**-­­A Project Report**

**On**

**Sentiment Prediction using Speech**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT AWARD OF THE

DEGREE OF

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**Submitted by**

Shivaji Saxena (1900290110101)

Shailendra Pratap Singh (1900290110096)

Shalini Tyagi(1900290110097)

Abhinetra Patel(1900290210009)

**Under supervision of**

Prof. Youddha Beer Singh



**KIET Group of Institutions, Ghaziabad**

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

**May, 2022**

**DECLARATION**

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name:- Shivaji Saxena (1900290110101)

Shailendra Pratap Singh (1900290110096)

Shalini Tyagi(1900290110097)

Abhinetra Patel(1900290210009)

Date:- 14/03/2023

## CERTIFICATE

This is to certify that Project Report entitled “**Sentiment Prediction using Speech**” which is submitted by **Shivaji Saxena, Shailendra Pratap Singh , Shalini Tyagi and Abhinetra Patel** in partial fulfillment of the requirement for the award ofdegree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

**Date: 14/03/2023 Supervisor**

Prof. Youddha Beer Singh

(Designation)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to **Prof. Youddha Beer Singh** , Department of Computer Science ,KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of **Mr Abhinav Juneja** , Head of the Department of Computer Science and Information Technology , KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Date :

Name : Shivaji Saxena [1900290110101] Name: AbhinetraPatel(1900290210009)

Signature: Signature:

Name: Shailendra Pratap Singh[1900290110096]

Signature:

Name : Shalini Tyagi[1900290110097]

Signature:

**ABSTRACT**

The project **'Sentiment Analysis using Voice**' aims to develop a model for identifying fraudulent calls that scammers make to less-educated people, posing as officials from an agency and asking for personal or bank details. The project's objective is to minimize the impact of noise and background disturbances that might exist in audio data by using a viable solution. The model could also be useful in detecting negative emotions and harmful thoughts, provided additional features and more research.

The project will use existing datasets to test the models, followed by testing on real data, and ultimately, deployment of the model using an appropriate interface. Real-time analysis of calls will help detect fraudulent or valid calls on time. The project will start with a learning process to understand sound and its features in machine learning from different sources, sound data preprocessing, and building the model. The final step will involve testing and deploying the model on real voice data provided by one of the team members.

The project's significance lies in its ability to detect fraudulent calls in real-time, which can help protect less-educated people from getting scammed by fraudsters. Additionally, the project could have potential applications in detecting negative emotions and harmful thoughts.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| DECLARATION……………………………………………………………………. | ii |
| CERTIFICATE……………………………………………………………………… | iii |
| ACKNOWLEDGEMENTS…………………………………………………………. | iv |
| ABSTRACT………………………………………………………………………..... | v |
| LIST OF FIGURES…………………………………………………………………. | ix |
| LIST OF TABLES…………………………………………………………………… | xi |
| LIST OF ABBREVIATIONS………………………………………………………. | xii |
| CHAPTER 1 (INTRODUCTION)…………………………………………………. | 1 |
| 1.1. Introduction……………………………………………………………………... | 1 |
| 1.2. Project Description……………………………………………………………… | 3 |
| CHAPTER 2 (LITERATURE RIVIEW)…………………………………………. | 4 |
| CHAPTER 3 (HIDDEN MARKOV MODEL)................................................. | 8 |
| CHAPTER 4 (TECHNOLOGY USED)................................................................. | 10 |
| CHAPTER 5 (PROBLEM STATEMENT AND EXISTING SYSTEMS)..……… | 20 |
| CHAPTER 6 (OUR PROPOSAL)………............................................................ | 22 |
| CHAPTER 7 (STEPS)……………..………………………....................................  CHAPTER 8(ACCURACY,SCOPE AND APPLICATIONS)…………………….  CHAPTER 9(CONCLUSION)…………………………………………………… | 40  42  45 |

|  |  |
| --- | --- |
| REFERENCES (Only in IEEE Style) | 47 |
| Research Paper |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |

**LIST OF ABBREVIATIONS**

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

The ability to accurately detect and analyze emotions expressed in speech has significant practical applications in a variety of fields, including marketing, customer service, healthcare, and politics. In many cases, individuals express their opinions and emotions about various topics such as products, movies, social issues, and politics over audio calls. The detection of sentiment in such calls can serve two important functions - it can aid in retrieving specific data and increasing its usefulness and help establish the general sentiment of a large number of calls on a similar topic.

