Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to Dr. Bivas Mitra for his exemplary guidance, monitoring, and constant encouragement throughout the course of this project. His insights and expertise have been invaluable.

I also extend my heartfelt thanks to Mam Salma Mandi for her unwavering support and assistance throughout the duration of the project. Her contributions and guidance were crucial to the completion of my work.

The motivation I gained from this project will go a long way in shaping my future endeavours.

Contents

Introduction

A context-sensitive mobile application can sense various user contexts such as location, mental state, and physical environment, and respond accordingly. Among these contexts, the user's emotion plays a crucial role in the decisions an application makes to provide its services. Therefore, it is essential to recognize the user's emotions accurately.

Typically, a context-sensitive application relies on a supervised machine learning model to detect emotions. This model is trained on data collected from mobile sensors, correlated with emotion labels. These emotion labels are usually collected through the Experience Sampling Method (ESM), where users manually provide self-reports. However, in long-term studies, the manual collection of self-reports imposes a burden on users, leading to incorrect reports or participants dropping out of the study. Thus, it is essential to find alternative strategies to reduce participation burden.

Literature shows that various auxiliary modalities, such as facial images, voice clips, physiological signals, and body gestures, are correlated with emotion. The SELFI paper proposed a methodology to leverage facial images to reduce the participation burden. This approach opens up opportunities to investigate other modalities for reducing the burden since emotions are expressed through multiple channels. Additionally, it allows for a comparative study of the effectiveness of different modalities in emotion self-report detection.

Objective: Explore different auxiliary modalities to reduce the self-report burden leveraging the SELFI framework. Specifically, in this project, we experimented with video and voice clips of individual subjects.

Data Collection:

We are using the MELD dataset, a popular dataset that includes scenes from the TV show "Friends." This annotated dataset comprises approximately 1,400 dialogues and 13,000 utterances, segmented into Training (Train), Development (Dev), and Test sets. Each utterance is meticulously annotated with features such as:

- **Utterance:** Individual utterances
- **Speaker:** Name of the speaker associated with the utterance
- **Emotion:** The emotion expressed by the speaker in the utterance
- **Sentiment:** Sentiment expressed in the utterance
- **Dialogue_ID:** The index of the dialogue starting from 0
- **Utterance_ID:** The index of the particular utterance in the dialogue starting from 0
- **Season:** The season number to which a particular utterance belongs
- **Episode:** The episode number to which a particular utterance belongs
- **StartTime:** The starting time of the utterance in the given episode in the format "hh:mm
  ,ms"
- **EndTime:** The ending time of the utterance in the given episode in the format "hh:mm
  ,ms"

We have in total **304 unique speakers** in the given dataset but only 6 speakers have uttered more than 90 utterances which implies that we only have **6 speakers** who have more than **90 data points.**

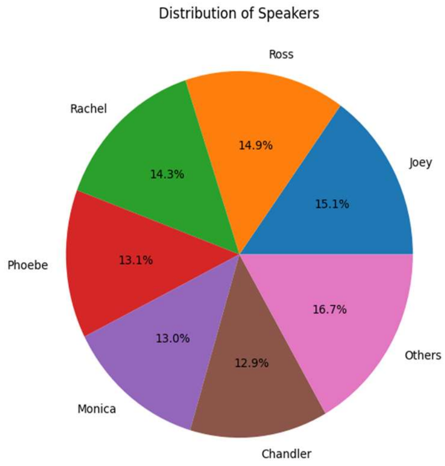


Fig : Distribution of unique speakers

Others consist of the rest of all the 298 speakers for whom we have less than 90 data points.

The six speakers are:

Joey, Ross, Rachel, Phoebe, Monica and Chandler.

For the emotion labels we have seven different emotions consisting of neutral, joy, surprise, anger, sadness, fear and disgust.

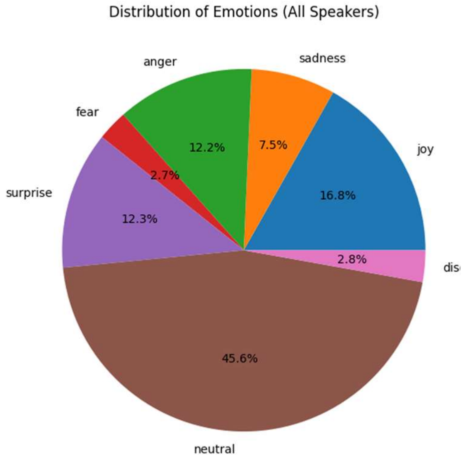


Fig : Distribution of unique emotions

Data Processing:

The dataset was filtered to focus on the six main speakers: Rachel, Ross, Joey, Chandler, Monica, and Phoebe. Data points with emotion labels "neutral" and "surprise" were removed. This left us with 4,085 data points consisting of six speakers and five emotions.

a. Separation of the Speakers:

We have filtered the dataset to create six subsets, each corresponding to one of the main six speakers: **Rachel**, **Ross, Joey**, **Chandler**, **Monica**, and **Phoebe**.

