**Raga Prediction for Indian Classical Music**

**1.INTRODUCTION**

**A.Project Overview:**

Raga prediction for Indian classical music entails the utilization of machine learning models to identify and categorize ragas based on audio inputs. This process necessitates a substantial dataset of audio recordings that preferably contain labeled information about the corresponding ragas. Attributes such as pitch, timbre, rhythm, and dynamics can be extracted from the audio signals and employed as inputs for the machine learning model.

The ultimate goal is to train the model to accurately recognize and predict the unique characteristics of each raga, ensuring it can generalize effectively to unfamiliar recordings. The success of the project relies heavily on the careful selection and engineering of relevant features, in addition to the appropriate design of the model architecture and training algorithms. These elements collectively contribute to achieving high accuracy in raga prediction, facilitating the system's ability to discern and classify different ragas with accuracy and efficiency.

**B.Goals and Objectives for Phase 2 :**

* The project’s second phase, which runs from 2nd June,2023 to 8th June,2023, concentrates on building and testing various ML and DL models.
* The Goals and Objectives of the current phase are mentioned below,
* Model Selection: Select appropriate ML/DL algorithms based on your project requirements, dataset characteristics, and problem statement. Consider algorithms such as linear regression, logistic regression, decision trees, random forests, support vector machines, convolutional neural networks (CNN), recurrent neural networks (RNN), etc.
* Model Development: Implement the selected ML/DL models using your chosen programming language or framework (e.g., Python with scikit-learn, TensorFlow, PyTorch, etc.). Follow these steps:
* Splitting the dataset: Divide your dataset into training, validation, and testing sets to evaluate model performance accurately.
* Training the models: Fit the ML/DL models to the training data using appropriate training algorithms and hyper-parameters. Experiment with different settings to optimize model performance.
* Model tuning: Fine-tune the models by adjusting hyper-parameters (e.g., learning rate, regularization strength, number of layers, etc.) through techniques like grid search or random search.
* Regularization techniques: Apply regularization techniques like dropout, L1/L2 regularization, or early stopping to prevent overfitting and improve generalization.
* Model Evaluation: Assess the performance of your ML/DL models using suitable evaluation metrics. Consider metrics such as accuracy, precision, recall, F1-score, area under the curve (AUC), mean squared error (MSE), etc.
* Evaluate models on validation data: Use the trained models to predict outcomes on the validation dataset and compute the relevant evaluation metrics.
* Compare model performances: Compare the performances of different models and identify the most promising ones based on the chosen metrics. Consider using techniques like cross-validation for more robust evaluation.
* Model Testing and Iteration: Test the selected models on the unseen testing dataset to evaluate their generalization ability. Analyze the results and iterate on your models if necessary:
* Measure performance on testing data: Apply the trained models to the testing dataset and evaluate their performance using the chosen evaluation metrics.
* Iteration and improvement: Analyze the model's strengths, weaknesses, and areas for improvement. Consider adjusting hyper-parameters, exploring different architectures, or revisiting data pre-processing techniques to enhance model performance.

**2.MODEL BUILDING AND TESTING**

**A. Description of ML and DL models explored :**

* We have explored various techniques through which we can create a Raga prediction model. But after lots of considerations and research, we have decided to go ahead with 5 different models for ML model building and two different models for DL model building.
* The ML models are : 1) Logistic Regression 2) KNN classifier 3) SVM classifier 4) Decision Tree classifier 5) Random Forest Classifier

1. Logistic Regression:

Logistic regression is a statistical technique employed for binary classification tasks. It relies on the utilization of the logistic or sigmoid function, which maps input values to probabilities ranging from 0 to 1.

During the training phase of the logistic regression model, a method called maximum likelihood estimation is employed. This involves iteratively adjusting the feature weights in order to maximize the likelihood of the observed data. The primary objective of the optimization process is to identify the optimal set of weights that yield the best fit to the training data, while simultaneously minimizing the error in predicting the class labels.

1. KNN Classifier :

It employs a distance-based approach to predict the outcome of new data points by considering the labels of their closest neighbors in the training dataset. The parameter "k" specifies the number of neighbors to be considered. During prediction, KNN calculates the distances between the new data point and all the training data points, and then selects the K nearest neighbors based on these distances. The class or value of the new data point is determined through majority voting (classification) or averaging (regression) among its K nearest neighbors.