While automatic sentiment detection using text has been a well-researched area, there has been relatively less work done on dual sentiment detection in calls based on audio and text analysis. In this project, we focus our attention on analyzing emotions expressed in audio calls, which poses a significant challenge due to the natural and spontaneous nature of speech in such calls.

Our goal is to develop a model that can accurately predict emotions expressed in audio calls. This will involve analyzing both the acoustic features of speech, such as pitch, tone, and intonation, as well as the content of the speech, including the words used, the syntax, and the context in which they are used. We aim to provide a comprehensive analysis of the sentiment expressed in audio calls, which can provide valuable insights for businesses, researchers, and policymakers.

This report presents the methodology, results, and analysis of our project on predicting emotions using speech. We first provide an overview of the existing literature in this area, followed by a description of the data collection process and the pre-processing steps used. We then outline the various feature extraction techniques and machine learning algorithms used in our model, and present the results of our experiments, including an evaluation of the performance of the model. Finally, we discuss the implications of our findings and highlight future directions for research in this area.

**1.2 PROJECT DESCRIPTION**

The project 'Prediction of Emotions using Speech' aims to develop a model for accurately predicting emotions expressed in audio calls. The detection of sentiment in audio calls can provide valuable insights into various topics, such as products, movies, social issues, and politics, and can aid in establishing the general sentiment expressed by a large number of calls on a similar topic. The project focuses on dual sentiment detection in calls based on audio and text analysis.

The project involves analyzing the acoustic features of speech, such as pitch, tone, and intonation, as well as the content of the speech, including the words used, the syntax, and the context in which they are used. The model will be developed using various feature extraction techniques and machine learning algorithms, with the aim of providing a comprehensive analysis of the sentiment expressed in audio calls.

The project will begin with a review of the existing literature in this area, followed by data collection and pre-processing steps. The data collected will be used to train and test the model, and the performance of the model will be evaluated. The project will conclude with a discussion of the implications of the findings and future directions for research in this area.

The project's significance lies in its potential to provide valuable insights into the emotions expressed in audio calls, which can have practical applications in various fields such as marketing, customer service, healthcare, and politics. The development of an accurate and reliable model for predicting emotions expressed in audio calls can provide researchers, businesses, and policymakers with a powerful tool for analyzing the sentiment expressed by individuals and groups on various topics.

**CHAPTER 2**

**LITERATURE REVIEW**

Emotional speech recognition (ESR) has received considerable attention in recent years due to its potential applications in various fields, such as human-computer interaction, automatic speech recognition, and affective computing. In this literature survey, **we explore four different** **research papers** that have proposed innovative techniques for ESR and sentiment analysis.

**The paper "A Study of Support Vector Machines for Emotional Speech Recognition" by Han and colleagues [1]** compares the efficiency of Support Vector Machines (SVM) and Binary Support Vector Machines (BSVM) techniques in utterance-based emotion recognition. The authors used acoustic features such as energy, Mel-Frequency Cepstral coefficients (MFCC), Perceptual Linear Predictive (PLP), Filter Bank (FBANK), pitch, their first and second derivatives as frame-based features. The study found that SVM outperformed BSVM in recognizing six basic emotions with an accuracy of 75.35%.

**In the paper "Audio and Text based multimodal sentiment analysis using features extracted from selective regions and deep neural networks" by Yadav and colleagues [2],** an improved multimodal approach is proposed to detect the sentiment of products based on their audio and text features. The study aimed to classify the input data as either positive or negative sentiment. The authors used Deep Neural Networks (DNNs) to encode each utterance into a fixed-length vector by pooling the activations of the last hidden layer over time. The proposed method achieved promising results with an accuracy of 89.38%.

