UDC 004.032.26

DEVELOPMENT OF A MUSIC GENRE CLASSIFICATION SYSTEM USING CLOUD RESOURCES

***A.Y. Mishchenko*[1]**, *Y.I. Dorofieiev*[2]

[1] master student of the Department of System Analysis and Information-Analytical Technologies, NTU "KhPI", Kharkiv, Ukraine
[2] head of the Department of System Analysis and Information-Analytical Technologies, doctor tech. Sciences, NTU "KhPI", Kharkiv, Ukraine
anton20435@outlook.com

Nowadays, when the average user has access to numerous online music libraries, there is a need to classify them into subsections so that the listener can find more similar songs that match his preferences. With the widespread use of various music platforms, more and more music is distributed every day, causing chaos for the audience and the platform holders and resulting in the need to solve the task of organizing this music. Clearly, it is impossible to organize and distinguish today's amount of music by conventional means, so the task of creating a convenient way to solve this problem is an important but difficult one, and an accurate classification of music genres is crucial for music platforms to be able to structure music that belongs to various genres [1]. For this reason, the task of music genre classification has recently become very relevant and attracts a lot of attention in the field of audio data mining. The goal of this work is to develop a music classification system for 10 genres based on the analysis of a small section of an audio file.

To date, enough research has been carried out [2], the results of which allow to assess the prospects of using one or another method of solving the problem of genre classification of musical works. The most successful options are SVM (support vector machines), k-NN (k-nearest neighbors algorithm), CNN (convolutional neural networks) and their variations.

The first stage of solving the given task is to determine the features of the audio file, which include MFCCs (mel-frequency cepstral coefficients), spectral centroid, spectral rolloff, and spectral bandwidth. Since CNN has the ability to independently extract features, the mel spectrogram was used for its training. To evaluate the performance, each of the models was tested on the test data set, and metrics were calculated to evaluate the classification accuracy of each class. The obtained results are shown in table 1.

Summarizing, it's clear that the CNN-based model performed best, on average 10–30% better than classical ML models. This is due to the complexity of the feature construction process. Even simply converting an audio signal to a vector of features, which, moreover, should not be correlated, with minimal loss of information is a difficult task, and when it comes to the classification of music genres, the complexity increases many times. Mel-frequency cepstral coefficients, which are today considered the best features in the fields of audio signal analysis and speech recognition, are only a compressed representation of the spectrogram [3]. In turn, the spectrogram is the most comprehensive representation of all sound characteristics, moreover, with minimal loss of information (it is possible to construct the reverse process of converting the spectrogram into an audio signal). Convolutional neural networks are able to "analyze" the spectrogram and extract the necessary features, which is why CNNs classify music at such a high level. The k-nearest neighbors method showed the worst results; on the other hand, it should be noted that k-NN is the simplest, in terms of implementation and computational complexity, among the tested models.

Separately, it is necessary to highlight the issue of further use of the developed ML model and its deployment in a business product. There are two main ways to deploy a model today: cloud-based deployment and embedded model. Embedded model – is a model that is built into a software product or even a device. The main advantages of such models include the absence of delay, system reliability, confidentiality of processed data. ML models, built into microcontrollers, are capable of obtaining significant information from data that would otherwise be unavailable. Disadvantages include limitations in computing power and the practical impossibility of further support or improvement of the built-in model. On the other hand, deployment using cloud technologies requires a separation of the model from the data source. Most often, in this case, the models are installed on separate servers where all calculations are performed, and the input data for the model is transmitted over the network. This type of deployment offers much more flexibility than embedded models. It becomes possible to fully support or improve the model, you can simultaneously run different versions of the model and conduct testing on target groups. The disadvantages of this approach include: lower reliability, insufficient confidentiality of transmitted data, processing delay, and network traffic limits, which is associated with the need of transferring data through the network.

Considering all the advantages and disadvantages, it was decided to use cloud technologies to deploy the model. Google Cloud Platform (GCP) is chosen for deployment. It is one of the most popular provider of computing resources for developing, deploying, and operating applications on the Web and offers lots of the cloud features such as compute engine, cloud storage, cloud run, app engine etc. But the most famous is the Google Kubernetes Engine (GKE). Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Containers is a good way to bundle and run applications, they are similar to virtual machines (VMs), but they have relaxed isolation properties to share the operating system among the applications. Therefore, containers are considered lightweight in comparison to VMs. In a production environment, GKE helps to manage the containers that run the applications and ensure that there is no downtime.

Table 1 – Testing results

Class name	k-NN			SVM			CNN		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Blues	0,57	0,47	0,52	0,73	0,69	0,71	0,91	0,91	0,89
Classical	0,72	0,93	0,81	0,81	0,95	0,87	0,87	0,94	0,90
Country	0,35	0,32	0,33	0,62	0,55	0,58	0,82	0,74	0,82
Disco	0,29	0,45	0,35	0,50	0,39	0,44	0,81	0,89	0,85
Hip-hop	0,30	0,19	0,23	0,51	0,41	0,46	0,90	0,82	0,86
Jazz	0,63	0,72	0,67	0,70	0,88	0,78	0,90	0,82	0,91
Metal	0,66	0,64	0,65	0,95	0,74	0,83	0,93	0,89	0,86
Pop	0,59	0,65	0,61	0,62	0,91	0,73	0,94	0,80	0,88
Reggae	0,43	0,28	0,34	0,90	0,47	0,47	0,87	0,90	0,80
Rock	0,42	0,38	0,40	0,47	0,55	0,58	0,79	0,81	0,85

References:

1. *McKay C.* Musical genre classification: is it worth pursuing and how can it be improved?/ *C. McKay, I. Fujinaga //* In Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006), 2006. Corpus ID: 7531206.

2. *Chettiar G.* Music genre classification techniques/ *G. Chettiar, K. Selvakumar* // International conference on innovation challenges and advances in engineering. – 2021.

3. *Nagesh S. C.* Audio data analysis using deep learning with Python (Part1). 2020. URL:https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html.