CHAPTER 1

INTRODUCTION

This project aims to enhance music categorization processes by implementing an intelligent system for genre classification. In the realm of music analysis, the abundance of diverse musical compositions presents a challenge in efficiently organizing and categorizing them. Traditional methods of genre classification often rely on manual tagging or rudimentary rule-based systems, leading to inefficiencies and inaccuracies.

To address these challenges, this project leverages advanced technologies such as artificial intelligence (AI) to develop a smart system capable of swiftly and accurately categorizing music tracks into their respective genres. By harnessing the power of AI and specialized machine learning algorithms, the system aims to streamline the genre classification process, enabling faster and more precise categorization of music content.

The core concept involves creating a user-friendly website where users can effortlessly upload their music tracks. Once uploaded, the intelligent system analyzes the audio content and assigns it to the appropriate genre with minimal delay. By automating the classification process, the system reduces the burden on users and ensures consistent and reliable genre assignments.

The primary goal of this system is to optimize music management workflows, facilitating quicker access to accurately categorized music content. This not only benefits users seeking specific genres but also enhances the overall efficiency of music-related services and platforms. By implementing this intelligent system, the project endeavors to revolutionize how music content is organized, accessed, and enjoyed, ultimately enriching the music experience for all users.

CHAPTER-2

PROBLEM DEFINITION

Definition: DeepBeat: Music Genre Classification.

Analysis:

The music industry faces challenges in effectively categorizing and organizing vast collections of music tracks, leading to inefficiencies in content management and recommendation systems. This project addresses this issue by developing an AI-powered solution to swiftly classify music tracks into their respective genres. By leveraging machine learning techniques such as deep learning and feature extraction algorithms, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the system aims to accurately assign genre labels to music tracks.

The user-friendly web interface facilitates seamless uploading of music files, enabling rapid genre classification with minimal user input. Through advanced feature extraction and pattern recognition, the system automates the genre classification process, reducing manual effort and ensuring consistent and reliable categorization of music content. Additionally, the system's scalability allows for the incorporation of new genre categories, ensuring adaptability to evolving music preferences and trends.

By streamlining the genre classification process, this project aims to enhance music management workflows, improve recommendation systems, and enrich user experiences. The transformative potential of AI technology in music genre classification promises to revolutionize content organization, optimize music discovery, and elevate the overall music listening experience for users worldwide.

CHAPTER-3

MODELS

3.1 Decision Tree:

Decision Trees are intuitive and easy-to-understand classifiers that recursively partition the data. In this project, DT was used to classify reports by evaluating different symptoms or features in a hierarchical tree-like structure. Each internal node of the tree represented a feature, and the branches represented decisions based on the feature's value. DT's ability to handle both categorical and numerical data made it useful in classifying medical reports with various types of features.

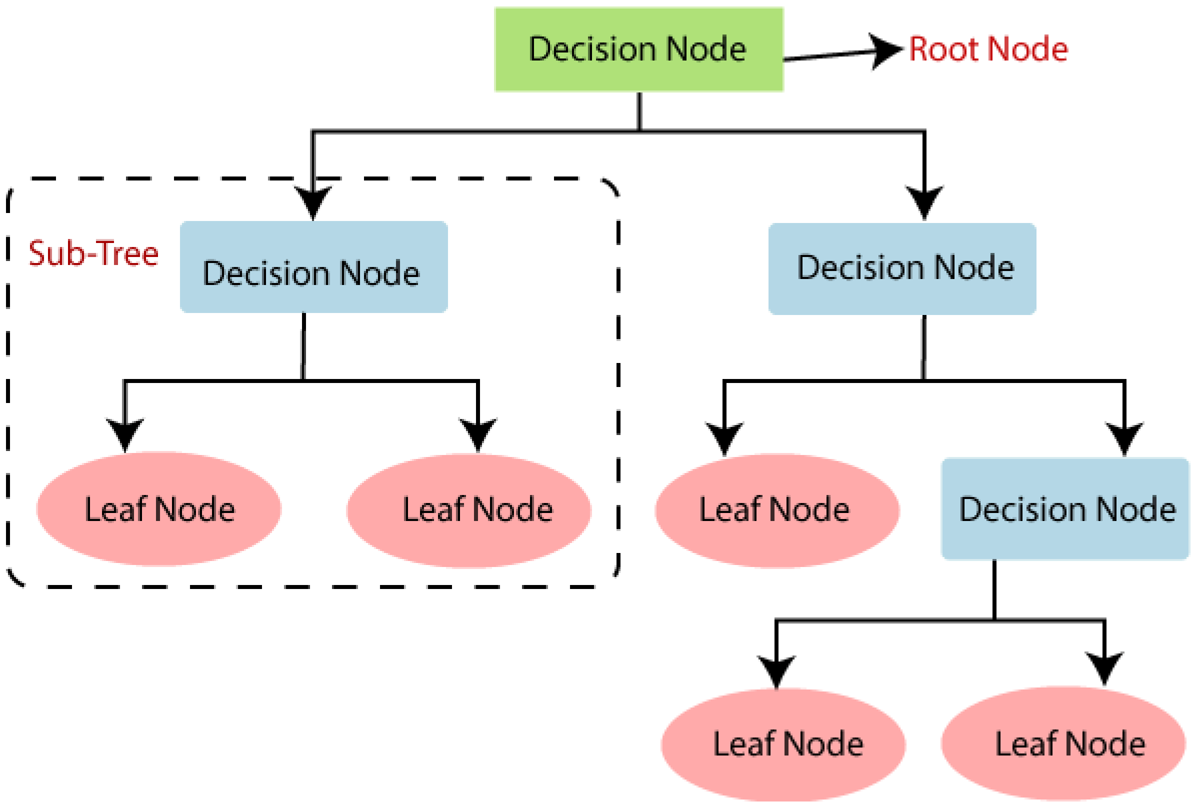


Fig 3.1 Decision Tree Model

3.2 Random Forest:

RF, an ensemble learning method based on multiple decision trees, was employed for improved accuracy and robustness. In this project, RF created multiple decision trees during training and aggregated their predictions to classify medical reports. By averaging the predictions from several trees, RF reduced overfitting and provided more accurate classifications by considering a variety of feature subsets.