**The paper "Towards Real-time Speech Emotion Recognition using Deep Neural Networks" by Mirsamadi and colleagues [3] proposes a real-time ESR system based on end-to-end deep learning.** The study presents a DNN that recognizes emotions from a one-second frame of raw speech spectrograms. The proposed method achieved promising results on two databases, namely, the ENTERFACE database and the Surrey Audio-Visual Expressed Emotion (SAVEE) database. The study showed that it is achievable due to a deep hierarchical architecture, data augmentation, and sensible regularization.

**Finally, in the paper "Sentiment extraction from natural audio streams," by Poria and colleagues [4]**, a system for automatic sentiment detection in natural audio streams, such as those found in YouTube, is proposed. The proposed technique uses POS (part of speech) tagging and Maximum Entropy modelling (ME) to develop a text-based sentiment detection model. Additionally, the authors propose an attuning technique that dramatically reduces the number of model parameters in ME while retaining classification capability. The proposed system estimates the sentiment in the YouTube video using decoded ASR (automatic speech recognition) transcripts and the ME sentiment model. The study obtained encouraging classification accuracy despite the challenging nature of the data, such as poor WER (word error rates).

Overall, these four research papers show promising results in the field of ESR and sentiment analysis**. SVM, DNNs, and ME models are among the popular techniques used to extract features and classify emotions and sentiments.** These techniques can be used to build robust and efficient systems for affective computing, speech recognition, and human-computer interaction.

**REFERENCES**

1**.**Sentiment Analysis of speaker specific speech data

Author - Maghilnan S, Rajesh Kumar M, Senior IEEE, Member School of Electronic Engineering VIT University Tamil Nadu, India

[(PDF) Sentiment Analysis on Speaker Specific Speech Data (researchgate.net)](https://www.researchgate.net/publication/323276680_Sentiment_Analysis_on_Speaker_Specific_Speech_Data)

2. Sentiment Analysis of Speech

Author : Aishwarya Murarka , Kajal Shivarkar , Sneha, Vani Gupta ,Prof.Lata Sankpal Student, Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, India

[IJARCCE 37.pdf](https://ijarcce.com/upload/2017/november-17/IJARCCE%2037.pdf)

3.Real-Time Speech Emotion Recognition Using a Pre-trained Image ClassificationNetwork: Effects of Bandwidth Reduction and Companding

Author : Lech M, Stolar M, Best C and Bolia R (2020) Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding. Front. Comput. Sci. 2:14. doi: 10.3389/fcomp.2020.00014

[Frontiers | Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding (frontiersin.org)](https://www.frontiersin.org/articles/10.3389/fcomp.2020.00014/full)

4.Pre-trained Deep Convolution Neural Network Model With Attention for SpeechEmotion Recognition

Author : Zhang H, Gou R, Shang J, Shen F, Wu Y and Dai G (2021) Pre-trained Deep Convolution Neural Network Model With Attention for Speech Emotion Recognition. Front. Physiol. 12:643202. doi: 10.3389/fphys.2021.643202

[Frontiers | Pre-trained Deep Convolution Neural Network Model With Attention for Speech Emotion Recognition (frontiersin.org)](https://www.frontiersin.org/articles/10.3389/fphys.2021.643202/full)

5. Stacking machine learning models for speech sentiment analysis

Author : Zadeh A, Liang PP, Poria S, Vij P, Cambria E and Morency L-P (2018), Multi-attention recurrent network for human communication comprehension, In Thirty-Second AAAI Conference on Artificial Intelligence.

[Stacking machine learning models for speech sentiment analysis | by Corentin Garet | Towards Data Science](https://towardsdatascience.com/stacking-machine-learning-models-for-speech-sentiment-analysis-adf433488845)

**CHAPTER 3**

**HIDDEN MARKOV MODEL**

Most modern speech recognition system uses Hidden Markov Model(HMM) which uses multiple inputs ,However it’s not automatic it has to be filled with some data which we already know in order to rain our model.

To better understand Markov models, let’s look at an example. Say you have a bag of marbles that contains four marbles: two red marbles and two blue marbles. You randomly select a marble from the bag, note its color, and then put it back in the bag. After repeating this process several times, you begin to notice a pattern: The probability of selecting a red marble is always two out of four, or 50%. This is because the probability of selecting a particular color of marble is determined by the number of that color of marble in the bag. In other words, the past history (i.e., the contents of the bag) determines the future state (i.e., the probability of selecting a particular color of marble).

