DS 5010 Report Due: 4/21/2024

Done by: Sujal Thakkar

- **Title:** Mental Health Emotion Detection
- **Author:** Sujal Thakkar
- **Github:** https://github.com/TSujal/Mental-Health-Emotion-Detection.git
- **Summary**

  The Project aims to recognize Emotion or we can even say that it is able to detect the mental status/emotion of a given person using its voice as an audio data which we fed into our Machine learning and Deep learning model. The dataset which we use in this project that consists of the audio files which is diversified into speech of 2 actor and each actor having 7 classes in them which is basically the human emotion which are been labeled as [Fear, Sad, Happy, Surprised, Disgusted, Angry, Neutral].

  As it has been obvious the emotions are basic human emotions which have been captured in the speech of the individual actor in which each emotion has nearly 200 different sentences which a particular actor speaks in a given one emotion. For example to demonstrate the dataset, let's assume actor 1 is speaking 200 different lines in a Fearful manner that way he speaks all the 7 labeled emotions i.e. each emotion per actor is having 200 samples of audio data. Similarly for actor 2 which has exactly 7 labeled emotion and each has again 200 samples. Thus in Total we will be having 400 samples per emotion.

  We utilize the python Matplotlib library and seaborn library to plot the various graph for our dataset, also we do uses scikit-learn library to import machine learning and deep learning algorithms where in to be specific we have used 2 Machine learning algorithm which works quite well in the classification problem.

  And also we have implemented a Deep Learning Feed Forward model in which we have achieved our highest accuracy.

- **Design:**
  - **Data Collection and Preprocessing:**

    We fetched the dataset from kaggle, where we analyzed various dataset in the beginning and at last, we found the proper dataset which had been bifurcated properly into all different classes. The dataset was properly labeled as well so no more grinding of data for the project was needed.

  - **Exploratory Data Analysis (EDA):**

    We then visualized the distribution of emotions using various plots where we did take the audio data and using the visualizing library and Librosa as a package we plotted some waveforms and spectrograms by which we were able to witness the difference in the graphs of each and every emotion.

  - **Feature Extraction**

    Feature Extraction is one of the crucial steps without which we cannot carry forward our project. Where we convert our audio samples to numeric form.

    Now, Converting audio to numeric format can be done in many ways but we have implemented it using MFCCs which stands for Mel-Frequency Cepstral Coefficients. Which has basically been simply done using Librosa library.

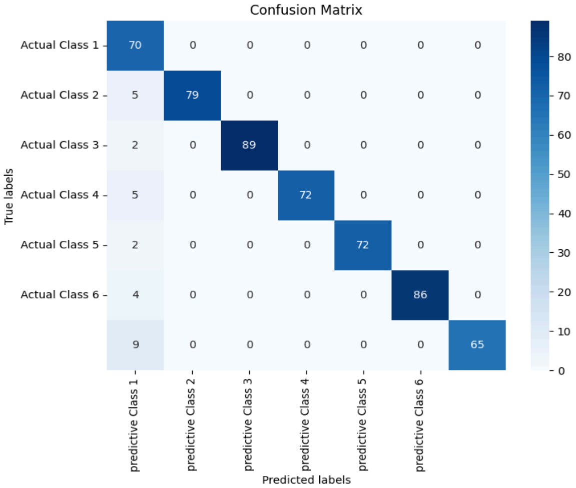
  - **Machine learning Algorithms:**

    We implemented Here Random Forest Classifier and Decision Tree algorithm to classify emotions based on the MFCC features.

    - **Random Forest Classifier:**

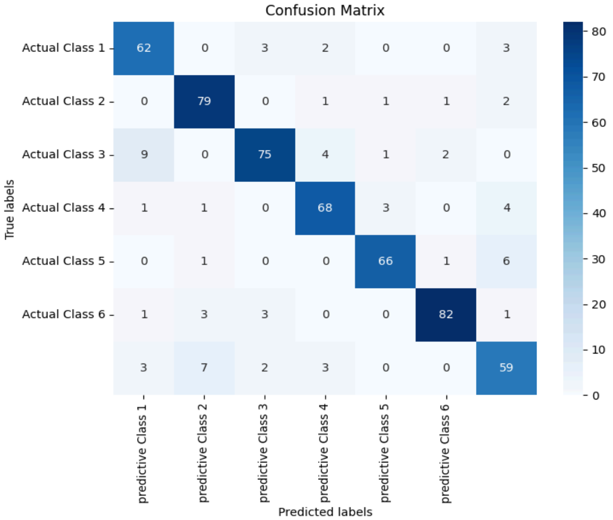
    Where in to be precise we have achieved the accuracy of 94.82% in Random Forest Classifier. We have also implemented cross validation methodology in here were in we test this algorithm by dividing the whole data into 5 samples and no sample matches each other and in cross validation methodology we fed in this different sample of data also shuffling them so as to get the score of accuracy upon each sample set so we can accurately justify the accuracy where in by taking the mean of 5 accuracy score we reach up-to accuracy score of 96.03%.