We have also dropped data points with emotion label neutral and surprise.
After we have filtered our dataset we are left with roughly 4085 data points consisting of six speakers and five emotions.

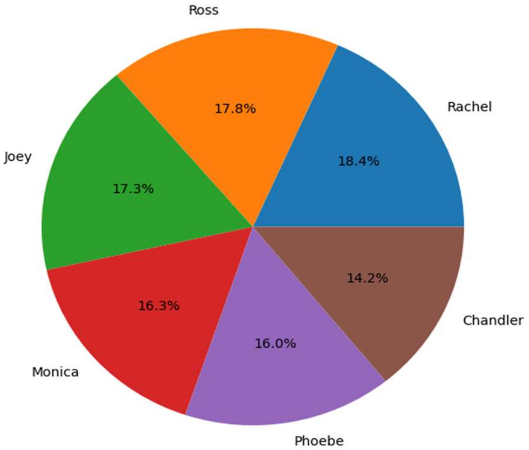


Fig : Distribution of six speakers(filtered)

Distribution of speakers after filtering with their number of data points associated with each:

Joey	:	833
Monica	:	784
Chandler	:	681
Phoebe	:	769
Ross	:	853
Rachel	:	884

Distribution of emotions after filtering with their number of data points associated with each:

Joy	:	1920
Anger	:	1397
Sadness	:	855
Fear	:	310
Disgust	:	322

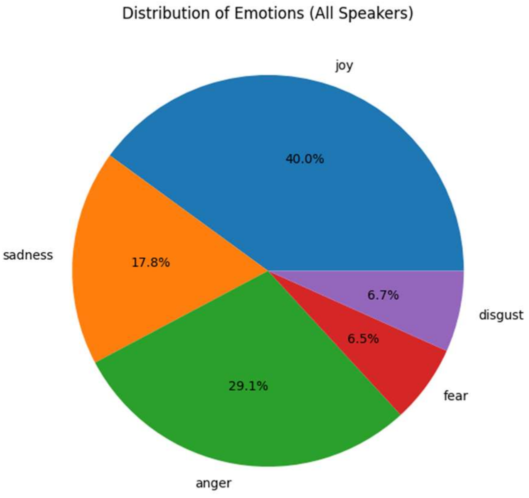


Fig : Distribution of five emotions(filtered)

b. Mapping of Emotions with Valence and Arousal:

Valence and arousal are two dimensions used to describe emotions in the Circumplex model. Valence refers to the positive (pleasant) or negative (unpleasant) degree of an emotion, while arousal indicates the level of activation or energy associated with the emotion.

As directed, we have ignored the "neutral" and "surprise" emotions before mapping as it is difficult to map those 2 emotions into high and low valence(arousal). For the remaining emotions, we have used the following mapping table:

Emotion	**valence**	**Arousal**
Fear	0	1
Sadness	0	0
Joy	1	1
Disgust	0	1
Anger	0	1

Table 1: Mapping of emotions with valence and arousal

Our mapping is based on the understanding that emotions like fear, disgust, and anger have a negative valence and positive arousal, while sadness has a negative valence and arousal. Joy, on the other hand, has a positive valence and arousal. This mapping helps in quantifying the emotional context for better analysis and modelling.

c. Allocation of Scene Number:

Based on the dialogue ID and utterance ID, we have allocated a scene number representing the complete dialogue. Additionally, we have sorted the data by dialogue ID and utterance ID to maintain the sequence.

After performing these steps, our basic dataset is ready for further use.

Methodology:

The SELFI framework aims to predict emotional states using a combination of self-reported emotion labels and auxiliary data (facial image or audio). The framework involves three main processing blocks: Generation of self-report features, calculation of auxiliary modality features, and feature reduction . The goal is to develop models that can predict valence and arousal using crucial features extracted from self-reported data combined with facial or audio features.

Generation of Self-report Features:

Using the SELFI paper as a reference, we have incorporated the concepts of influence and sequence length into our self-report dataset to predict emotions more accurately. The main objective is to calculate the Influence (Fei) and Emotion Sequence Length (Lei) for each emotion label.

a. Definitions and Justifications:

1. **Influence (Fei):** Influence measures how the current self-report emotion ei affects the next self-report emotion ei+1. To compute Influence, we use a 2 × 2 state-transition matrix (P), where each element p_{ij} represents the probability of transitioning from emotion label ei to label ej. The influence also considers the normalised elapsed time t_{ei+1} between the current and next self-reports. This feature helps in understanding how persistent an emotion is over time and its impact on subsequent emotions.
   The formula for Influence is:

   $$F_{ei+1} = p_{eiei+} \times (1 - t_{ei+1})$$

   Where p_{eiei+} is the probability of transitioning from ei to ei+1 and $1 - t_{ei+1}$ is the weight of the current emotion's impact over time.
   Emotion Sequence Length (Lei): Emotion Sequence Length captures how long a user continues to report the same emotion ei consecutively. It reflects the consistency of the emotion state over a sequence of self-reports. This feature is crucial for identifying patterns in emotional responses and understanding the duration of specific emotional states.
   The formula for Emotion Sequence Length is:

   L_{ei} = Number of consecutive occurrences of ei

b. Processing Steps:

1. **Transition Matrix:** First, we calculate the transition matrix (P) to determine the probabilities of transitioning between different emotion states. The transition matrix is computed by counting the transitions between valence states (0 or 1) and normalising these counts to obtain probabilities.
2. **Calculation of Influence:** Using the transition matrix and elapsed time, we calculate the Influence Fei for each emotion transition. This helps in capturing the temporal dynamics of emotions.
3. **Emotion Sequence Length Calculation:** We determine the sequence length Lei for each emotion label to understand how long a specific emotion persists.

