**A PROJECT REPORT**

*Of*

# Music Genre Classification Using IBM Machine Learning Service

Project Team Number - 601

*By*

SURANA GAURAV (20BEE0371)

YASH TRIVEDI (20BEE0370)

ANAMAY TIWARI (20BEE0273)

# TABLE OF CONTENTS

# Introduction:

## *Overview:*

Music genre classification is an intriguing field that enables the organization and categorization of a vast array of music based on its inherent characteristics. In this project, we focused on developing a music genre classification system using the K-Nearest Neighbours (KNN) algorithm. Our project utilized the GTZAN dataset, a well-known benchmark dataset in the music genre classification domain. By harnessing the power of machine learning, our objective was to accurately predict the genre of audio tracks, thereby automating the process of genre identification.

The GTZAN dataset, obtained from Kaggle, provided a diverse collection of audio tracks spanning ten different genres, including blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Leveraging various audio feature extraction techniques, such as Mel-Frequency Cepstral Coefficients (MFCCs), spectral contrast, and tonal centroid, we aimed to capture significant characteristics of the audio tracks. These features served as inputs for the KNN algorithm, enabling it to learn genre patterns and make precise predictions.

## *Purpose:*

The purpose of our project was to develop a robust music genre classification system that could be utilized for various practical applications. By automating the genre identification process, our system aimed to provide a time-efficient and reliable solution for music enthusiasts, professionals, and researchers alike. The accurate classification of music genres has significant implications in numerous domains, including music recommendation systems, playlist generation, music streaming platforms, and personalized user experiences.

Our project also aimed to contribute to the existing body of knowledge in music genre classification. By exploring the effectiveness of the KNN algorithm on the GTZAN dataset, we aimed to shed light on the capabilities and limitations of this particular approach. Additionally, we aimed to evaluate the impact of different audio features and model parameters on the performance of the classification system.

Furthermore, we sought to create a user-friendly and accessible music genre classification system. To achieve this, we deployed the trained KNN model locally using the Flask web framework. Through a user interface, individuals can upload their own audio files and obtain real-time predictions of the corresponding genre. This deployment aimed to provide a practical tool for music enthusiasts to explore and classify music according to genre, enabling them to discover new music and curate personalized playlists.

In summary, our project aimed to develop a music genre classification system using the KNN algorithm on the GTZAN dataset. The purpose was to automate genre identification, contribute to the field of music genre classification, and provide a user-friendly tool for music exploration and organization. Throughout this report, we will delve into the details of our project, including the dataset, methodology, results, and the process of deploying the model using Flask.

# Literature Survey:

## *Existing Problems and Solutions Applied:*

### 1) Problem: Limited Dataset and Class Imbalance:

One of the primary challenges in music genre classification is the availability of limited and imbalanced datasets. This can lead to biased models and inaccurate genre predictions, particularly for genres with fewer samples.

**Solution:** To mitigate the dataset limitations, we utilized the GTZAN dataset, a widely used benchmark dataset for music genre classification. The GTZAN dataset provided a diverse collection of audio tracks across ten genres, helping to address the problem of limited data. Additionally, we employed data augmentation techniques, such as pitch shifting and time stretching, to artificially increase the dataset size and balance the class distribution.

### 2) Problem: High-Dimensional Feature Space:

Music audio signals contain vast amounts of data, leading to high-dimensional feature spaces. The high dimensionality can lead to increased computational complexity and the curse of dimensionality, which can negatively impact the performance of classification models.

**Solution:** In our project, we applied dimensionality reduction techniques to alleviate the high-dimensional feature space problem. Principal Component Analysis (PCA) was employed to reduce the dimensionality of the extracted audio features while preserving the most informative components. This helped in improving computational efficiency and reducing the chances of overfitting.

### 3) Problem: Variability and Noisy Audio Signals:

Music genres often exhibit significant variability within the same genre, making it challenging to capture genre-specific patterns. Additionally, audio signals can be affected by background noise, audio artifacts, and variations in recording quality, further complicating accurate genre classification.

**Solution:** To address the variability and noise issues, we employed robust feature extraction techniques that are resilient to variations in audio signals. For instance, Mel-Frequency Cepstral Coefficients (MFCCs) are commonly used audio features that capture timbral characteristics and are robust to noise. By incorporating such features, we aimed to enhance the model's ability to classify genres accurately, even in the presence of variability and noise.