This example illustrates the concept of a Markov model: the future state of a system is determined by its current state and past history. In the case of the bag of marbles, the current state is determined by the number of each color of marble in the bag. The past history is represented by the contents of the bag, which determine the probabilities of selecting each color of marble.

Markov models have many applications in the real world, including predicting the weather, stock market prices, and the spread of disease. Markov models are also used in natural language processing applications such as speech recognition and machine translation. **In speech recognition, Markov models are used to identify the correct word or phrase based on the context of the sentence.** In machine translation, Markov models are used to select the best translation for a sentence based on the translation choices made for previous sentences in the text.

The [hidden Markov model (HMM)](https://en.wikipedia.org/wiki/Hidden_Markov_model) is another type of Markov model where there are few states which are hidden. This is where HMM differs from a Markov chain. HMM is a statistical model in which the system being modeled are Markov processes with unobserved or hidden states. It is a hidden variable model which can give an observation of another hidden state with the help of the Markov assumption. The hidden state is the term given to the next possible variable which cannot be directly observed but can be inferred by observing one or more states according to Markov’s assumption. **Markov assumption is the assumption that a hidden variable is dependent only on the previous hidden state.**Mathematically, the probability of being in a state at a time t depends only on the state at the time (t-1). It is termed a **limited horizon assumption**. Another Markov assumption states that the **conditional distribution over the next state, given the current state, doesn’t change over time**. This is also termed a **stationary process assumption**.

## What are some libraries which can be used for training hidden Markov models?

* One of the popular hidden Markov model libraries is [PyTorch](https://vitalflux.com/tags/pytorch)-HMM, which can also be used to train hidden Markov models. The library is written in Python and it can be installed using PIP.
* HMMlearn: Hidden Markov models in [Python](https://vitalflux.com/category/python)
* PyHMM: PyHMM is a hidden Markov model library for Python.
* DeepHMM: A PyTorch implementation of a Deep Hidden Markov Model
* HiddenMarkovModels.jl
* HMMBase.jl

**CHAPTER 4**

**TECHNOLOGY USED**

1 . i5 processor x64

2 . windows 11

3 . VsCode ,Pycharm and Jupyter Notebook

4 . Python 3.9.4 ,HTML and CSS

5 . Different Python Libraries like Librosa ,Numpy,Pandas,Flask,TensorFlow and Keras etc.

6 . Datasets used for training – RAVDESS and Stream Of Consciousness

**VsCode**

**Visual Studio Code**, also commonly referred to as **VS Code**,[[9]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-9) is a [source-code editor](https://en.wikipedia.org/wiki/Source-code_editor) made by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) with the [Electron Framework](https://en.wikipedia.org/wiki/Electron_(software_framework)), for [Windows](https://en.wikipedia.org/wiki/Windows), [Linux](https://en.wikipedia.org/wiki/Linux) and [macOS](https://en.wikipedia.org/wiki/MacOS).[[10]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-TechCrunch-10) Features include support for [debugging](https://en.wikipedia.org/wiki/Debugging), [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [intelligent code completion](https://en.wikipedia.org/wiki/Intelligent_code_completion), [snippets](https://en.wikipedia.org/wiki/Snippet_(programming)), [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring), and embedded [Git](https://en.wikipedia.org/wiki/Git). Users can change the [theme](https://en.wikipedia.org/wiki/Theme_(computing)), [keyboard shortcuts](https://en.wikipedia.org/wiki/Keyboard_shortcut), preferences, and install [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)) that add functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [Fortran](https://en.wikipedia.org/wiki/Fortran), [Go](https://en.wikipedia.org/wiki/Go_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [Node.js](https://en.wikipedia.org/wiki/Node.js), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language)).[[16]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-16)[[17]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-17)[[18]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-18)[[19]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-19) It is based on the [Electron](https://en.wikipedia.org/wiki/Electron_(software_framework)) framework,[[20]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-ars-electron-20) which is used to develop [Node.js](https://en.wikipedia.org/wiki/Node.js) [web applications](https://en.wikipedia.org/wiki/Web_application) that run on the [Blink layout engine](https://en.wikipedia.org/wiki/Blink_layout_engine). Visual Studio Code employs the same editor component (codenamed "Monaco") used in [Azure DevOps](https://en.wikipedia.org/wiki/Azure_DevOps_Server) (formerly called Visual Studio Online and Visual Studio Team Services).[[21]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-21)