By integrating these features, our self-report dataset becomes a robust tool for predicting emotions. The combined use of influence and sequence length offers a detailed understanding of emotional dynamics, enhancing the accuracy of emotion recognition models.

Generation of Auxiliary Modality Features:

1. Facial Image Features:

Splitting of the video into frames:

Using OpenCV the video was splitted into frames, the videos available to us has the frame rate per second equal to 23.7, with each video having the total frames in the range of 3 to 1000. We have set the base interval of 0.25 seconds which means that after every 0.25 seconds we capture the frame and do further process of facial analysis. For the video which has less than 24 frames we have set intervals as 0.10 to capture most of the information from the video. Also if the total frame goes below 10 we do analysis of each and every frame.

Detection of face:

Amazon Rekognition is a powerful image and video analysis service that can identify objects, people, text, scenes, and activities, as well as detect inappropriate content. In our project, we leverage Amazon Rekognition to analyze faces within video frames, which helps in accurately identifying speakers associated with utterances.

In each frame, we use the DetectFaces function of Amazon Rekognition through its API to detect faces. We compare the detected faces with known faces to ensure that we are extracting the facial features of the correct speaker. This step is crucial for filtering out the speaker from the video, especially in scenes where multiple characters are present.

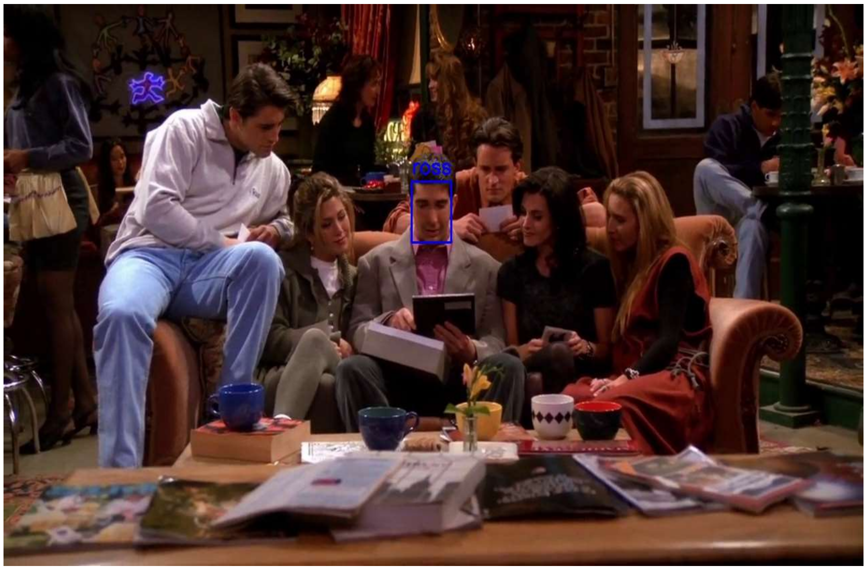


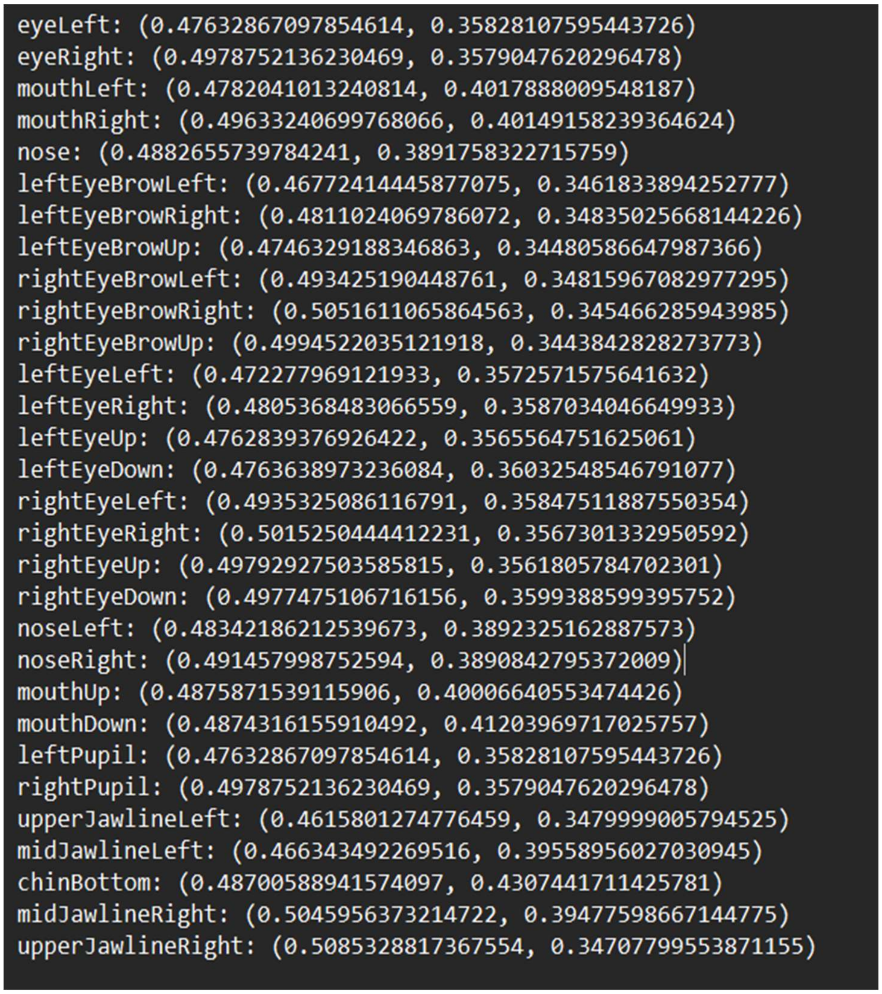
Fig : frame with detected face.