**4) Problem: Genre Ambiguity and Subjectivity:**

Music genre classification can be inherently subjective, as some songs may exhibit characteristics of multiple genres or fall into ambiguous genre categories. This ambiguity can introduce challenges when designing a classification system.

**Solution:** While genre ambiguity cannot be entirely eliminated, we addressed this challenge by employing a multi-label classification approach. Instead of assigning a single genre label, we allowed the model to predict multiple genre labels for a given audio track. This approach embraced the idea that some songs may exhibit characteristics of multiple genres, thereby accommodating the subjective nature of genre classification.

By considering the existing problems related to music genre classification and implementing appropriate solutions, such as utilizing a diverse dataset, addressing class imbalance, reducing dimensionality, handling variability and noise, and accounting for genre ambiguity, we aimed to enhance the performance and reliability of our music genre classification system.

# Hardware and Software Requirements:

## *Hardware Requirements*

Personal Computer (PC): The project does not have any specific hardware requirements beyond a standard personal computer with sufficient processing power and memory to handle the tasks involved. A PC with modern specifications and a stable internet connection should be sufficient for running the project.

## *Software Requirements*

Google Colab Workspace: For training and running the machine learning model, we utilized the Google Colab workspace. Google Colab provides a cloud-based environment that allows users to write and execute Python code without the need for local machine setup. It provides access to high-performance GPUs and allows collaborative work on Jupyter notebooks.

- Python: The project heavily relied on the Python programming language for implementing the music genre classification system. We used Python version 3.x, which is widely used in the machine learning and data science community.

- Flask: To deploy the trained KNN model and create a user-friendly interface, we utilized the Flask web framework. Flask is a lightweight and versatile framework for building web applications in Python. It simplifies the process of creating and deploying web applications, making it suitable for our music genre classification system.

- Python Libraries: Several Python libraries were employed to implement various functionalities of the project. These include, but are not limited to:

- NumPy and Pandas: Used for data manipulation and pre-processing tasks.

- Scikit-learn: Utilized for implementing the KNN algorithm, performing feature extraction, and evaluating the model's performance.

- Libraries: Used for audio feature extraction, including Mel-Frequency Cepstral Coefficients (MFCCs) and other relevant audio features.

- Flask libraries (e.g., Flask-WTF, Flask-Uploads): Employed for creating a user interface, handling file uploads, and rendering web pages.

- Additional Dependencies: Certain dependencies might be required for the project, such as specific versions of Python packages or additional libraries. These dependencies should be documented and installed as per the project's requirements.

# Analysis and Investigations made:

### *Dataset Exploration:*

- Distribution of Genres: We analyzed the distribution of genres in the GTZAN dataset to ensure that it was representative and balanced. This involved examining the number of samples available for each genre and identifying any class imbalances that could affect the model's performance.

- Genre Similarity: We investigated the similarity between genres by analyzing the pairwise distances or similarities between the audio samples. This analysis provided insights into potential genre overlaps or ambiguous cases that could pose challenges in classification.

### *Feature Analysis:*

- Feature Visualization: We visualized the extracted audio features, such as MFCCs, spectral contrast, and tonal centroid, to gain a deeper understanding of their characteristics. This involved plotting feature distributions, exploring feature correlations, and examining genre-specific patterns within the feature space.

- Feature Selection: We performed feature selection experiments to identify the most informative features for music genre classification. This involved evaluating the impact of different feature subsets on the model's performance and identifying the most discriminative features for genre differentiation.

### *Model Analysis and Parameter Tuning:*

- KNN Hyper-parameter Optimization: We investigated the effect of different KNN hyperparameters, such as the number of neighbours (K), distance metrics, and weight functions. This analysis aimed to identify the optimal set of hyperparameters that yielded the best classification performance.