1. SVM Classifier :

Its objective is to identify an optimal hyperplane that effectively separates different classes within the dataset. SVM achieves this by mapping the input data to a higher-dimensional feature space and locating the hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class. It can handle linear and non-linear classification problems through the use of various kernel functions that facilitate the transformation of data into higher dimensions.

1. Decision Tree Classifier :

It constructs a tree-shaped model, with internal nodes representing features or attributes and leaf nodes representing class labels or predicted values. The algorithm divides the data based on feature values, making decisions at each node to minimize impurity or maximize information gain. This recursive process continues until a stopping criterion or maximum depth is reached. Decision trees are renowned for their interpretability and visualizability, accommodating both categorical and numerical features.

1. Random Forest Classifier :

It operates as an ensemble method, leveraging multiple decision trees to generate predictions. In a Random Forest, each tree is trained on a different subset of the data, utilizing a random selection of features. During prediction, the classifier combines the outputs from all the trees to determine the final class label. This ensemble approach enhances accuracy and mitigates overfitting risks associated with a single decision tree. Random Forest demonstrates robustness in handling noisy data and accommodates both categorical and numerical features. It is valued for its ability to capture intricate relationships and provide insights into feature importance.

* The DL models are, 1) CNN Classifier and 2) ANN Classifier.

1. CNN Classifier :

CNNs employ a combination of convolutional layers, pooling layers, and fully connected layers to automatically learn and extract features from input data. Convolutional layers apply filters to identify patterns and spatial relationships in the data. Pooling layers reduce the spatial dimensions and enhance robustness. Fully connected layers connect the extracted features to the final output layer for classification or regression.

1. ANN Classifier :

Artificial Neural Networks (ANNs), also referred to as neural networks, are highly versatile models extensively utilized in machine learning for various purposes. Inspired by the structure and functioning of the human brain, ANNs consist of interconnected neurons organized into layers: an input layer, hidden layers, and an output layer. Each neuron applies mathematical operations to its inputs and transmits the results to subsequent layers. Training ANNs involves adjusting the inter-neuron weights to optimize model performance, often utilizing backpropagation. Backpropagation calculates errors and updates weights by propagating them backward through the network.

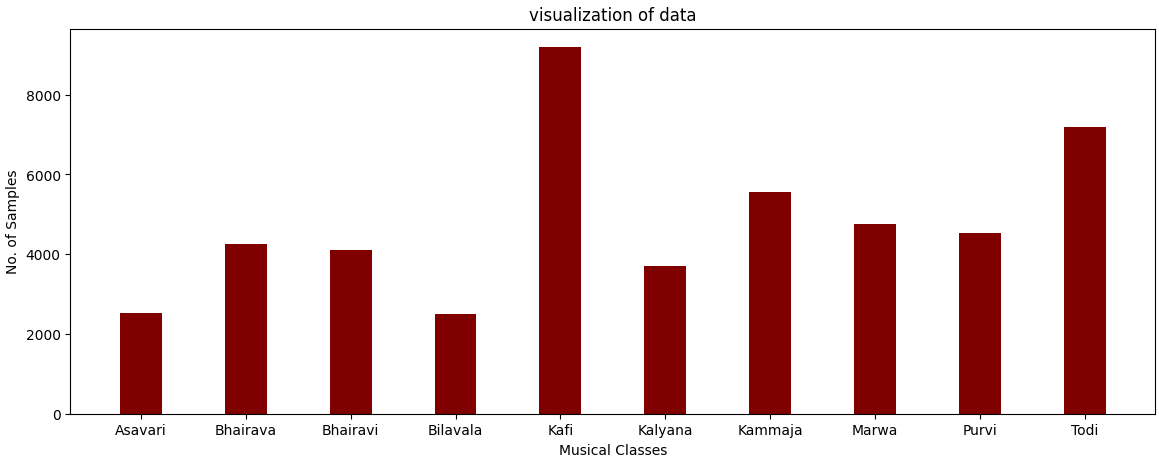
**B. Model Architecture and Design :**

**C. Training Procedure and Techniques :**

* We have

**Summary statistics and Descriptive Analysis**

Bar charts: Bar charts are useful for illustrating categorical factors in respect to the raga class labels, which include ragas. They assist uncover any biases or imbalances by offering information on the distribution and ratios of various categories within the dataset. The bar chart that we used for visualization includes Musical classes in x-axis and number of samples present in y-axis.