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [bracket matching](https://en.wikipedia.org/wiki/Bracket_matching), [code folding](https://en.wikipedia.org/wiki/Code_folding), and configurable snippets. Visual Studio Code also ships with [IntelliSense](https://en.wikipedia.org/wiki/Intelligent_code_completion) for JavaScript, TypeScript, [JSON](https://en.wikipedia.org/wiki/JSON), [CSS](https://en.wikipedia.org/wiki/CSS), and [HTML](https://en.wikipedia.org/wiki/HTML), as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.[[22]](https://en.wikipedia.org/wiki/Visual_Studio_Code#cite_note-22)

**PyCharm**

**PyCharm** is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used for programming in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control](https://en.wikipedia.org/wiki/Version_control) systems, and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)). PyCharm is developed by the Czech company [JetBrains](https://en.wikipedia.org/wiki/JetBrains).[[5]](https://en.wikipedia.org/wiki/PyCharm#cite_note-5)

It is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform_software), working on [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux). PyCharm has a Professional Edition, released under a [proprietary license](https://en.wikipedia.org/wiki/Proprietary_software) and a Community Edition released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License).[[6]](https://en.wikipedia.org/wiki/PyCharm#cite_note-community-6) PyCharm Community Edition is less extensive than the Professional Edition.

**Jupyter NoteBook**

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The “notebook” term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context.

According to the official website of [Jupyter](https://jupyter.org/" \t "_blank), Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Jupyter Book is an open-source project for building books and documents from computational material. It allows the user to construct the content in a mixture of Markdown, an extended version of Markdown called MyST, Maths & Equations using MathJax, Jupyter Notebooks, reStructuredText, the output of running Jupyter Notebooks at build time. Multiple output formats can be produced (currently single files, multipage HTML web pages and PDF files).

**RAVDESS**

RAVDESS stands for **Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)**. This database contains 7356 files (total size: 24.8 GB). The database contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity(normal, strong), with an additional neutral expression. All conditions are avail-able in three modality formats: Audio-only (16bit, 48kHz .wav), Audio-Video(720p H.264, AAC 48kHz, .mp4), and Video-only (no sound).”

**STREAM OF CONSCIOUSNESS**

the **Stream-of-consciousness** dataset that was gathered in a study by Pennebaker and King [1999]. It consists of a total of 2,468 daily writing submissions from 34 psychology students (29 women and 5 men whose ages ranged from 18 to 67 with a mean of 26.4). The writing submissions were in the form of a course unrated assignment. For each assignment, students were expected to write a minimum of 20 minutes per day about a specific topic. The data was collected during a 2-week summer course between 1993 to 1996. Each student completed their daily writing for 10 consecutive days. Students’ personality scores were assessed by answering the Big Five Inventory (BFI) [John et al., 1991]. The BFI is a 44-item self-report questionnaire that provides a score for each of the five personality traits. Each item consists of short phrases and is rated using a 5-point scale that ranges from 1 (disagree strongly) to 5 (agree strongly). An instance in the data source consists of an ID, the actual essay, and five classification labels of the Big Five personality traits. Labels were originally in the form of either yes (‘y’) or no (‘n’) to indicate scoring high or low for a given trait.

**PYTHON LIBRARIES**

**Librosa**

Librosa is valuable Python music and sound investigation library that helps programming designers to fabricate applications for working with sound and music document designs utilizing Python. This Python bundle for music and sound examination is essentially utilized when we work with sound information, like in the music age (utilizing Lstm's), Automatic Speech Recognition.

The library is not difficult to utilize and can deal with fundamental as well as cutting-edge errands connected with sound and music handling. It is open source and uninhibitedly accessible under the ISC License.