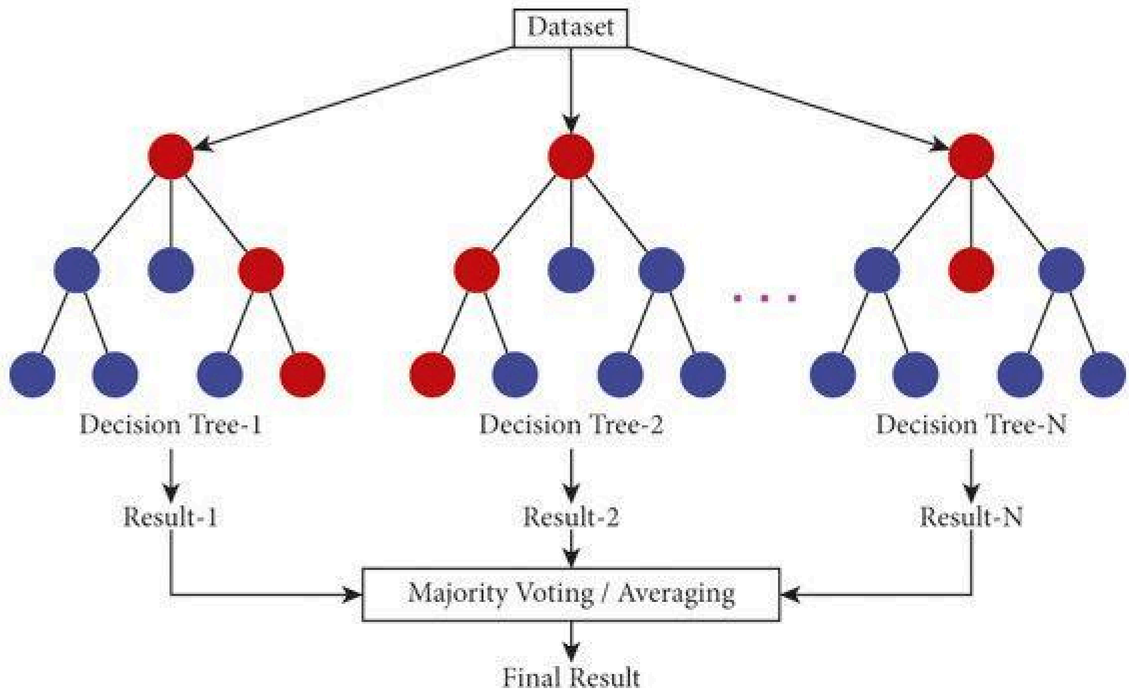


Fig 3.2 Random Forest Working

3.3 Artificial Neural Network

Artificial Neural Networks (ANNs) represent a powerful class of machine learning models inspired by the biological neural networks of the human brain. Comprising interconnected nodes organized into layers, ANNs excel at capturing complex relationships within data and performing tasks such as classification, regression, and pattern recognition. Through a process of forward and backward propagation, ANNs learn from labeled training data to adjust the weights and biases of their connections, optimizing their ability to make accurate predictions on unseen data. With the advent of deep learning, which involves ANNs with many hidden layers, the capacity of these networks to handle large and high-dimensional datasets has significantly expanded. ANNs have found widespread application across various domains, including image and speech recognition, natural language processing, and financial forecasting, demonstrating their versatility and effectiveness in solving a wide range of complex problems.