**Identification of Outliers or Anomalies**

Data points known as outliers differ greatly from the overall pattern shown in the dataset. They might be the result of mistakes made during data collecting or special cases. Tools for locating outliers include box graphs and scatter plots. It is crucial to evaluate outliers' validity and choose the best handling techniques because they might have a big impact in the following analysis and model performance. On the other hand, anomalies are unusual or unexpected observations that may need more research to fully comprehend and determine whether they have any impact on the raga prediction task.

**Insights and Observations from EDA**

EDA offers useful observations and insights on the dataset, enabling better decision-making in later stages. Researchers can determine the raga classes central tendency as well as their variability through summary statistics, which can help them, chooses the best modeling approaches. A deeper knowledge of the properties of the dataset is provided by descriptive analysis techniques, which highlight trends, patterns, and possible errors within the data.

**Data Engineering**

**Class Creation**

We generated classes using the raga class information provided if the dataset contains explicit class labels for raga types. We specified raga types and assigned class labels in accordance with them. By doing this, we can make sure that our dataset has a balanced representation of all the different Raga classes.

**Handling Unbalanced Data**

We can use strategies to balance the representation of different Raga classes to handle the imbalanced data problem, which occurs when a particular class of Raga has more number of samples than the the other classes. Among the techniques that can be employed are:

1. Under sampling: To make the minority class equal in size, removing samples at random from the majority class

3. Stratified Sampling: Ensuring that the distribution of raga classes is maintained while dividing the data into training and testing sets. This eliminates bias and guarantees that the model is trained on a representative sample.

**Feature Extraction**

**Extracted Features and their Significance to the Project:**

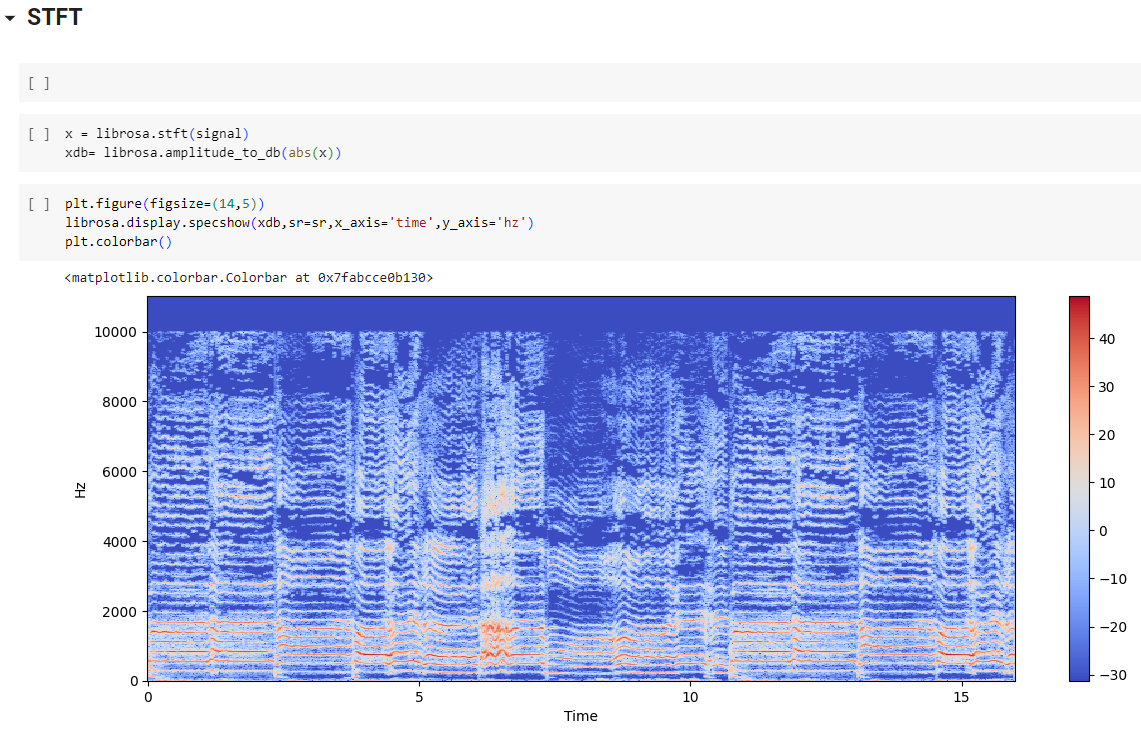
The features that we have selected to be extracted from our audio file dataset and to be used to train the model are, 1) STFT 2) Melspectogram 3) MFCC 4) Chroma Features.