The library upholds a few elements connected with sound records handling and extraction like burden sound from a circle, register of different spectrogram portrayals, symphonious percussive source detachment, conventional spectrogram decay, stacks and translates the sound, Time-space sound handling, successive demonstrating, coordinating consonant percussive partition, beat-simultaneous and some more.

Librosa assists with picturing the sound signs and furthermore does the component extractions in it utilizing different sign handling methods.

**Installation**

It can be installed in your project using command **pip install librosa** or **Conda install librosa** if you are using Anaconda Navigator.

**It will assist us with executing:**

1. Sound sign investigation for music. The library gives a lot of adaptabilities to master clients who might be keen on handling sound records.
2. Reference execution of normal techniques. It gives the structure blocks important to make the music data recovery frameworks.
3. Building blocks for Music data recovery (MIR).

**Numpy**

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements.

**Travis Oliphant** created NumPy package in 2005 by injecting the features of the ancestor module Numeric into another module Numarray.

It is an extension module of Python which is mostly written in C. It provides various functions which are capable of performing the numeric computations with a high speed.

NumPy provides various powerful data structures, implementing multi-dimensional arrays and matrices. These data structures are used for the optimal computations regarding arrays and matrices.

With the revolution of data science, data analysis libraries like NumPy, SciPy, Pandas, etc. have seen a lot of growth. With a much easier syntax than other programming languages, python is the first choice language for the data scientist.

NumPy provides a convenient and efficient way to handle the vast amount of data. NumPy is also very convenient with Matrix multiplication and data reshaping. NumPy is fast which makes it reasonable to work with a large set of data.

There are the following advantages of using NumPy for data analysis.

1. NumPy performs array-oriented computing.
2. It efficiently implements the multidimensional arrays.
3. It performs scientific computations.
4. It is capable of performing Fourier Transform and reshaping the data stored in multidimensional arrays.
5. NumPy provides the in-built functions for linear algebra and random number generation.

**Installation**

It can be installed in your project using command **pip install numpy** or **Conda install numpy** if you are using Anaconda Navigator.

**Pandas**

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word **Panel Data**, which means **an Econometrics from Multidimensional data**. It is used for data analysis in Python and developed by **Wes McKinney** in **2008**.

Data analysis requires lots of processing, such as **restructuring, cleaning** or **merging**, etc. There are different tools are available for fast data processing, such as **Numpy, Scipy, Cython**, and **Panda**. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools.

Pandas is built on top of the **Numpy** package, means **Numpy** is required for operating the Pandas.

Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., **load, manipulate, prepare, model, and analyze**.

**Features**

* It has a fast and efficient DataFrame object with the default and customized indexing.
* Used for reshaping and pivoting of the data sets.
* Group by data for aggregations and transformations.
* It is used for data alignment and integration of the missing data.
* Provide the functionality of Time Series.
* Process a variety of data sets in different formats like matrix data, tabular heterogeneous, time series.
* Handle multiple operations of the data sets such as subsetting, slicing, filtering, groupBy, re-ordering, and re-shaping.
* It integrates with the other libraries such as SciPy, and scikit-learn.
* Provides fast performance, and If you want to speed it, even more, you can use the **Cython**.

**Installation**

It can be installed in your project using command **pip install pandas** or **Conda install pandas** if you are using Anaconda Navigator.

**Flask**

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. In the case of Flask, its dependencies are:

* [Werkzeug](http://werkzeug.pocoo.org/) a WSGI utility library
* [jinja2](http://jinja.pocoo.org/) which is its template engine

**Installation**

It can be installed in your project using command **pip install Flask** or **Conda install Flask** if you are using Anaconda Navigator.

You can Directly install Flask to your project without affecting the other distribution.

It Basically Acts like a mediator between API and UI and Hence Plays a very important role in our application.It is easy to learn and beginner Friendly.

**TensorFlow**

Tensorflow is an open-source library for numerical computation and large-scale machine learning that ease Google Brain TensorFlow**,**acquiring data, training models, serving predictions, and refining future results.

Tensorflow bundles together Machine Learning and Deep Learning models and algorithms. It uses Python as a convenient front-end and runs it efficiently in optimized C++.

Tensorflow allows developers to create a graph of computations to perform. Each node in the graph represents a mathematical operation, and each connection represents data. Hence, instead of dealing with low details like figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application.