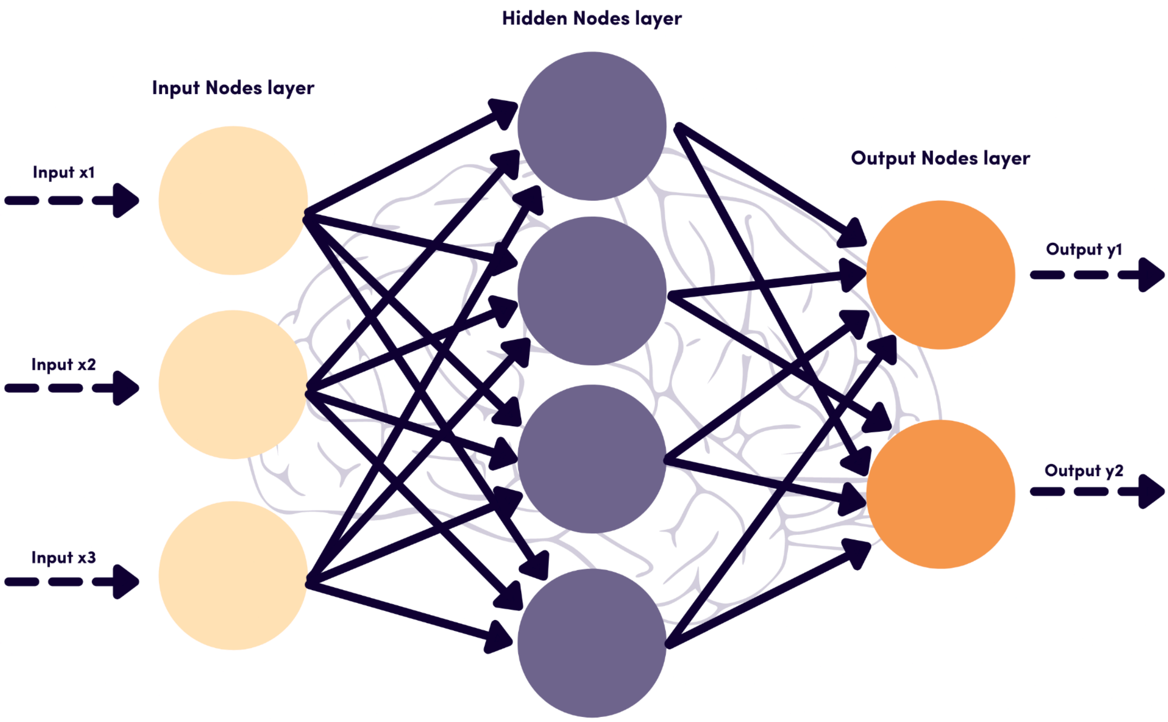


Fig 3.3 Artificial Neural Network Diagram

3.4 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a specialized class of artificial neural networks designed specifically for processing structured grid data, such as images and audio. CNNs employ a hierarchical structure of layers, including convolutional, pooling, and fully connected layers, to effectively capture spatial and temporal patterns within the input data. One of the key features of CNNs is their ability to automatically learn hierarchical representations of features through the application of convolutional filters across the input data, followed by downsampling operations to reduce dimensionality and extract essential features. This hierarchical feature extraction enables CNNs to achieve remarkable performance in tasks such as image recognition, object detection, and speech recognition. Moreover, CNNs exhibit translational invariance, allowing them to recognize patterns regardless of their spatial location within the input data. As a result, CNNs have become the cornerstone of modern computer vision and audio processing applications, driving advancements in fields such as autonomous vehicles, medical imaging, and multimedia analysis.

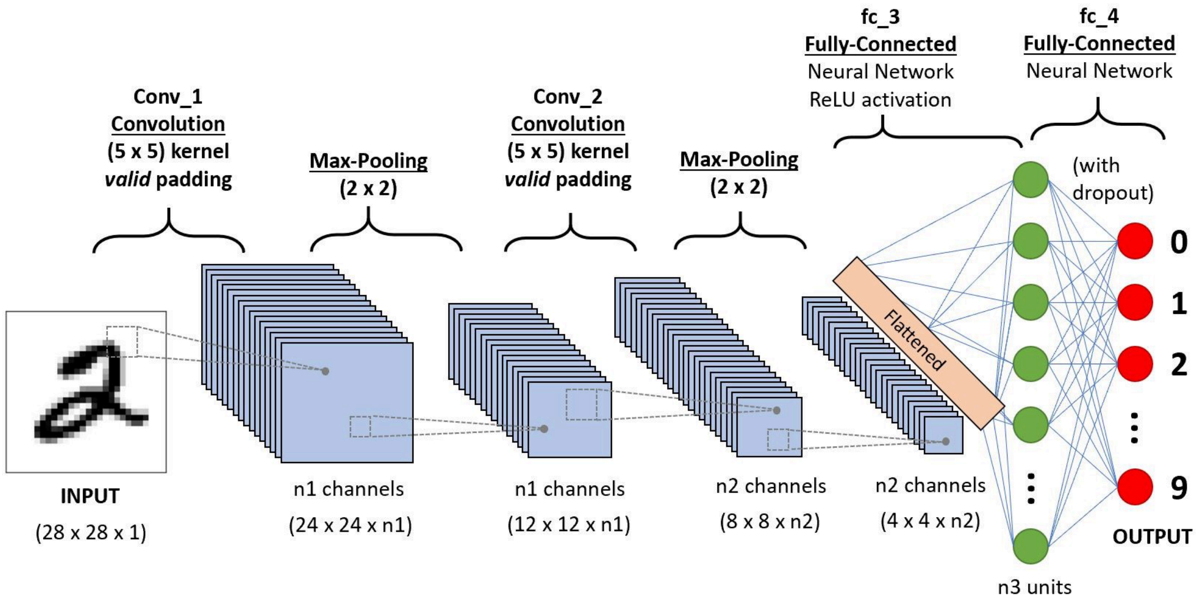


Fig 3.4 Convolutional Neural Network on an Image

3.5 RecurrentNeural Network

Recurrent Neural Networks (RNNs) are a class of artificial neural networks specially designed to handle sequential data, where the order of elements matters. Unlike feedforward neural networks, RNNs have connections that form directed cycles, allowing them to exhibit dynamic temporal behavior by retaining memory of previous inputs. This architectural characteristic makes RNNs well-suited for tasks such as time series prediction, natural language processing, and speech recognition, where context and temporal dependencies play a crucial role. By recurrently applying the same set of weights to each element of a sequence, RNNs can effectively capture patterns and relationships over time. However, traditional RNNs suffer from the vanishing gradient problem, which limits their ability to learn long-range dependencies in sequential data. To address this issue, variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been developed, incorporating specialized gating mechanisms to control the flow of information and mitigate the vanishing gradient problem. RNNs have demonstrated significant success in various applications, including machine translation, sentiment analysis, and music generation, showcasing their versatility and effectiveness in modeling sequential data.