For all of the above mentioned features extraction, we have used the pandas built-in library called “Librosa”.

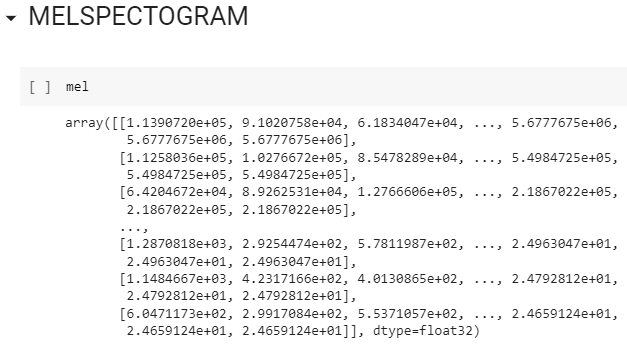
1. The Short-Time Fourier Transform (STFT) is a technique that divides a signal into smaller, overlapping segments and applies the Fourier Transform to each segment. This allows for the examination of how the frequency components of the signal change over time.
2. A Melspectrogram involves first applying the Short-Time Fourier Transform (STFT) to the signal to capture the frequency components over time. Then, a filterbank is utilized to group the frequency bins into mel-frequency bands, which are spaced based on human perception of pitch.
3. MFCCs (Mel-Frequency Cepstral Coefficients) are valuable in capturing significant acoustic characteristics of a signal, such as the vocal tract's shape. The use of MFCCs helps to reduce the dimensionality of the signal while preserving important information, enhancing the effectiveness of audio analysis and machine learning algorithms.
4. Chroma features have practical applications in music genre classification, chord recognition, and melody extraction. They offer a concise representation of the harmonic content in a musical piece and are especially useful in analyzing tonal music.

