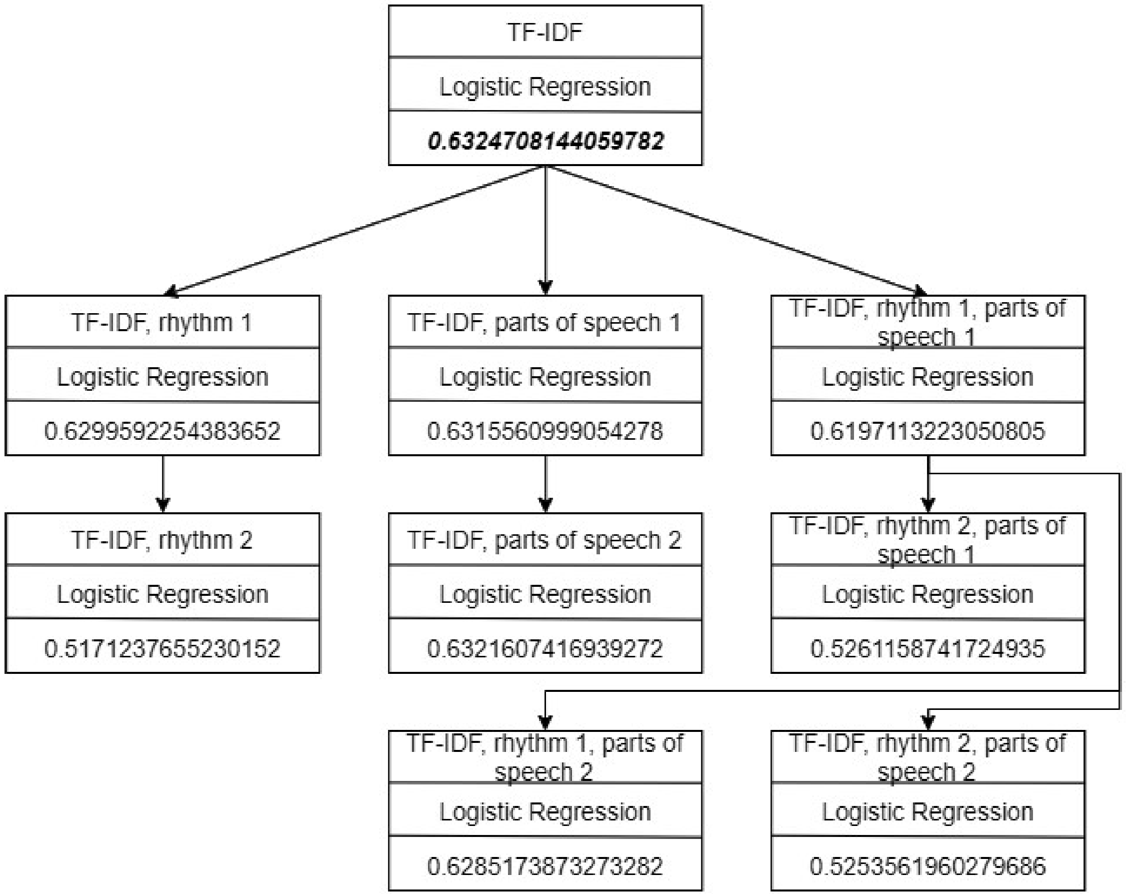

Fig. 8. Structural and logical scheme, p.

Table 1
Evaluation of the research results

№	Name	Labor units (points 0 - 9)	Depth of Innovation (points 0 - 9)	Result	Influ ence
Before					
1.0	BoW[a]	3	0	0.6179	
Proposed					
1.1	R1[b]	7	9	0.6221	+
1.2	PS1[c]	5	5	0.6180	+
1.3	R2[d]	8	8	0.5506	-
1.4	PS2[e]	5	5	0.6196	+
1.5	R1,PS1	5	6	0.6197	+
1.6	R1,PS2	5	6	0.6174	-
1.7	R2,PS1	5	6	0.5540	-
1.8	R2,PS2	5	6	0.5440	-
Before					
2.0	TF-IDF	3	0	0.6324	
Proposed					
2.1	R1	7	9	0.6299	-
2.2	PS1	5	5	0.6315	-
2.3	R2	8	8	0.5171	-
2.4	PS2	5	5	0.6321	-
2.5	R1,PS1	5	6	0.5261	-
2.6	R1,PS2	5	6	0.6285	-
2.7	R2,PS1	5	6	0.5261	-
2.8	R2,PS2	5	6	0.5253	-