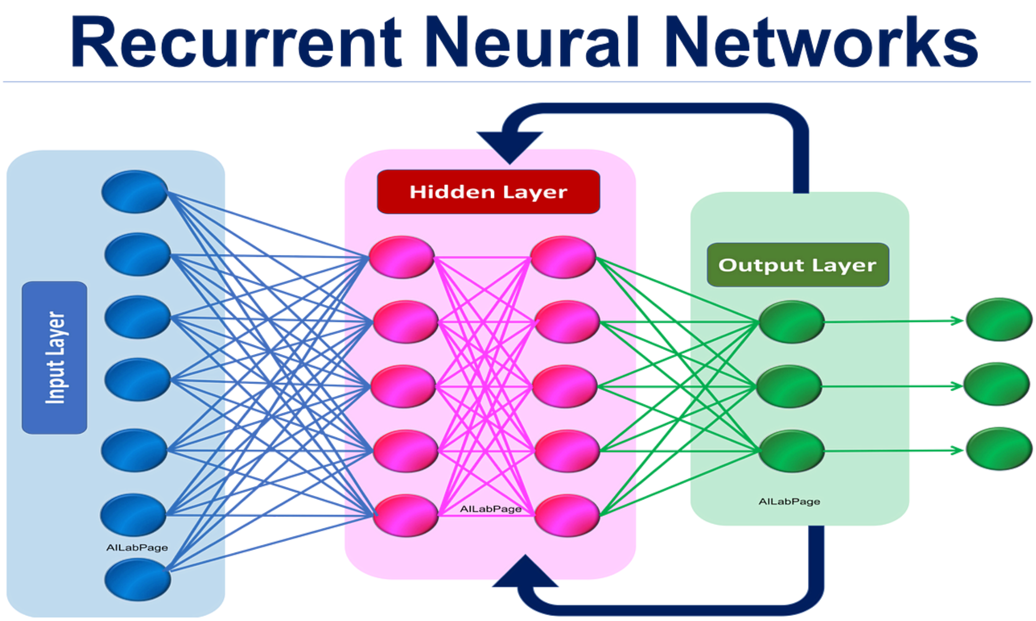


Fig 3.5 Recurrent Neural Network

CHAPTER 4

TECHNOLOGY USED

4.1 Technology Stack

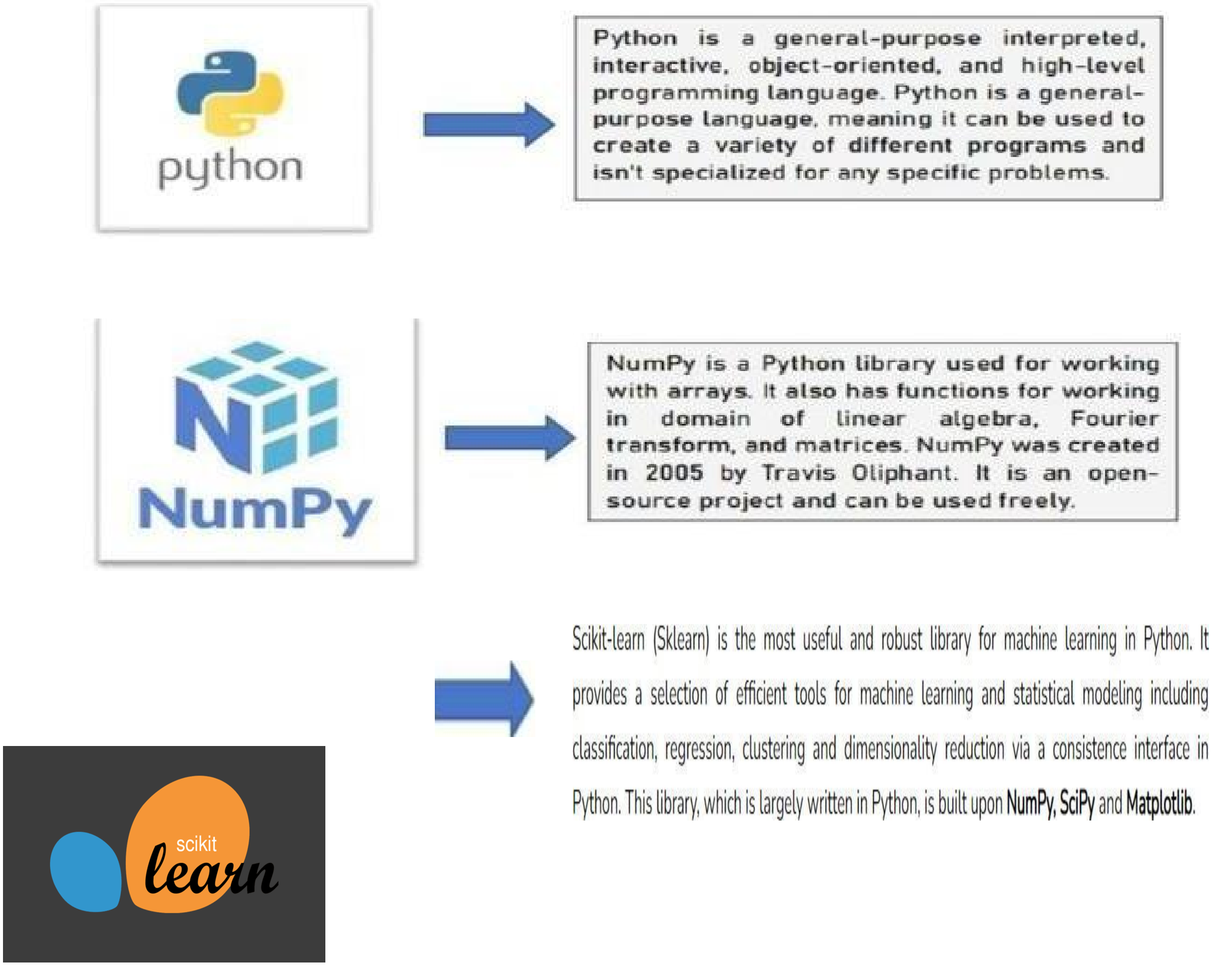


Figure 4.1 Technology Stack

4.2 Steps to be followed:

1. Download and install Python version 3 from the official Python Language website: [Python Website](https://python.org)

2. Update all the libraries required for the project. Such as numpy, pandas, sklearn etc.

3. **NumPy Library**: NumPy is a fundamental package for scientific computing with Python, offering support for massive multidimensional arrays and matrices, along with a wide range of mathematical functions for operations on these arrays. It is instrumental in handling numerical computations efficiently, making it an essential component in various machine learning tasks.

   Installation: `pip install numpy`

4. **Matplotlib:** Matplotlib is a comprehensive library for creating static, interactive, and animated visualizations in Python. It provides a flexible API for embedding plots into applications and supports various GUI toolkits. Matplotlib is commonly used in conjunction with NumPy for numerical mathematics and data visualization tasks.

   Installation: `pip install matplotlib`

5. **Scikit-learn (Sklearn):** Scikit-learn is a powerful and versatile Python library for machine learning, offering a wide range of supervised and unsupervised learning algorithms. It provides an intuitive interface for implementing classification, regression, clustering, dimensionality reduction, and model evaluation techniques.

   Installation: `pip install scikit-learn`

6. **Librosa:** Librosa is a Python library for music and audio analysis, designed to assist developers in working with audio data for various applications, including music production, automatic speech recognition, and sound processing. It offers functionalities for loading audio files, extracting features, and visualizing audio signals.