**Visualizations or Results from Feature Extraction:**

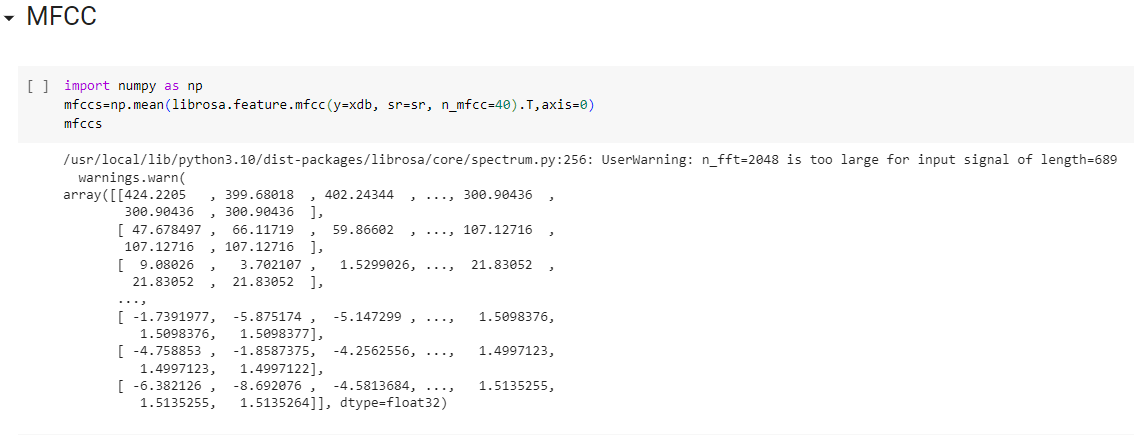
1. STFT:-



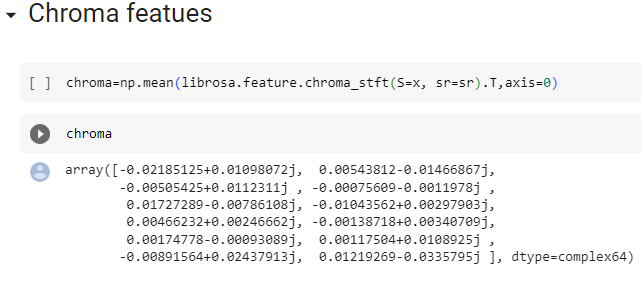
1. Melspectogram:-



1. MFCC :-



1. Chroma Feature :-



**Challenges and Solutions**

**Description of Challenges Encountered and Strategies, Approaches used**

* Predicting the raga (melodic framework) of Indian classical music using machine learning involves numerous challenges during the data pre-processing and feature extraction phases. These challenges are unique to Indian classical music due to its intricate and nuanced nature. Here are some significant challenges faced:
* Data Collection and Annotation: Acquiring a diverse and sizable dataset of Indian classical music recordings with accurate raga annotations can be a daunting task. Raga labeling often requires expertise from musicians or musicologists, which may not always be readily available. Moreover, differences in interpretations by various artists can introduce inconsistencies in the annotations, demanding meticulous handling.
* Signal Pre-processing: Indian classical music recordings frequently contain different types of noise, including ambient sounds, microphone distortions, and audience applause. Effectively removing noise while preserving the integrity of the music and minimizing any artifacts is crucial and requires careful application of noise removal techniques.
* Segmenting and Aligning: Indian classical music compositions comprise distinct sections such as alap, vilambit, and drut, each characterized by unique melodic patterns and tempo. Accurate segmentation and alignment of these sections are essential for precise feature extraction. However, due to the absence of well-defined boundaries or variations in section durations, this task becomes challenging.
* Pitch Estimation and Svara Extraction: Indian classical music relies on a sophisticated system of melodic intervals called svaras. Accurate pitch estimation is pivotal for identifying these svaras, which form the foundation of ragas. However, pitch estimation in Indian classical music is challenging due to intricate melodic ornamentations, gamakas, and microtonal variations, necessitating the use of advanced signal processing techniques.
* Feature Extraction: Extracting pertinent features that capture the essence of ragas from preprocessed audio is crucial for machine learning models. Conventional audio features like MFCCs may not fully capture the melodic characteristics unique to Indian classical music. To overcome this, designing domain-specific features such as spectral features emphasizing tonal centers, note transitions, or svara-based features may be necessary to accurately represent the intricate melodic patterns of ragas.
* Data Imbalance: The distribution of ragas in a dataset often exhibits imbalance, with some ragas having significantly fewer instances than others. This imbalance can introduce bias in machine learning models, favoring ragas with more data during training. To mitigate this challenge, applying appropriate sampling techniques or generating synthetic data can help balance the dataset.
* Generalization across Artists and Styles: Indian classical music showcases considerable variation in rendition style, ornamentation, and improvisation choices across different artists. Machine learning models need to be robust enough to generalize across these variations and make accurate predictions for unseen artists. Cross-validation techniques and carefully designed training strategies can aid in achieving better generalization performance.

**Lessons learnt and Recommendations for future work**

**Lessons learned:**

* Importance of Data Quality: The effectiveness of the raga prediction models is significantly influenced by the quality of the obtained data. Important processes in data pre-processing include ensuring data accuracy, dealing with missing numbers, and handling data inconsistencies.
* EDA as a Foundation: Extensive exploratory data analysis reveals patterns and anomalies in the dataset and provides useful insights. During feature extraction and data pre-processing, it supports in making wise selections.
* Feature Extraction: Using sophisticated feature extraction methods, such CNNs, it is possible to extract valuable information from audio files. The effectiveness of raga prediction models can be considerably impacted by the architectural choice and fine-tuning parameters.
* Iterative Approach: Iterative refinement is frequently needed throughout the data pre-treatment and feature extraction stages. To increase the accuracy and dependability of the models, it is crucial to analyse and enhance the data cleaning and transformation processes.