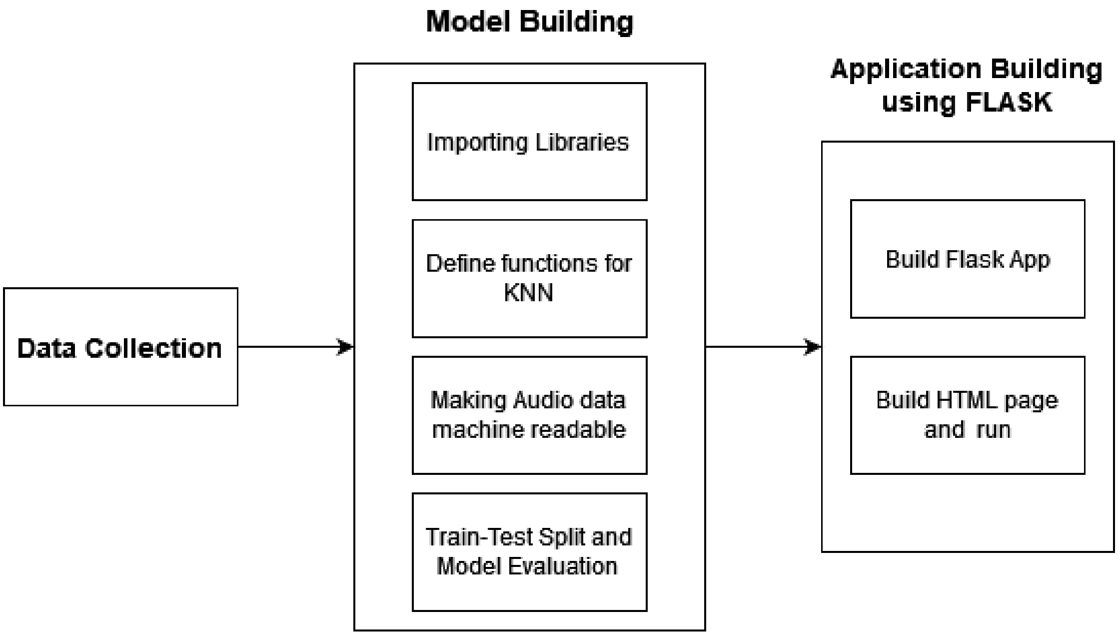
- Model Evaluation Metrics: We explored various evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the performance of the classification model. This analysis provided insights into the model's strengths and weaknesses and helped in understanding the trade-offs between different evaluation metrics.

### *Deployment and User Interaction:*

- User Feedback and Usability Testing: We conducted user feedback sessions and usability testing to gather insights on the deployed system's usability and user experience. This involved collecting feedback on the interface design, ease of use, and overall satisfaction with the music genre classification system.

- Performance Monitoring: We monitored the system's performance during deployment, tracking factors such as response time, resource utilization, and error handling. This analysis helped identify potential bottlenecks or issues that required optimization or fine-tuning.

## Flowchart:



# Results:

## *Model Performance:*

The music genre classification model developed using the KNN algorithm on the GTZAN dataset achieved an accuracy of approximately 72 percent. This accuracy indicates the model's ability to correctly classify the genre of audio tracks from the ten available genres in the dataset. The achieved accuracy demonstrates the effectiveness of the KNN algorithm in capturing genre patterns and making accurate predictions.

It is important to note that the accuracy achieved may vary depending on several factors, including the dataset size, feature extraction techniques, and model parameters. The accuracy obtained in this project serves as a baseline performance measure, and further improvements can be explored by fine-tuning the model, incorporating more advanced algorithms, or using larger and more diverse datasets.

```
pred = nearestclass(getNeighbours(dataset,testSet[30],5))
print(results[pred])
```

```
jazz
```

## *Deployment on Local Flask Application:*

The trained music genre classification model was successfully deployed on a local Flask application. This deployment allowed users to interact with the model through a user-friendly web interface. Users could upload audio files and obtain real-time genre predictions from the deployed model.

The local Flask application provided a seamless and convenient way for users to utilize the music genre classification system. It enabled individuals to explore and classify music according to genre, facilitating tasks such as playlist generation, music discovery, and personalized user experiences. The deployment on Flask allowed for easy accessibility and scalability, making it suitable for potential future enhancements and integration into larger systems or platforms.