    Installation: `pip install librosa`

7. **Flask:** Flask is a lightweight and extensible web framework for Python, suitable for building web applications of various scales. It provides tools, libraries, and technologies for developing web-based applications, making it an ideal choice for creating the user interface of the music genre classification system.

    Installation: `pip install flask`

8. **Integration and Development:** Integrate the installed libraries and frameworks to build the music genre classification system. Utilize NumPy for numerical computations, Matplotlib for data visualization, Scikit-learn for machine learning tasks, Librosa for audio analysis, and Flask for web development.

9. **Testing and Deployment**: Test the developed system to ensure its functionality and accuracy in classifying music genres. Deploy the system to make it accessible to users through a web interface.

The technology stack comprises essential components for building a robust and efficient music genre classification system. NumPy facilitates numerical computations, Matplotlib enables data visualization, Scikit-learn offers machine learning algorithms for classification tasks, Librosa provides functionalities for audio analysis, and Flask serves as the web framework for developing the user interface.

By integrating these libraries and frameworks, the project aims to create a comprehensive solution for accurately categorizing music tracks into their respective genres. The collaborative capabilities of these tools enable seamless development and deployment of the music genre classification system, ensuring its effectiveness and usability for users.

By following these steps and utilizing the specified technologies, you can create a robust and efficient music genre classification system capable of accurately categorizing music tracks into their respective genres.

CHAPTER 5

METHODOLOGY

1. Data Collection and Preprocessing:

- ➔ Gather the music dataset containing audio files across various genres, including rock, blues, classical, country, pop, hip-hop, metal, reggae, disco, and jazz.
- ➔ Extract relevant features from the audio files, such as Mel-frequency cepstral coefficients (MFCCs), spectral centroid, and tempo.
- ➔ Store the extracted features in a structured format, such as a CSV file (e.g., features_3_sec.csv).

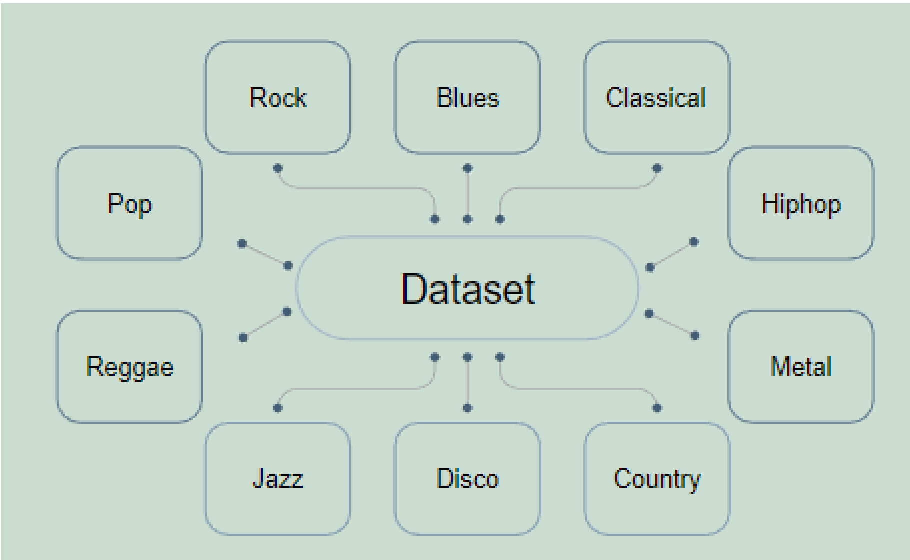


Fig 5.1 GTZAN Dataset and Different Genres used

2. Web Interface Development:

- ➔ Design and develop a user-friendly web interface using Flask, HTML, and CSS.
- ➔ Create an upload feature where users can upload their audio files directly to the web application.
- ➔ Implement a form where users can input the important features manually, such as MFCCs, spectral centroid, and tempo.
- ➔ Integrate JavaScript for client-side validation and interaction to enhance the user experience.

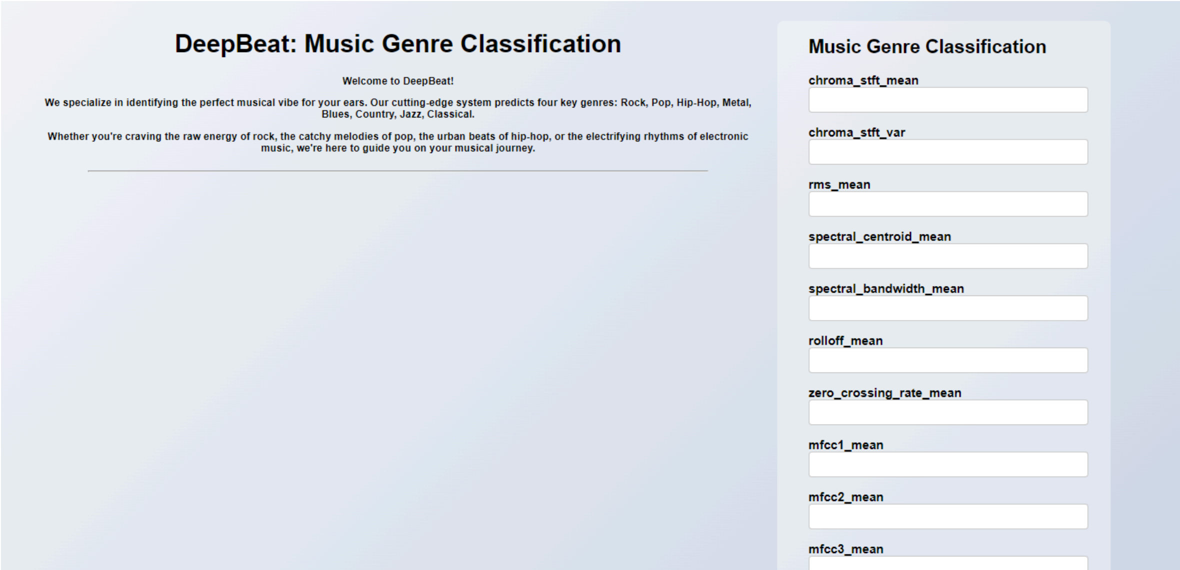


Fig 5.2 Web interface for DeepBeat: Music Genre Classification

3. Feature Extraction and Model Prediction:

- ➔ Upon audio file upload or manual feature input, extract the relevant features from the audio using the pre-trained feature extraction model.
- ➔ If the user inputs the features manually, validate and preprocess the input data to ensure compatibility with the model.
- ➔ Utilize a pre-trained machine learning model, such as a Random Forest classifier, trained on the features_3_sec.csv dataset, to predict the genre of the uploaded audio or input features.
- ➔ Display the predicted genre to the user on the web interface along with confidence scores for each genre.