[a]BoW – Bag of Words, [b]R1 – Rhythm 1, [c]PS1 – Parts of speech 1, [d]R2 – Rhythm 2, [e]PS2 – Parts of speech 2

Thus, not only the approaches were found to improve the accuracy of defining song genres. Also, the difference in the effectiveness of solution when changing methods was tracked. This allowed us to build a trajectory in space of possible solutions, which led to the best solution. In addition, this trajectory can be improved according to the change in the conditions of the task, in order to obtain the best method for new conditions. This result lies in the plane of management of purposeful receipt of new ideas and can be considered an extension of existing approaches to the theory of invention, for example, a morphological table of possible solutions.

Since the preliminary results were calculated for the same test data, 10 experiments were conducted to calculate the mathematical expectation, using logistic regression for the features that give the best result, namely: TF-IDF; TF-IDF, rhythm 1; TF-IDF, rhythm 1, parts of speech 1. The results of experiments are shown in the graph (Fig. 9):

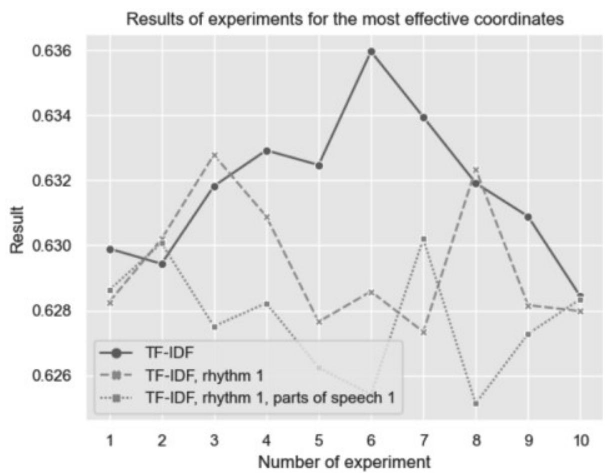

Fig. 9. Results of experiments for the most effective features

To verify the distribution of results, Q-Q (quantile-quantile) graphs were built and Shapiro-Wilk tests were conducted. This showed that the resulting distribution is slightly different from the normal. Based on these results and considering the possible error associated with a small data sample, we concluded

Table 2
Result table

	TF-IDF	TF-IDF, Rhythm 1	TF-IDF, rhythm 1, parts of speech 1
Average	0.6318	0.6294	0.6277
Median	0.6319	0.6284	0.6279
Maximum	0.6359	0.6328	0.6302
Minimum	0.6284	0.6273	0.6251
Standard Deviation	0.0021	0.0019	0.0016
Dispersion	0.0000045	0.0000036	0.0000027

that the results are distributed normally. Thus, the results can be represented by the following table (Table 2):

Therefore, our features in some cases add up to 2% of accuracy to the main method. Therefore, even though the highest accuracy was obtained without the use of new features, further studies of rhythm have great potential.

Conclusions

1. As a result of the work, its purpose was achieved, namely, the accuracy of determining the genre of the song by its lyrics was increased through the development of theoretical approaches and corresponding software for mobile and embedded digital systems that use new factors, namely the rhythm of the text and its morpho-syntactic structure.

2. Scientific novelty is the use of a new method, namely the determining of the text rhythm and its parts of speech, to classify songs by genres by creating new features. The best results were obtained due to the feature TF-IDF (Term Frequency - Inversed Document Frequency) and its combinations with features to indicate rhythm and rhythm with parts of speech. During the research, we realized that the rhythm potential is much larger than the scale of our project, since the presentation of text in the form of a rhythm allows you to binarize any text-like information.