The above frame is from the video file dia176_utt9.mp4 with the speaker as ross. The speaker ross is correctly detected out of multiple faces.

Feature Extraction :

Once the correct face is identified, we extract the following features using Amazon Rekognition: bounding box, landmarks (eyes, mouth, nose, eyebrows, jawline, pupils), pose, quality, confidence, emotions (calm, surprised, confused, sad, happy, angry, fear, disgusted), age range, smile, eyeglasses, sunglasses, gender, beard, mustache, eyes open, and mouth open.

Our primary focus is on the landmarks, which are treated as the facial features of the person. Since the facial expression of a person in the video is almost similar across the split frames within that video, we can take the average of each feature to get a set of averaged landmarks. Thus, corresponding to each utterance associated with one video, we have an average facial feature or landmarks, consisting of 16 facial features.



```
eyeLeft: (0.47632867097854614, 0.35828107595443726)
eyeRight: (0.4978752136230469, 0.3579047620296478)
mouthLeft: (0.478204101324081‌4, 0.4017888009548187)
mouthRight: (0.49633240699768066, 0.40149158239364624)
nose: (0.4882655739784241, 0.3891758322715759)
leftEyeBrowLeft: (0.46772414445877075, 0.3461833894252777)
leftEyeBrowRight: (0.4811024069786072, 0.34835025668144226)
leftEyeBrowUp: (0.4746329188346863, 0.34480586647987366)
rightEyeBrowLeft: (0.493425190448761, 0.34815967082977295)
rightEyeBrowRight: (0.5051611065864563, 0.345466285943985)
rightEyeBrowUp: (0.4994522035121918, 0.3443842828273773)
leftEyeLeft: (0.472277969121933, 0.357257157564163‌2)
leftEyeRight: (0.4805368483066559, 0.3587034046649933)
leftEyeUp: (0.4762839376926422, 0.3565564751625061)
leftEyeDown: (0.4763638973236084, 0.36032548546791077)
rightEyeLeft: (0.4935325086116791, 0.35847511887550354)
rightEyeRight: (0.5015250444412231, 0.3567301332950592)
rightEyeUp: (0.49792927503585815, 0.3561805784702301)
rightEyeDown: (0.4977475106716156, 0.3599388599395752)
noseLeft: (0.48342186212539673, 0.389232516288757‌3)
noseRight: (0.491457998752594, 0.3890842795372009)
mouthUp: (0.4875871539115906, 0.40006640553474426)
mouthDown: (0.4874316155910492, 0.4120396971702575‌7)
leftPupil: (0.47632867097854614, 0.35828107595443726)
rightPupil: (0.4978752136230469, 0.3579047620296478)
upperJawlineLeft: (0.461580127477645‌9, 0.347999900579452‌5)
midJawlineLeft: (0.466343492269516, 0.39558956027030945)
chinBottom: (0.48700588941574097, 0.430744171142578‌1)
midJawlineRight: (0.5045956373214722, 0.39477598667144775)
upperJawlineRight: (0.5085328817367554, 0.34707799553871155)
```

Fig : Landmarks of the detected face in the above frame.

Averaging of Features:

Features extracted from each of the frames of an individual video is averaged to get the averaged landmarks, we do this as the facial expression of the person for a utterance in a video does not change much. These averaged features are collected for the entire dataset and are further used in the process of feature reduction.