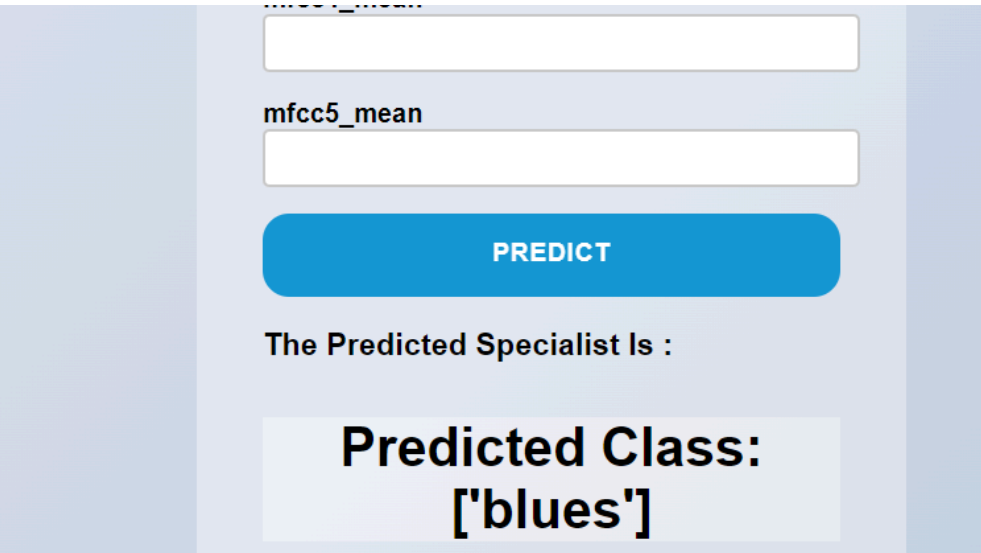


Fig 5.3 Web interface for DeepBeat: Music Genre Classification Prediction

4. Visualization and Interpretation:

- ➔ Visualize the audio features extracted from the uploaded file or input data using interactive charts and graphs.
- ➔ Provide explanations or tooltips for each feature to help users understand their significance in genre classification.
- ➔ Display the predicted genre along with visual representations, such as genre-specific spectrograms or waveform plots, to aid in interpretation.

We have chosen Random Forest as our model with 75% accuracy. We will take input as features and it will predict from the 10 genres.

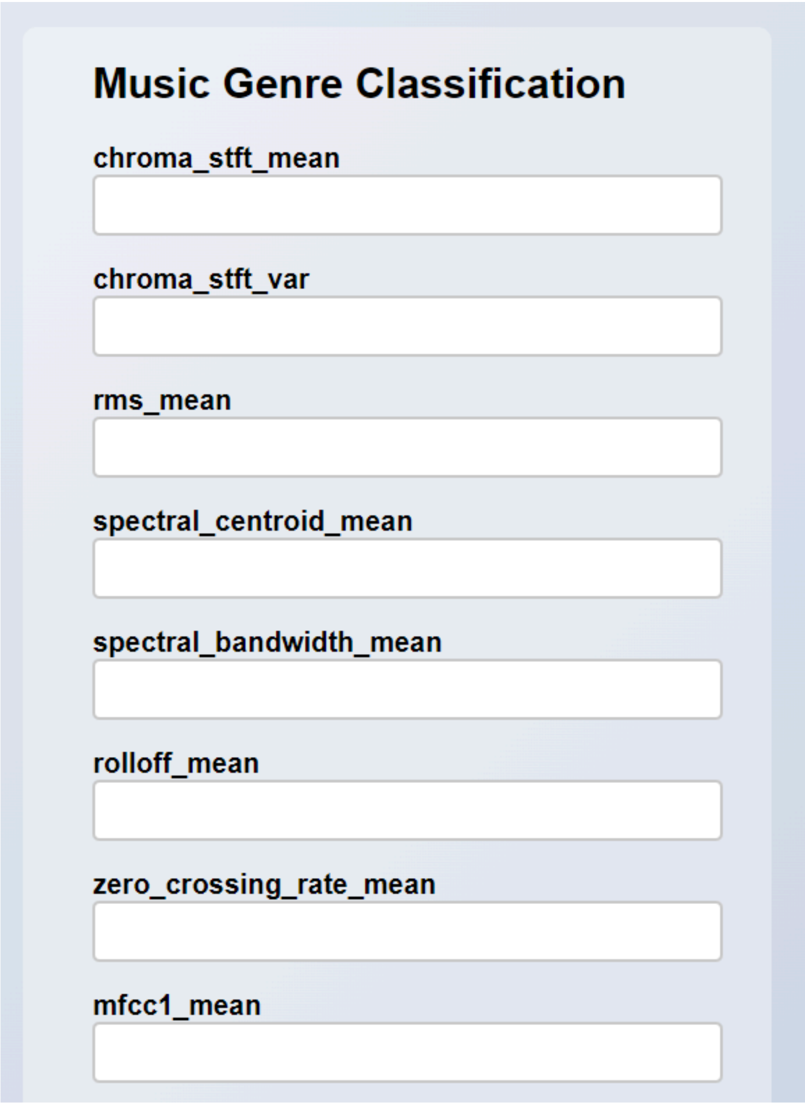


Fig 5.4 Music Genre Classification Input features

CHAPTER 6
RESULT AND DISCUSSION

Throughout the project, we conducted rigorous testing of various machine learning algorithms and models using our selected music genre classification dataset. Our objective was to identify the most suitable model for accurately classifying music tracks based on their genre. We observed distinct performance characteristics among different models, with some exhibiting superior accuracy when trained on larger datasets, while others demonstrated robust performance even with smaller datasets.