3. The study progressed according to the principles of invention theory, which is reflected in the structural and logical scheme of research, which was built to analyze the results. This made it possible to see that not only approaches were found to improve the accuracy of defining song genres, but also the effectiveness of the solution when changing methods was tracked. Thus, a trajectory was built in the space of possible solutions, which led to the best solution.

4. The created program code was successfully approbated by placing it on a web service for joint software development GitHub [8].

5. The practical application of this innovation is not limited to the obvious musical application, namely the improvement of music recommendation algorithms or assistance for young authors. More generally, the analysis of rhythm will allow to find non-obvious patterns in such texts as:

- political speeches: to edit speech text to achieve the best perception by the audience;
- historical documents: to analyze their authenticity or belonging to a particular historical period;
- promotional texts: to edit the text for the best targeting;
- songs: to find musical plagiarism;
- dialogues from movies: to script features that make movies popular.

References

1. Greer, T. and Narayanan, S., 2019. Using Shared Vector Representations of Words and Chords in Music for Genre Classification. SMM19, Workshop on Speech, Music and Mind 2019, [online] pp.46-49. Available at: <https://www.isca-speech.org/archive/SMM_2019/pdfs/SMM19_paper_19.pdf>.
2. Sadovsky, A. and Chen, X., 2006. Song Genre and Artist Classification via Supervised Learning from Lyrics. [online] pp.1-18. Available at: <https://nlp.stanford.edu/courses/

cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf>.

3. Brennan, C., Paul, S., Yalamanchili, H. and Yum, J., 2018. Classifying Song Genres Using Raw Lyric Data with Deep Learning. [online] GitHub. Available at: <https://github.com/hiteshyalamanchili/SongGenreClassification>.
4. Speech.cs.cmu.edu. 2021. The CMU Pronouncing Dictionary. [online] Available at: <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.
5. Brownlee, J., 2021. Logistic Regression Tutorial for Machine Learning. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning>.
6. Shevchenko, V., 2011. Optimization Modeling in Strategic Planning. CVSD NUOU, pp. 283.
7. Brownlee, J., 2019. Logistic Regression Tutorial for Machine Learning. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning>.
8. Triantafillu A., 2021. Song Genre Predictor GitHub Project. [online] Available at: <https://github.com/triantafillu/Song-Genre-Predictor>.

Received: 18.05.2021

About the authors:

Aleksandra A. Triantafillu
Bachelor student of Taras Shevchenko National University of Kyiv
Publications: 2
29 Lobanovskoho ave., ap.10, 03037, Kyiv, Ukraine
https://orcid.org/0000-0003-2595-8699

Mykola A. Mateshko
Bachelor student of Taras Shevchenko National University of Kyiv
Publications: 2
52-V Evhena Svertyuka str., ap. 42, 02002, Kyiv, Ukraine
https://orcid.org/0000-0002-6203-0577

Viktor L. Shevchenko
Dr.Sc., Prof., Professor at Software systems and technologies Department of Taras Shevchenko National University of Kyiv.
Shovkunenka st. 3-72, 03049, Kyiv, Ukraine.
Publications: more than 300
Foreign scientometric publications: 17
H=3
https://orcid.org/0000-0002-9457-7454

Igor P. Sinitsyn
Dr.Sc., Senior Researcher,
Head of Department at the Institute of Software Systems of the National Academy of Sciences of Ukraine.
Publications: more than 300
Foreign scientometric publications: 3
H=1
https://orcid.org/0000-0002-4120-0784

Affiliations:

Taras Shevchenho National University of Kyiv
Volodymyrska St, 60, Kyiv, 01033
Phone: +380442393333
Fax: +380442393388
E-mail: office.chief@univ.net.ua
E-mail: alexandra00219@gmail.com
nmateshko@gmail.com
gii2014@ukr.net

Institute of Software Systems of the National Academy of Sciences of Ukraine.
03187, Kyiv-187, Acad. Hlushkov avenue,
Phone: +380445263319
E-mail: iss@isofts.kiev.ua