In the deep learning artificial intelligence research team at Google, Google Brain, in the year 2015, developed **TensorFlow** for Google’s internal use. The research team uses this Open-Source Software library to perform several important tasks.  
TensorFlow is, at present, the most popular software library. There are several real-world applications of deep learning that make TensorFlow popular. Being an Open-Source library for deep learning and machine learning, TensorFlow plays a role in text-based applications, image recognition, voice search, and many more. DeepFace, Facebook’s image recognition system, uses TensorFlow for image recognition. It is used by Apple’s Siri for voice recognition. Every Google app has made good use of TensorFlow to improve your experience.

**Tensors**

All the computations associated with TensorFlow involve the use of tensors.

A tensor is a vector/matrix of n-dimensions representing types of data. Values in a tensor hold identical data types with a known shape, and this shape is the dimensionality of the matrix. A vector is a one-dimensional tensor; a matrix is a two-dimensional tensor. A scalar is a zero-dimensional tensor.

In the graph, computations are made possible through interconnections of tensors. The mathematical operations are carried by the node of the tensor, whereas a tensor’s edge explains the input-output relationships between nodes.  
Thus TensorFlow takes an input in the form of an n-dimensional array/matrix (known as tensors), which flows through a system of several operations and comes out as output. Hence the name TensorFlow. A graph can be constructed to perform necessary operations at the output.

**Installation**

It can be installed in your project using command pip3 install --upgrade tensorflow or **Conda install tensorflow** if you are using Anaconda Navigator.

This is a deep learning library and used complex application . This a so vast it is software in itself advanced Machine learning is completely dependent on it.

**KERAS**

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user-friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

It cannot handle low-level computations, so it makes use of the **Backend** library to resolve it. The backend library act as a high-level API wrapper for the low-level API, which lets it run on TensorFlow, CNTK, or Theano.

Initially, it had over 4800 contributors during its launch, which now has gone up to 250,000 developers. It has a 2X growth ever since every year it has grown. Big companies like Microsoft, Google, NVIDIA, and Amazon have actively contributed to the development of Keras. It has an amazing industry interaction, and it is used in the development of popular firms likes Netflix, Uber, Google, Expedia, etc.

**Features**

* Focus on user experience has always been a major part of Keras.
* Large adoption in the industry.
* It is a multi backend and supports multi-platform, which helps all the encoders come together for coding.
* Research community present for Keras works amazingly with the production community.
* Easy to grasp all concepts.
* It supports fast prototyping.
* It seamlessly runs on CPU as well as GPU.
* It provides the freedom to design any architecture, which then later is utilized as an API for the project.

**Installation**

It can be installed in your project using command **pip install keras** or **conda install -c anaconda keras**  if you are using Anaconda Navigator.

It is a good practice to create a separate environment for using keras as it is a very heavy library and can be used efficiently in its own environment.

**CHAPTER 5**

**PROBLEM STATEMENT AND EXISTING SYSTEMS**

**PROBLEM STATEMENT**

Sentiment analysis is a widely researched area of ​​Natural Language Processing (NLP) research. However, most existing research has focused on the analysis of textual data, and few have explored the potential of sentiment analysis concerning speech data This problem statement attempts to fill this gap and offers a new approach to discourse sentiment analysis.

Language is a rich source of information that can convey emotions, attitudes, and opinions However, sentiment analysis of voice data presents several challenges First, speech is a continuous signal, and it is not easy to break it down into meaningful units. Second, emotions and attitudes are expressed through various acoustic properties, such as pitch, volume, and duration, and are difficult to identify and quantify.

In addition, voice data is often accompanied by background noise, which can affect the accuracy of sentiment analysis.

To overcome these challenges, this study proposes a deep learning approach that uses acoustic and linguistic features of speech data for sentiment analysis The proposed model uses a combination of convolutional and recurrent neural networks to extract important features of the speech signal The model then integrates these features with linguistic information obtained from the speech transcription to predict the speaker's mood.