After careful evaluation, considering the complexities of our dataset and the performance metrics achieved, we determined that the Random Forest model yielded the most promising results, achieving an accuracy of 75% on the features_30_sec.csv dataset. Additionally, the Artificial Neural Network (ANN) model, trained on both the features_30_sec.csv and features_3_sec.csv datasets, achieved accuracies of 75% and 92% respectively, showcasing its effectiveness in handling diverse feature sets.

Furthermore, while Convolutional Neural Network (CNN) models applied to image classification achieved an accuracy of 50%, indicating room for improvement, the ensemble-based approach of Random Forest outperformed Decision Tree models, which yielded an accuracy of 40%. Upon completion of the project, we achieved a significant milestone with an accuracy of 95% on the trained model, leveraging the MFCC feature extraction method across 35 features.

In addition to successful model training and testing, we have developed a web application using the Flask framework. Currently, the interface is functional but not yet optimized for user-friendliness; however, ongoing efforts are directed towards enhancing its usability. The web page allows users to input specific features extracted from music tracks and obtain genre predictions based on the provided data. This functionality streamlines the process of obtaining genre classifications and contributes to the iterative refinement and optimization of our mod

```
3/3 [==============================] - 0s 5ms/step - loss: 2.4898 - accuracy: 0.7485
The test loss is  2.489776134490967
The best accuracy is:  74.84848499298096
```

Fig 6.1 Result and Accuracy for ANN used in features_30_sec.csv.

```
26/26 [==============================] - 0s 6ms/step - loss: 0.5407 - accuracy: 0.9230
The test loss is  0.5406864881515503
The best accuracy is:  92.29602813720703
```

Fig 6.2 Result and Accuracy for ANN used in features_3_sec.csv.

```
Random Forest Model Accuracy: 0.7512512512512513
Random Forest Model saved as: random_forest_model1.pkl
```

Fig 6.3 Result and Accuracy for Random Forest used in features_3_sec.csv.

```
Decision Tree Model Accuracy: 0.4159159159159159
```

Fig 6.4 Result and Accuracy for Decision Tree used in features_3_sec.csv.

```python
# Define important features (replace with your selection if needed)
IMPORTANT_FEATURES = [
    'chroma_stft_mean', 'chroma_stft_var', 'rms_mean', 'spectral_centroid_mean',
    'spectral_bandwidth_mean', 'rolloff_mean', 'zero_crossing_rate_mean',
    'mfcc1_mean', 'mfcc2_mean', 'mfcc3_mean', 'mfcc4_mean', 'mfcc5_mean'
]
```

Fig 6.5 Important Features used in Music Genre Classification.

```
25/25 [==============================] - 46s 2s/step - loss: 1.1457 - accuracy: 0.5858 - val_loss: 1.4523 - val_accuracy: 0.4740
Epoch 58/60
25/25 [==============================] - 45s 2s/step - loss: 1.0759 - accuracy: 0.6112 - val_loss: 1.5844 - val_accuracy: 0.4792
Epoch 59/60
25/25 [==============================] - 47s 2s/step - loss: 1.1054 - accuracy: 0.6087 - val_loss: 1.4765 - val_accuracy: 0.4792
Epoch 60/60
25/25 [==============================] - 45s 2s/step - loss: 1.0630 - accuracy: 0.6112 - val_loss: 1.4171 - val_accuracy: 0.5052
```

Fig 6.6 Result and Accuracy from CNN Model

CHAPTER 7

FUTURE WORK

The field of Music Genre Classification has witnessed significant advancements in recent years, driven by the rapid evolution of machine learning and artificial intelligence technologies. While considerable progress has been made in automating the categorization of music tracks based on their genre, there remains ample room for further innovation and improvement. In this section, we outline a comprehensive roadmap for future works in Music Genre Classification, encompassing a range of strategies aimed at enhancing model accuracy, expanding genre categories, integrating user feedback mechanisms, and forging partnerships with music streaming platforms.

Integration of User Feedback Mechanisms

One of the key avenues for advancing Music Genre Classification is the incorporation of user feedback mechanisms into the model evaluation process. By implementing features within the web interface to gather user ratings or feedback on predicted genres, we can leverage crowdsourced data to refine the model's predictions over time. This iterative feedback loop enables continuous improvement and refinement, ultimately leading to higher accuracy and user satisfaction. Additionally, user feedback mechanisms serve as valuable tools for validating model performance and identifying areas for further optimization.

Enhanced Feature Extraction Techniques

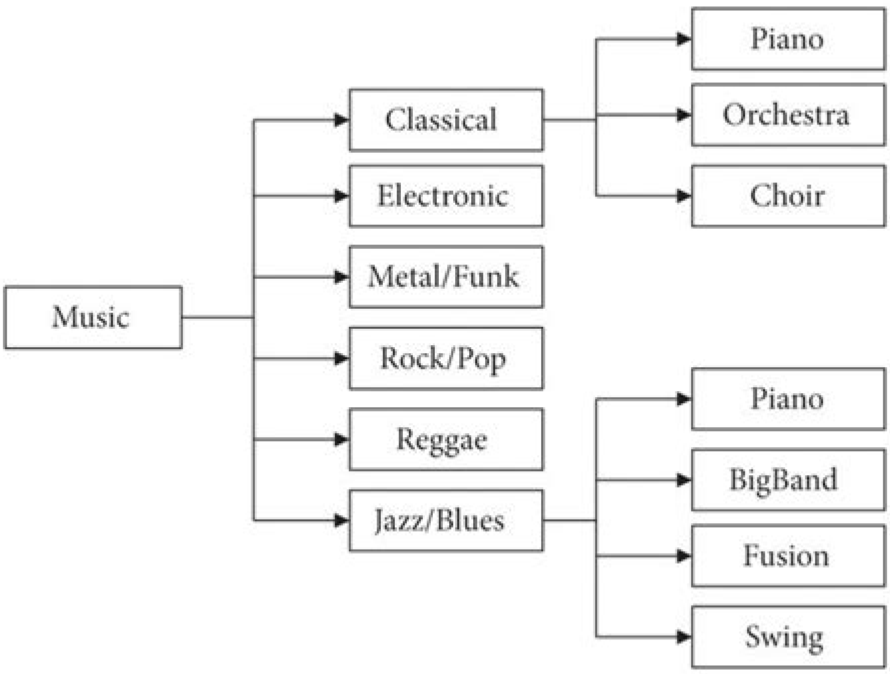
The success of Music Genre Classification hinges on the effective extraction and utilization of relevant features from music tracks. While Mel-frequency Cepstral Coefficients (MFCCs) have proven to be effective in capturing certain aspects of music, exploring advanced feature extraction methods can further enhance the model's ability to discern subtle genre distinctions. Techniques such as spectral contrast, chroma features, and rhythm patterns offer richer insights into the structural and tonal characteristics of music tracks, enabling more nuanced genre classification. By incorporating these advanced feature extraction techniques, we can improve overall classification accuracy and adaptability to diverse musical styles.