We also perform the confusion matrix to showcase the Actual class vs the predicted class graph as a matrix distribution.



- **Decision Tree**

Just for testing proposed we do implement Decision Tree which is one of the famous algorithms, easy to implement and does not worry much about the shape of the input data which makes it sometimes easy and first algorithm for programmers to implement/test their data upon.
Where we achieved the accuracy of 89.67%, then later on testing the algorithm up-on Cross validation score we got the mean accuracy of 5 models to be 90.25%. Which is a bit lower than random forest.
Performing again the confusion matrix which showcases where our actual class is getting wrong when predicting it through a decision tree algorithm, which based on the above matrix image for random forest will be a bit different as clearly the accuracy is a bit low so the mistake/false positive will be much more in the matrix**.**



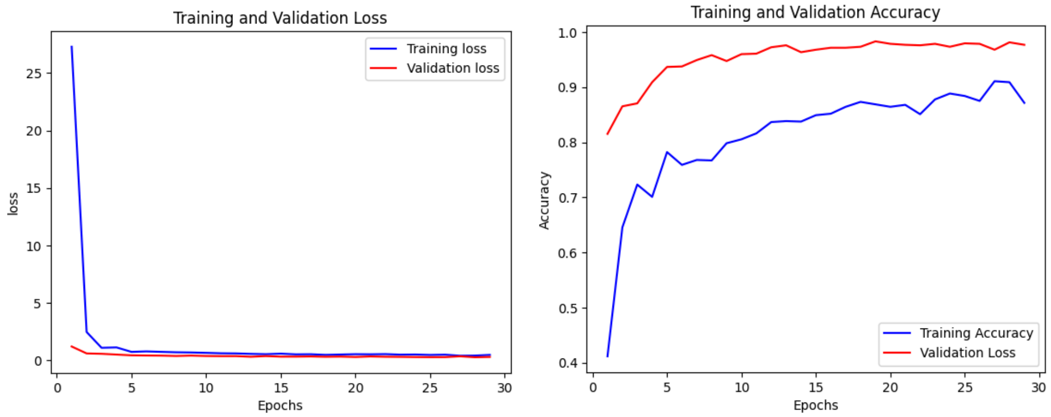
The confusion matrix speaks a lot about the mistakes our decision tree is making when given an actual class of Fear and predicting Sad is some mistakes we can witness here for better understanding.

Deep learning Model:

Here in we have implemented many models Recurrent Neural Network in which because I presumed the data to be continuous in nature because its a audio data i have implement LSTM layer model in which the model was not performing that accurately, plus also overfitting which was a disaster model to begin with but later simplifying the model I went with the Feed forward Neural network in which i do have bunch of Dense layer and a output layer with specific activation function "Relu" and for output layer the activation function is "Softmax".

Also the optimizer i used here was Adam in which the learning rate kept as 0.001. Where in the layer i have also used regularization techniques which penalize the model when performing inaccurate thus giving me solution for overfitting plus while splitting the Data before feeding into a Deep learning model
I have splitted the data into 3 format Training, Validation and Testing in which the Training data we do used inside the model where validation data is for validation while model is running the model will be testing the accuracy upon the validation data but after the model stops training we do evaluate the our model on total unseen data which is our Testing Data which is a bit lower samples but totally unseen Thus the model when ran upon the Test data gives accuracy of 98.21% precisely. While beside the accuracy the model also gives us the loss which is equally important to witness. And the Test Loss we got in here is 0.31.

Thus we plot the model's performance after making it to run upon barely 30 epochs.
We plot its loss graph ie training loss vs validation loss



Which not only depicts the increase in accuracy when run upon 0 to 30 epochs but also tells us that the model is not overfitting if not the difference in validation and training would be a huge gap. Which I witnessed when using the RNN (LSTM) Layer model in the beginning.

- **Usage:** We did use Librosa as our main package while dealing with the audio data which was the soul of the project because without the librosa package we would not be able to implement the audio data nor would be able to easily convert to MFCC.
- **Discussion:** The project provides a comprehensive solution for mental emotion detection of the human mind and behavior by their speech. Which could be crucial to identify the current mood of the person.
- **Statement of Contribution:** Sujal Thakkar : Completed the entire project, Including data collection, preprocessing, Feature Extraction, Model Implementation, Evaluation, Documentation, and Repository Management.
- **Reference:**
  Data-Set: https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess
  Libraries: Numpy, Librosa, Pandas, Scikit-Learn, TensorFlow, Keras