**Recommendations:**

* Using pre-trained models on huge audio datasets, examine transfer learning methodologies. With little training data, improving these models' performance for raga prediction may be beneficial.
* Investigate how to combine various Raga prediction models using ensemble methods, such as model averaging or stacking. Ensemble methods can improve precision and produce more reliable predictions.
* Consider using appropriate Raga prediction evaluation measures, such as accuracy, to evaluate the effectiveness of the models. These measurements offer a thorough insight of the model's performance and accuracy across various Raga classes.
* Cross-Dataset Evaluation: Conduct cross-dataset analysis to check the raga prediction models' propensity for generalisation. The robustness and performance of the models may be determined by testing them on various datasets.
* Investigate approaches for explaining and interpreting the predictions made by Raga prediction models. The areas of the audio that is most important for raga prediction.
* Future research in human age estimate from photos can significantly increase the accuracy, reliability, and accessibility of the models by taking into account these suggestions and building on the lessons learnt.

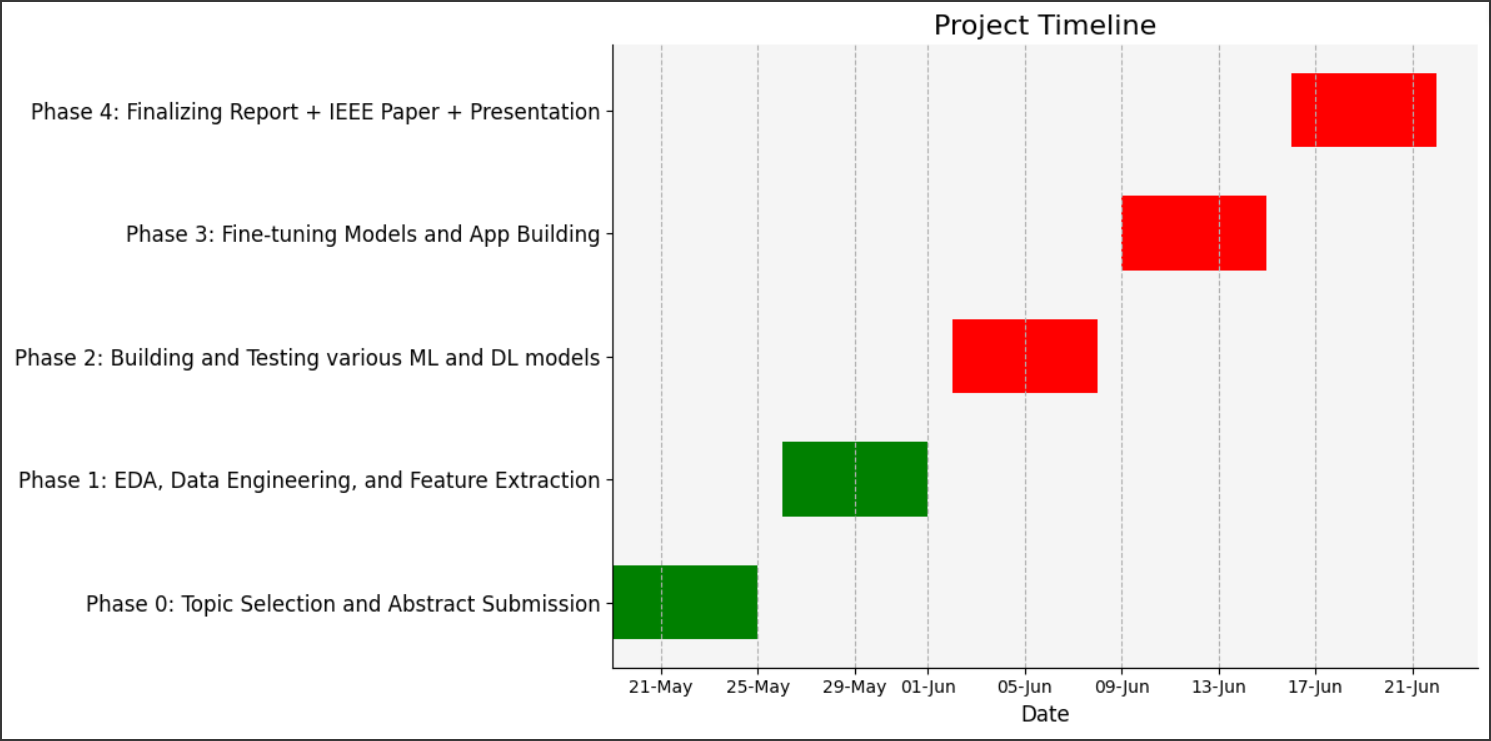
**Timeline and Milestones**

**Review of Project Timeline and Progress**

Phase 0: Topic selection and Abstract Submission [19th MAY 2023 to 25th MAY 2023]

We successfully chose our project topic and submitted the abstract, therefore this part is now marked as complete. It lays the groundwork for the following stages of your project. Our project's initial phase, Phase 0 was primarily concerned with topic selection and abstract submission. This stage is essential since it establishes the framework for the succeeding stages and determines the course of your research. Here are some more thorough explanations of the actions and ideas in Phase 0:

* Topic Selection
* Problem definition
* Abstract writing
* Research Plan
* Ethical Consideration



**Milestones Achieved in Phase 1**

Phase 1: EDA, Data Engineering, and Feature Extraction [26th MAY 2023 to 1st JUNE 2023]

We have been actively involved in completing exploratory data analysis (EDA), data engineering tasks, and feature extraction throughout Phase 1, which is a vital point in our project. By assuring an accurate understanding of the dataset, preparing the data for modelling, and collecting relevant characteristics for raga prediction, this phase establishes the way for next phases. Here is a more thorough overview of what was done during Phase 1:

* EDA: Exploratory Data Analysis
* Data Engineering:
* Feature Extraction

**Planned Milestones for Next Phase**

Phase 2: Building and testing various ML and DL models [2nd June to 8th June 2023]