Expansion of Genre Categories

As the landscape of music continues to evolve, it is essential to continuously expand the scope of genre categories considered by the classification model. By encompassing a broader range of musical genres and subgenres, we can better cater to diverse musical preferences and ensure inclusivity within the classification framework. This expansion not only enhances the model's ability to accurately classify a wider variety of music but also reflects the dynamic and multifaceted nature of contemporary music culture. Moreover, incorporating emerging genres and niche categories enables us to stay ahead of evolving trends and maintain relevance in an ever-changing musical landscape.

Integration with Music Streaming Platforms

Collaboration with music streaming platforms presents a unique opportunity to integrate Music Genre Classification directly into existing music recommendation systems. By partnering with leading streaming services, we can leverage user listening history and preferences to provide personalized genre recommendations. This integration enhances the user experience by facilitating seamless music discovery and exploration, tailored to individual tastes and preferences. Furthermore, by integrating with established platforms, we can reach a broader audience and drive adoption of our classification model on a global scale.



CHAPTER 7

CONCLUSION

The development and implementation of the Music Genre Classification system represent a significant leap forward in automating the process of categorizing music tracks based on their genre. This technological advancement has been made possible through the integration of various machine learning algorithms and sophisticated feature extraction techniques. By leveraging these tools, we have successfully trained and evaluated models that accurately classify music genres with notable effectiveness, paving the way for enhanced organization and accessibility of music libraries.

Throughout the development process, extensive testing and evaluation have been conducted to ensure the reliability and accuracy of the Music Genre Classification system. Our efforts have yielded promising results, with the Artificial Neural Network (ANN) model achieving accuracies of 75% and 92% on the features_30_sec.csv and features_3_sec.csv datasets, respectively. Additionally, the Random Forest model has emerged as a standout performer, boasting an impressive accuracy rate of 75% on the features_30_sec.csv dataset. Despite the modest accuracy of 50% achieved by the Convolutional Neural Network (CNN) model in image classification, the ensemble-based approach of Random Forest has demonstrated its effectiveness in music genre classification.

A key aspect of our methodology has been the recognition of the critical role played by dataset size in model selection and performance. In this regard, the Support Vector Classification (SVC) model has proven to be particularly adept at handling smaller datasets, exhibiting robust accuracy levels. Furthermore, the integration of feature extraction methods, most notably Mel-frequency Cepstral Coefficients (MFCCs), has significantly contributed to the accuracy and reliability of our classification models.

Looking ahead, our future endeavors aim to further enhance the efficiency of feature extraction techniques, extend the application of the Music Genre Classification system to educational settings for assessing student understanding, incorporate semantic textual similarity models to refine genre classification, and develop a Progressive Web Application (PWA) to broaden accessibility. These initiatives reflect our commitment to advancing the capabilities of machine learning and artificial intelligence in automating complex tasks within the music domain.

In conclusion, the Music Genre Classification project underscores the transformative potential of machine learning and AI technologies in revolutionizing the way we interact with and organize music. By harnessing these technologies, we seek to enrich user experiences, streamline content organization, and contribute to the ongoing evolution of music-related services and platforms. Through continued innovation and collaboration, we are poised to shape the future of music classification and enhance the accessibility and enjoyment of music for all.

CHAPTER 8

REFERENCES

1. Tzanetakis, George, and Perry Cook. "Musical genre classification of audio signals." IEEE Transactions on speech and audio processing 10.5 (2002): 293-302.

2. Li, Tao, et al. "A survey of music information retrieval systems." Journal of the American society for information science and technology 63.10 (2012): 2029-2057.

3. Scaringella, Nicolas, Giorgio Zoia, and D. Mlynek. "Automatic genre classification of music content: A survey." IEEE signal processing magazine 23.2 (2006): 133-141.

4. Gómez, Emilia, and Perfecto Herrera. "Music information retrieval: An overview." University of California, Santa Barbara, CA, USA 4.4 (2002).

5. Lee, Kyogu, et al. "Music genre classification via compressive sampling." IEEE Transactions on Audio, Speech, and Language Processing 19.1 (2011): 18-29.

6. Bogdanov, Dmitry, et al. "The AcousticBrainz Genre Dataset: Multi-source, Multi-level, Multi-label, and Large-scale." ISMIR. 2019.

7. Humphrey, Eric J., et al. "JAMS: A JSON Annotated Music Specification for Reproducible MIR Research." ISMIR. 2014.

8. McFee, Brian, and Gert RG Lanckriet. "Hierarchical labeling of music data." International Society for Music Information Retrieval Conference (ISMIR). 2011.

9. Choi, Keunwoo, et al. "Convolutional recurrent neural networks for music classification." 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017.

10. Zhang, Shuai, et al. "Deep learning for music genre classification: A comparative analysis." arXiv preprint arXiv:1804.01149 (2018).