**Proposal Format**

|  |  |
| --- | --- |
| **Name** | Selva Vijin Raj A |
| **Email ID** | selvavijinrajaantony.25ec@licet.ac.in |
| **Phone Number** | 9943267833 |
| **GitHub ID** | https://github.com/Selvavijin |
| **Discord ID** | 1127861881269407785 |
| **Current occupation**  *(Working Professionals - add current organization & years of exp)* | Student |
| **Education Details**  *(College Name - Degree Name and branch of engineering or other course/specialization)* | Loyola ICAM College of Engineering and Technology – B.E and ECE |
| **Technical skills with level**  *(Mention tech skills/languages known/UI-UX and level - Novice/Intermediate/Expert)* | HTML,CSS,JS – Intermediate  Python, C – Intermediate  Digital Signal Processing - Intermediate |

**Title: Clustering large amount of audio for Feluda**

**Summary**

The project aims to enhance Feluda, a platform for analyzing multimedia content, with a focus on audio data. The objectives include reviewing literature, researching, and prototyping techniques for grouping similar audio content and visualizing trends. The solution should optimize RAM and CPU usage for scalability and integrate seamlessly into Feluda's architecture. The expected outcome is a CLI or scriptable interface for analyzing social media data, with configurable data sources and processing pipelines. The solution should output structured data for downstream use cases and undergo regular interactive demos and testing.

**Project Detail**

1. **Project Overview:**

1. **Understanding of the project**

The project revolves around enhancing Feluda, a platform designed for analyzing large quantities of multimedia content, particularly focusing on audio data prevalent in Indian social media. The overarching goal is to develop an end-to-end workflow capable of efficiently processing and analyzing this audio data to identify trends and patterns.

To achieve this, the project is divided into several key components:

Literature Review and Research: The team will conduct a comprehensive review of existing literature and research in the fields of machine learning (ML) and digital signal processing (DSP) to explore the state-of-the-art techniques for grouping similar audio content and visualizing trends.

Prototyping: Prototyping involves experimenting with different ML and DSP techniques to determine the most effective approaches for grouping and analyzing audio data. These prototypes will be crucial in selecting the optimal methods for implementation.

Optimization: An essential aspect of the project is optimizing the solution to ensure consistent RAM and CPU usage, particularly important for scalability as the platform aims to process millions of videos. This optimization aims to minimize resource spikes caused by variables such as file size and video length.

Integration with Feluda: The developed solution will be seamlessly integrated into the Feluda platform by creating an operator that adheres to Feluda's interface standards. This integration ensures that the new functionality aligns with Feluda's overarching objectives and architecture.

Configurability and Scriptability: The solution will be designed to be highly configurable via scripts and config files, allowing users to specify data sources (e.g., S3 bucket or database) and customize the data processing pipeline. This flexibility enables users to tailor the analysis to their specific needs.

Output Structuring: The output of the analysis will be structured data that can be easily passed onto a UI service (web or mobile) for downstream use cases. This structured data ensures compatibility with various visualization and reporting tools, facilitating further analysis and interpretation.

Testing and Demos: Regular interactive demos will be conducted with the team using public Jupyter notebooks pushed to the experiments repository. Additionally, the Feluda operator will undergo comprehensive testing to ensure its reliability and functionality, including the ability to run as an independent worker in the cloud to schedule processing jobs over large datasets.

1. **Solutions**

The possible solution for the project involves leveraging a combination of machine learning (ML) techniques for audio analysis and distributed computing for scalability. Here's a high-level overview of the solution:

Data Source Configuration: Users can specify the data source, which could be a combination of an S3 bucket and a SQL database containing file IDs or metadata. Additional sources such as streaming APIs can also be integrated if needed.

Data Processing Pipeline: The data processing pipeline consists of the following steps:

Data Ingestion: Audio files are ingested from the specified data sources.

Feature Extraction: Audio features such as MFCC (Mel-frequency cepstral coefficients), spectrograms, or embeddings are extracted from the audio files using pre-trained deep learning models or traditional signal processing techniques.