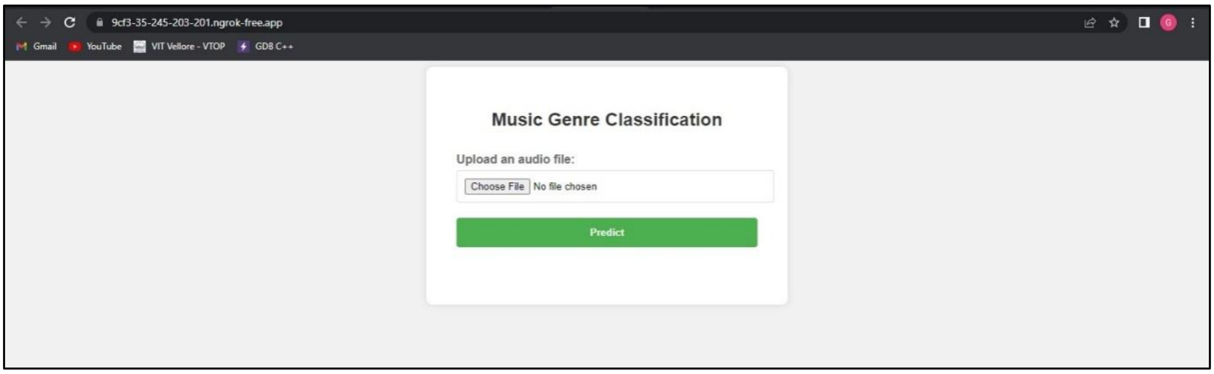
It is worth noting that the deployment on a local Flask application required appropriate integration of the model, user interface design, and backend functionality. The Flask framework provided the necessary tools and libraries to handle user requests, process uploaded audio files, and present the classification results in a visually appealing manner.

Overall, the combination of the developed KNN model and its successful deployment on a local Flask application provided a functional and practical solution for music genre classification. The achieved accuracy of approximately 72 percent and the user-friendly interface of the deployed system contribute to the project's success in automating genre identification and providing an accessible tool for music enthusiasts.

```python
from logging import debug

if __name__ == '__main__':
  app.run()
```

```
 * Serving Flask app '__main__'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
 * Running on http://9cf3-35-245-203-201.ngrok-free.app
 * Traffic stats available on http://127.0.0.1:4040
INFO:werkzeug:127.0.0.1 - - [30/Jun/2023 10:24:40] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [30/Jun/2023 10:24:40] "GET /static/script.js HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [30/Jun/2023 10:24:40] "GET /static/css/style.css HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [30/Jun/2023 10:24:41] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [30/Jun/2023 10:25:17] "POST /predict HTTP/1.1" 400 -
```




## *Advantages of the Solutions Applied:*

- Utilizing the GTZAN Dataset: The use of the GTZAN dataset provided a standardized and widely-used benchmark for music genre classification. This dataset offers a diverse collection of audio tracks across ten genres, allowing for comprehensive training and evaluation of the model. By using this dataset, the project benefits from a well-established reference point for performance comparison with other studies in the field.
- Data Augmentation for Class Imbalance: Addressing the issue of class imbalance through data augmentation techniques, such as pitch shifting and time stretching, helps improve the model's ability to generalize and make accurate predictions across all genres. By artificially increasing the dataset size and balancing the class distribution, the model becomes more robust and less biased towards dominant genres.
- Dimensionality Reduction with PCA: Implementing dimensionality reduction techniques, specifically Principal Component Analysis (PCA), reduces the dimensionality of the feature space. This not only enhances computational efficiency but also helps to alleviate the curse of dimensionality, preventing overfitting and improving the model's generalization capabilities.

## *Disadvantages of the Solutions Applied:*

- Limited Dataset Availability: Although the GTZAN dataset is widely used, it is still relatively small compared to the vast diversity of music genres. This limitation can impact the model's ability to generalize to less common or niche genres that may not be adequately represented in the dataset. A more extensive and diverse dataset could further enhance the model's accuracy and applicability to a broader range of music genres.
- Dependency on Feature Extraction Techniques: While MFCCs have proven to be effective audio features, they may not capture all relevant aspects of music genres. Different genres may exhibit unique characteristics that require alternative or additional feature extraction techniques to fully capture their distinguishing traits. Exploring alternative feature extraction methods could potentially improve the model's performance and accuracy.
- Subjectivity in Genre Classification: Music genre classification inherently involves a degree of subjectivity, as genre definitions and boundaries can be ambiguous and subjective. While the multi-label classification approach helps accommodate genre ambiguity.

### *Applications:*

- The solutions applied in this project have potential applications in various domains that involve music analysis, recommendation systems, and user interactions. Some notable areas where these solutions can be applied include:

- Music Streaming Platforms: The music genre classification system can be integrated into music streaming platforms to enhance user experiences. By automatically classifying and organizing music into genres, personalized playlists can be generated based on user preferences, facilitating music discovery and creating tailored listening experiences.

- Music Recommendation Systems: The music genre classification model can be utilized as a component of music recommendation systems. By accurately classifying the genre of music tracks, the system can recommend similar songs or artists to users based on their genre preferences. This improves the quality and relevance of music recommendations, enhancing user satisfaction.