This phase, which has not yet begun, will concentrate on developing and evaluating several machine learning (ML) and deep learning (DL) models for raga prediction. Model selection, training, hyper parameter tuning, and performance measurement are all part of the process.

* Development of the model
* Model Evaluation
* Model Testing and Iteration

Phase 3: Fine-tuning Models and App Building [9th June to 15th June 2023]

This step, which has not yet begun, will involve perfecting the chosen models, enhancing their functionality, and possibly developing an application for raga prediction based on the trained models. It involves:

* Model Fine-tuning:
* Model Comparison and assessment:
* Model Deployment:

Phase 4: Finalizing Report + IEEE paper + Presentation [16th June to 22nd June 2023]

This final phase has not yet begun. The project report must be completed, an IEEE paper must be written, and a presentation summarising the project's goals, approach, results, and conclusions must be made.

* Project's final report:
* Prepare an IEEE Paper:
* Proofreading and Editing
* Presentation and Submission

**Conclusion**

The project's first phase, which included tasks including data gathering, exploratory data analysis (EDA), data engineering, and feature extraction, was successfully finished. Important accomplishments during this stage include Exploratory Data Analysis (EDA), Data Engineering, and Feature Extraction, which went from May 26 to June 1, 2023.

* The gathering of a wide range of dataset made up of audio files of various raga classes. A starting point for developing and testing the raga predcition models will be provided by this dataset.
* Extensive exploratory data analysis, which revealed information on the features of the audio files, the potential anomalies in the data. The following data pre-processing stages were guided by this analysis.
* Cleaning the dataset, dealing with missing numbers, and assuring data quality are all examples of data engineering tasks. These steps were essential in creating a dependable and consistent dataset for additional research.
* Utilising sophisticated feature extraction methods, including convolutional neural networks (CNNs), to extract useful features from the audio files. The Raga prediction models will take these characteristics as inputs.

By creating a superior dataset and extracting useful features, Phase 1 created an effective foundation for the project. The development of the models, testing, fine-tuning, and app phases are all facilitated by the progress made in this phase. The project is on track and prepared to accomplish its goals of accurately predicting Raga from audios according to the Phase 1's successful conclusion.