Clustering: The extracted features are clustered using algorithms like K-means or hierarchical clustering to group similar audio content together.

Trend Analysis: Temporal trends within the clustered audio groups are analyzed to identify patterns and anomalies over time.

Optimization for Scalability: To optimize RAM and CPU usage, the solution can be implemented using distributed computing frameworks such as Apache Spark or Dask. This allows the processing workload to be distributed across multiple nodes, reducing resource spikes and enabling scalability for processing large datasets.

Integration with Feluda: The solution is integrated into Feluda by creating a custom operator that adheres to Feluda's interface standards. This operator encapsulates the data processing pipeline and can be seamlessly invoked within Feluda's workflow.

Output Storage and Structuring: The results of the analysis, including clustered audio groups and temporal trends, are stored in Elasticsearch or a similar database for easy retrieval and querying. The structured data can be passed onto a UI service for visualization and further analysis.

Configurability and Scriptability: Users can configure the data processing pipeline and parameters through scripts and configuration files, allowing for easy customization based on specific use cases and preferences.

Testing and Monitoring: The solution undergoes rigorous testing to ensure reliability and functionality. Continuous monitoring of resource usage and performance metrics helps identify and address any bottlenecks or issues.

1. **Implementation Details with timelines: Share the milestones you will achieve along with timelines (in terms of weeks) for doing this project.**

1. **Milestone 1**

Week 1-2: Conduct a comprehensive literature review on audio content analysis techniques, focusing on methods for grouping similar audio content and visualizing trends.

Week 3: Research and experiment with different algorithms and approaches for audio content analysis, including clustering, audio fingerprinting, and deep learning-based methods.

Week 4: Prototype the most promising techniques to assess their effectiveness and feasibility within Feluda's architecture.

1. **Milestone 2**

Week 5-6: Optimize RAM and CPU usage of the prototype solutions. Explore methods for efficient data storage, processing, and memory management.

Week 7: Design and implement seamless integration of the audio analysis capabilities into Feluda's existing architecture. Develop APIs or interfaces for communication between components.

Week 8: Conduct initial testing to ensure that the integrated solution meets performance and scalability requirements.

1. **Milestone 3**

Week 9-10: Develop a command-line interface (CLI) or scriptable interface for interacting with the audio analysis functionality. Allow users to specify data sources, configure processing pipelines, and customize analysis parameters.

Week 11: Design structured output formats (e.g., JSON, CSV) for the analysis results to facilitate downstream use cases.

Week 12: Conduct regular interactive demos and testing to gather feedback and ensure that the solution meets user requirements. Iterate on the design based on user input and real-world usage scenarios.

**Availability**

The duration of the coding period is from June to September. Please share your availability in detail

|  |  |
| --- | --- |
| Number of hours available to dedicate to this project per week | 10 hours |
| Do you have any other engagements during this period ? (projects/internships) | No |

**Personal Information**

**About Me:**

I'm a dedicated student interested in Engineering with a passion for leveraging technology to solve complex problems. I have a beginner level background in multimedia processing and data analysis, which I believe makes me well-suited for this project.

**What is your motivation to apply for this project? Answer briefly in 3-4 lines.**

I'm excited about the opportunity to contribute to enhancing Feluda's capabilities in analyzing multimedia content, particularly focusing on audio data. My motivation stems from the challenge of optimizing performance and scalability while integrating seamlessly into Feluda's architecture. I'm eager to apply my expertise to develop a solution that delivers actionable insights from social media data for various downstream use cases.

**Contribution in C4GT’s open community:**

In this section answer the questions about your participation in C4GT’s open community tickets,

provide the screenshot of the leaderboard with your GitHub ID and DPG points earned (if any).

|  |  |
| --- | --- |
| Have you contributed to tickets in C4GT’s open community? **(Mandatory to answer)** | No |
| Have you successfully completed C4GT’s GitHub Classroom Assignment? **(Mandatory to answer)** | No |
| Enter your DPG points **(Mandatory to answer, Enter 0 if not applicable)** | 0 |
| Screenshot of leaderboard with your GitHub ID **(Mandatory to answer, enter 0 if not applicable)** | 0 |