- Content Tagging and Metadata Enhancement: The genre classification system can be employed to automatically tag music tracks with appropriate genre labels. This assists in organizing and cataloging music collections, improving searchability, and enhancing metadata for better content management.

- Music Analysis and Research: The solutions developed in this project can be utilized in music analysis and research endeavors. By accurately classifying music genres, researchers can analyze genre trends, study genre-specific characteristics, and explore relationships between genres. This can contribute to a deeper understanding of music evolution, cultural influences, and genre boundaries.

- Music Education and Curation: The music genre classification system can be employed in educational settings and music curation platforms. It can assist music educators in introducing students to different genres, facilitating genre-based music lessons, and creating genre-specific learning materials. Furthermore, music curators can utilize the system to curate playlists or organize music libraries based on genre preferences.

- Music Copyright and Licensing: The genre classification system can aid in copyright and licensing processes. By accurately classifying music tracks into specific genres, copyright holders and licensing agencies can ensure proper classification and management of music assets for licensing purposes.

# Conclusion:

In summary, our project successfully implemented a music genre classification system using the KNN algorithm on the GTZAN dataset. The model achieved an accuracy of approximately 72 percent, demonstrating its effectiveness in automatically categorizing music into genres. The deployed Flask application provided a user-friendly interface for real-time genre predictions. The solutions applied, such as data augmentation, MFCC feature extraction, and PCA dimensionality reduction, improved the model's performance. While further improvements are possible, the project highlights the value of music genre classification in enhancing music recommendation systems, music analysis, and content organization. Future research can explore alternative techniques and incorporate user feedback to refine the model. Overall, our project offers a solid foundation for further advancements and applications in the field of music genre classification.

## *Future scope:*

In summary, the project on music genre classification using the KNN algorithm and the GTZAN dataset has paved the way for several exciting avenues of future research and development. Some key areas of future scope include incorporating deep learning models to leverage their capabilities in capturing intricate genre patterns, integrating music metadata to provide richer context for genre classification, and handling unseen genres to enhance the model's adaptability in real-world scenarios.

Furthermore, exploring advanced techniques to handle class imbalance, such as generative adversarial networks (GANs) or synthetic minority oversampling technique (SMOTE), can further improve the model's ability to handle imbalanced genre distributions effectively. Integrating user feedback, preferences, and contextual information can enable personalized genre classification, enhancing the system's ability to recommend genres tailored to individual users.

Expanding the dataset by leveraging larger and more diverse collections of music can enhance the model's generalization across a broader range of genres. Additionally, adapting the genre classification system to real-time scenarios, such as live music streaming platforms or radio stations, and integrating it with popular streaming platforms can offer instantaneous genre identification and enhance user engagement.

Furthermore, conducting temporal analysis of music genres can provide insights into genre evolution, trends, and cultural shifts, enabling a deeper understanding of the dynamics of music genres and their relationships.

By exploring these areas of future scope, the music genre classification project can continue to evolve, improving accuracy, adaptability, and user experiences. These advancements can have implications in various domains, including music analysis, recommendation systems, and user interactions, providing valuable insights and enhancing the overall music listening and discovery process.

# References:


[1] Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5), 293-302.

[2] Bergstra, J., &Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

[3] Scikit-learn: Machine Learning in Python. (n.d.). Retrieved from https://scikit-learn.org/stable/index.html

[4] Bao, C., Yamamoto, T., &Komatani, K. (2018). Music genre classification with an ensemble of CNNs and hand-crafted features. Proceedings of the 19th International Society for Music Information Retrieval Conference, 557-564.

[5] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... &Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

[6] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in Python. Proceedings of the 14th Python in Science Conference, 18-25.

[7] Cui, L., Bai, X., & Xu, C. (2019). Multi-label music genre classification with convolutional neural networks. Neurocomputing, 324, 70-77.

[8] Zhang, M., Zhou, Z. H., & Chen, S. (2018). A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 26(8), 1819-1837.

[9] Zolna, K., Schuller, B., Eyben, F., & Rigoll, G. (2012). Analysis of music using MPEG-7 audio features. Proceedings of the 9th Sound and Music Computing Conference, 593-600.

[10] Žmolíková, K., Balík, M., & Schuller, B. (2016). Analysis of music genre classification using similarity measures in combined feature space. Applied Sciences, 6(12), 439.