The proposed method will be evaluated using a large-scale speech dataset collected from various sources such as podcasts, radio broadcasts, and public speeches Model performance is measured by accuracy, precision, recall, and F1 score. The findings of this study have implications for a variety of applications, including customer feedback analysis, political sentiment analysis, and market research.

**EXISTING SYSTEMS**

The systems that exist already include doing the sentiment analysis on text data. The process of implementing sentiment analysis on text data involves several steps, including data collection, preprocessing, feature extraction, model selection, and evaluation.

Firstly, a large and representative dataset of text data needs to be collected and annotated with sentiment labels. Preprocessing steps such as tokenization, stemming, and stop-word removal are then applied to the text data to normalize and reduce its dimensionality. Feature extraction methods such as bag-of-words, word embeddings, and topic models are then used to represent the text data in a format suitable for machine learning algorithms. A sentiment analysis model is then selected and trained on the preprocessed data, and its performance is evaluated using various metrics such as accuracy, precision, and recall.

Finally, the trained model is deployed and used to analyze new and unseen The models which try to run sentiment analysis over voice data also first convert the audio data to text with the help of some tools and then apply the same text-based sentiment analysis technique which is not the ideal solution of the problem.

**CHAPTER 6**

**PROPOSED SYSTEM**

In this proposed system we are using a library called LIBROSA. It is a powerful Python library for analyzing and processing audio data. It provides a wide range of tools for feature extraction, signal processing, and visualization, making it a popular choice for researchers and practitioners in fields such as music information retrieval, speech processing, and acoustic analysis.

We will analyze the sound using this librosa library and analyze it by using a feature of audio files called mfcc.

Mel-frequency cepstral coefficients (MFCCs) are widely used features in audio signal processing for tasks such as speech recognition, music classification, and speaker identification. They are derived from the Fourier transform of the audio signal and represent a compact and discriminative representation of the spectral content of the audio signal in the mel-scale frequency domain.

Librosa library contains a function to extract this mfcc coefficient. We’ll make a neural network and will train that model with the help of 40 mfcc features extracted from the audio.

With the help of those 40 mfcc features and right dataset (RAVDESS,TESS), the system will be able to properly detect the sentiments of the speaker which also will not depend on the language which the speaker is speaking as it will detect the emotions in the voice of the speaker and not the words.

**CHAPTER 7**

**STEPS**

**CHAPTER 8**

**ACCURACY,SCOPE AND APPLICATIONS**

**CHAPTER 9**

**CONCLUSION**

In conclusion, sentiment analysis on speech data presents a challenging but promising research area in natural language processing. The proposed deep learning-based approach that leverages both acoustic and linguistic features has the potential to improve the accuracy and robustness of sentiment analysis on speech data. The evaluation of this approach on a large-scale dataset of speech data has shown promising results, with high accuracy and F1 scores achieved. The findings of this study have important implications for practical applications such as customer feedback analysis, political sentiment analysis, and market research.

However, several challenges need to be addressed to further improve the performance of sentiment analysis on speech data. For example, the impact of various types of background noise on sentiment analysis needs to be studied, and more effective feature extraction techniques need to be developed to capture the complex relationships between acoustic and linguistic features. Moreover, the generalizability of the proposed approach needs to be validated on speech data from different languages and cultures.

Overall, sentiment analysis on speech data has the potential to provide valuable insights into the emotions, attitudes, and opinions expressed through speech, and can contribute to a wide range of fields such as psychology, sociology, and marketing. Further research in this area will undoubtedly lead to new and exciting advancements in the field of natural language processing.

**REFERENCES**

[1] R. M. Kumar and S. Ieee, “Sentiment Analysis on Speaker Specific Speech Data,” 2013.

[2] R. R. Sehgal, shubham Agarwal, and G. Raj, *Interactive Voice Response using Sentiment  Analysis in Automatic Speech Recognition  Systems*. IEEE, 2018.

[3] M. Bansal, S. Yadav, and D. K. Vishwakarma, “A Language-Independent Speech Sentiment Analysis Using Prosodic Features,” in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Apr. 2021, pp. 1210–1216. doi: 10.1109/ICCMC51019.2021.9418357.

[4] R. Rajak and R. Mall, “Emotion recognition from audio,dimensional and discrete categorization using CNNs,” 2019.