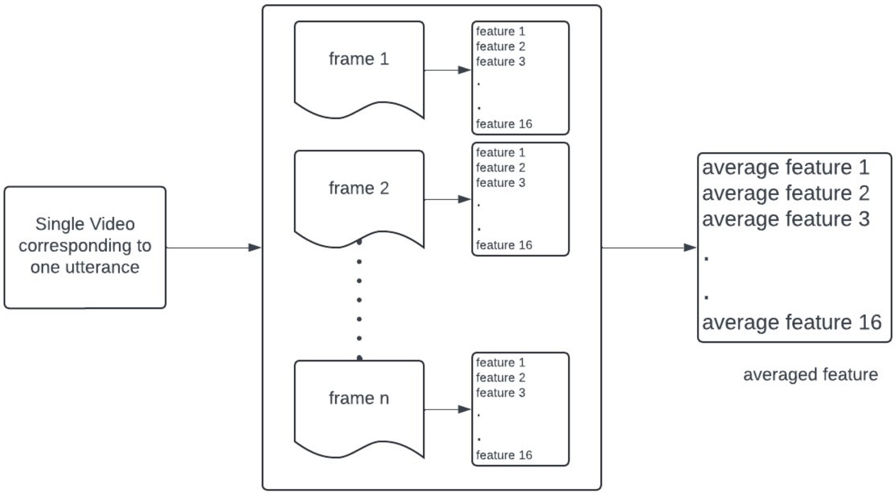


Fig : Pipeline for calculating average landmarks (features)

Feature Reduction:

We split the dataset into train and test sets and use Kernel Principal Component Analysis (KPCA) to reduce the dimensionality of our facial features. Specifically, we fit the KPCA model on the training set and transform the training set features using this fitted model. For the test set, we apply the transformation using the same KPCA model fitted on the training data. This approach ensures that the test data is transformed consistently with the training data, avoiding any data leakage or overfitting issues. By this method, we reduce the 16 features into a single facial feature corresponding to each utterance or a data point..

Pipeline of Facial Feature Computation :

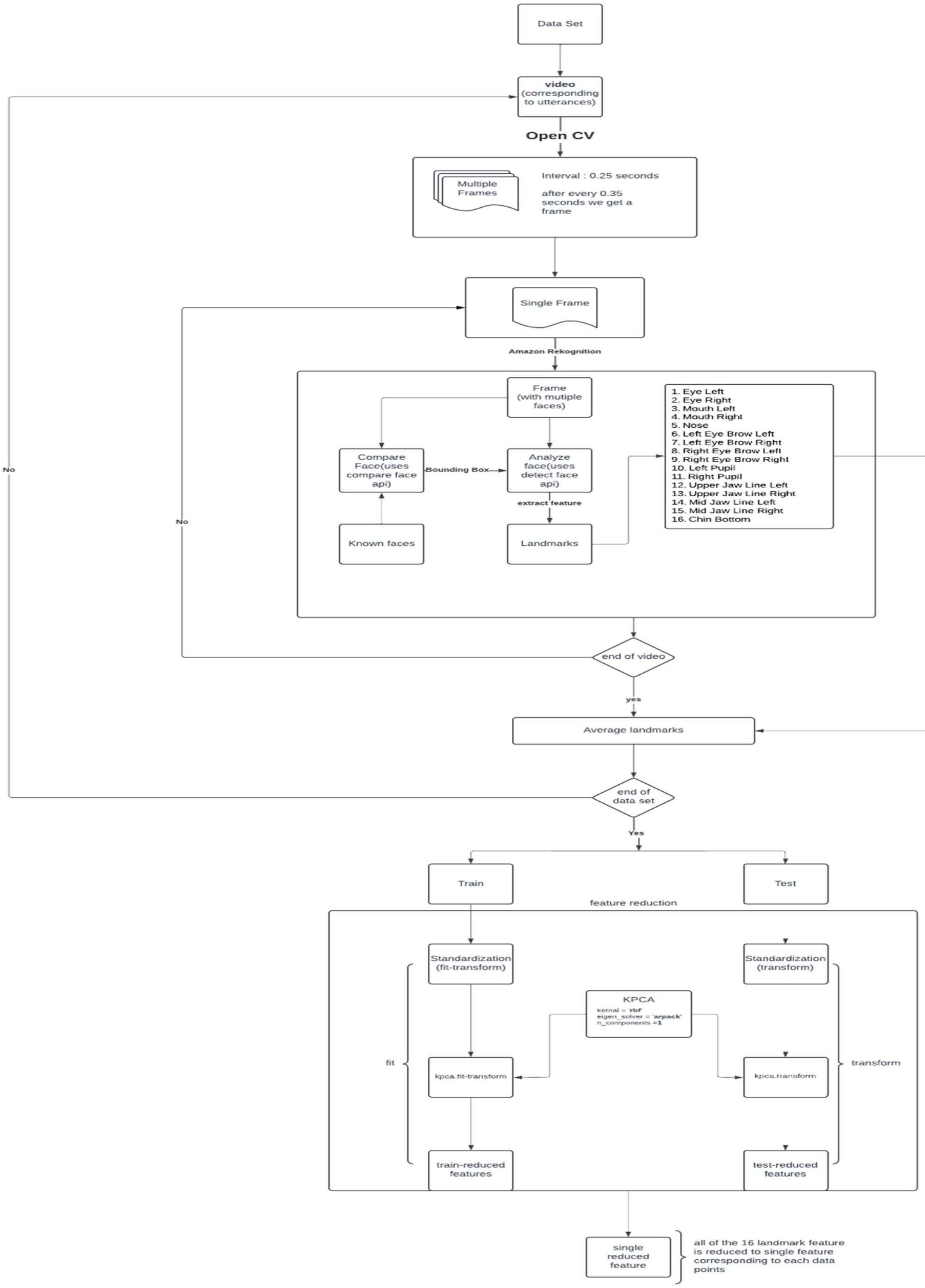


Fig : Pipeline for facial feature computation

Feature Extraction using MTCNN:

MTCNN (Multi-task Cascaded Convolutional Networks) tool is an alternative to Amazon Rekognition, which is a robust and efficient method for face detection and feature extraction. It is a serverless approach that processes images or video frames to detect faces and extract key facial landmarks. MTCNN is particularly effective in identifying faces and can be an alternative to Amazon Rekognition for certain applications.

We use OpenCV to process the video by splitting it into frames. We then employ MTCNN to detect faces within these frames. The detected faces are compared with known faces to ensure accurate speaker identification. Once the correct face is identified, we extract the facial features.

The features extracted using MTCNN are more limited compared to Amazon Rekognition, focusing primarily on key landmarks: eyes, nose, and mouth. This streamlined feature set can be sufficient for specific tasks, especially when computational resources are limited.

By incorporating MTCNN for face analysis, we offer a serverless, lightweight alternative to Amazon Rekognition, which can be advantageous in certain scenarios. The identified landmarks (eyes, nose, mouth) are then used for further processing and analysis.

2. Audio Analysis

In our project, we employed a two-step approach for audio feature extraction and reduction to enhance the dataset used for emotion prediction.

Audio Feature Extraction:

To extract audio features from video files, we used the Python library ***`librosa`***. The process began with extracting the audio from each video file using the ***`moviepy`*** library. The extracted audio was then processed to obtain various features:

1. **MFCCs (Mel Frequency Cepstral Coefficients)**: MFCCs capture the timbral aspects of the audio signal and are crucial for distinguishing different sounds and speech patterns. We computed the mean value of the MFCCs to reduce dimensionality.
2. **Mel Spectrogram**: The Mel spectrogram represents the power spectrum of the audio signal mapped onto the Mel scale, mimicking human ear perception. We also computed the mean value of the Mel spectrogram.
3. **Spectral Contrast**: Spectral contrast measures the difference in amplitude between peaks and valleys in the sound spectrum, capturing the texture and timbral characteristics of the audio. The mean value of the spectral contrast was computed.

These features were extracted for each video in our dataset, resulting in three sets of mean features corresponding to MFCCs, Mel spectrogram, and spectral contrast.

Plots to visualise features for different emotions taken from the training data points:

Anger:

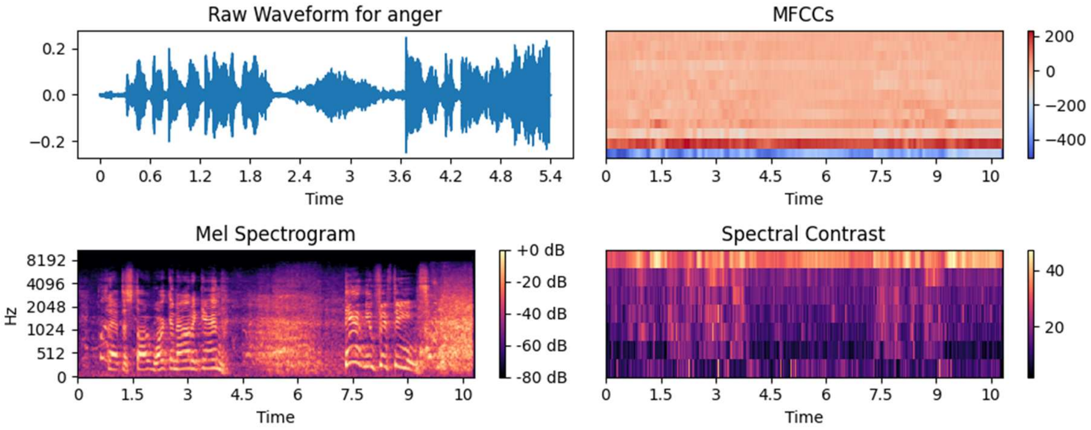


Fig : Audio sample, mfccs, mel spectrogram and spectral contrast for emotion anger

Fear:

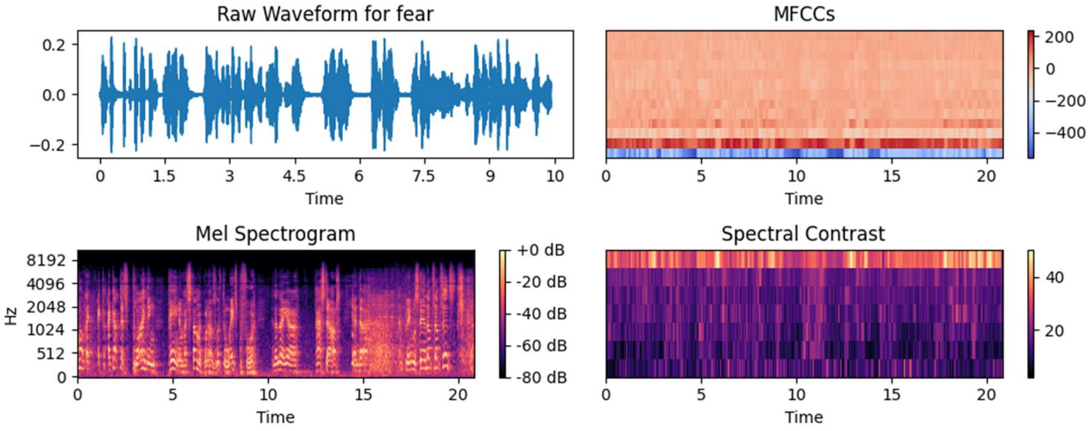


Fig : Audio sample, mfccs, mel spectrogram and spectral contrast for emotion fear

Sadness:

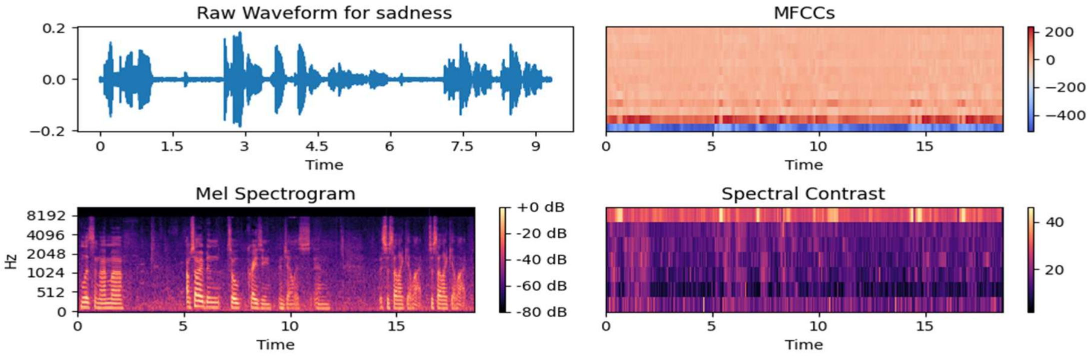


Fig : Audio sample, mfccs, mel spectrogram and spectral contrast for emotion sad

Disgust:

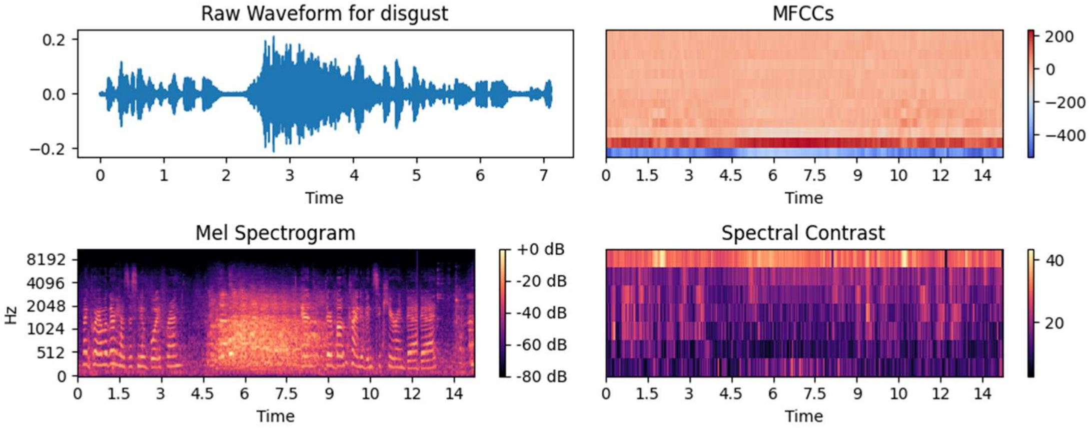


Fig : Audio sample, mfccs, mel spectrogram and spectral contrast for emotion disgust

Joy:

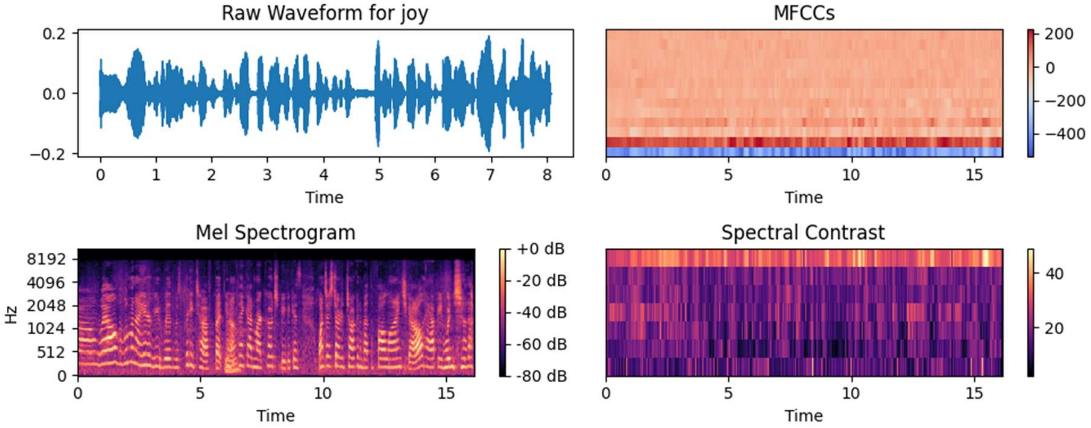


Fig : Audio sample, mfccs, mel spectrogram and spectral contrast for emotion joy

Audio Feature Reduction:

After extracting the audio features, we applied a reduction technique to consolidate these features into a single representative feature. The steps involved were:

1. **Flattening the Features**: We combined the MFCCs, Mel spectrogram, and spectral contrast features into a single flattened array for each data point.
2. **Scaling**: The combined features were scaled using `StandardScaler` to ensure they were on a comparable scale, improving the performance of the subsequent reduction method.
3. **Kernel PCA (KPCA)**: We used Kernel Principal Component Analysis (KPCA) with an RBF kernel to reduce the dimensionality of the flattened audio features. KPCA was fit on the training data and then used to transform both the training and test sets. This method ensured consistent transformation and prevented data leakage.

The result was a single reduced audio feature for each data point.

Model Building and Evaluation:

In the final phase of our project, we integrated all the extracted and reduced features into a machine learning model to predict the valence (emotional value) and arousal of each data point. This process involved several steps:

Feature Integration:

We combined the extracted and reduced facial and audio features with other relevant features, such as `Influence_0`, `Influence_1`, and `Sequence_Length`.

These features were selected based on their importance in predicting the emotional state, as they provide a comprehensive view of both visual and auditory cues, along with temporal context.

Model Selection and Training:

We chose the RandomForestClassifier from scikit-learn due to its robustness and ability to handle complex, non-linear relationships in the data. The model was configured with 100 estimators, the entropy criterion for information gain, and a maximum depth of 10 to prevent overfitting. We trained the model on the training dataset, which included the combined features and the valence labels.

Model Evaluation:

The trained model was used to predict valence on the test dataset. The performance was evaluated using the classification report and confusion matrix, which provided detailed insights into precision, recall, F1-score, and the confusion matrix for each class.

Cross-validation was performed with 5 folds on the training dataset to ensure the model's stability and generalizability. The cross-validation accuracy scores and their average were reported to provide a robust measure of model performance.

Evaluation metrics:

Precession:

It is the ratio of correctly predicted positive observations to the total predicted positives.

$$Precision = \frac{True\ Positive}{True\ Positive+Fals\ \ Positive}$$

For class 1(positive/high valence), precision indicates how many of the predicted high valence emotions were actually high valence. For class 0(negative/low valence), it indicates how many of the predicted low valence emotions were actually low valence.

Recall:

It is the ratio of correctly predicted positive observations to the all observations in actual class.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

For class 1, recall measures how many of the actual high valence instances were correctly identified by the model. For class 0, it measures how many of the actual low valence instances were correctly identified.

F1-score:

It is the harmonic mean of precision and recall. It provides a single metric that balances both concerns and is especially useful when we have an uneven class distribution.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1-score provides a balance between precision and recall for both valence classes, giving us a single measure of the model performance.

Confusion Matrix:

A confusion matrix is a table used to evaluate the performance of a classification algorithm. It visualises the performance of the model by showing the actual vs predicted classification.

True Positives(TP) : The number of times the model correctly predicted class 1 or high valence

True Negative(TN) : The number of times the model correctly predicted class 0 or low valence

False Positives(FP) : The number of times the model incorrectly predicted class 1 when it was actually class 0.

False Negative(FN) : The number of times the model incorrectly predicted class 0 when it was actually class 1.

.

Cross-validation Accuracy:

It is the average accuracy of the model over multiple folds of the data, providing an estimate of model performance that is less biassed by the training data. It provides a robust estimate of how well your model is likely to perform on unseen data, ensuring that the model generalises well beyond the training data.

Results and Discussion :

The classification report and confusion matrix provided a comprehensive evaluation of the model's predictive performance on unseen test data, highlighting areas of strength and potential improvement.

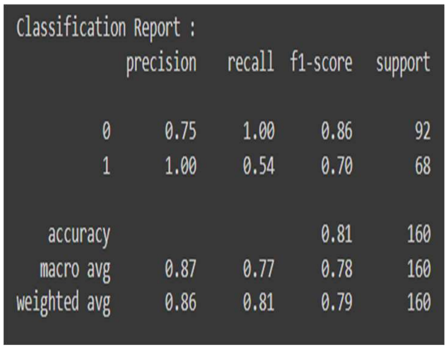
Below is the detailed result of the **speaker Ross** with 637 data points in train and 160 data points in test.

Out of the 637 data points in the train, 248 data points are with high valence or class 1 and 389 data points are with low valence or class 0.

Out of the 160 data points in the test, 68 data points are with high valence or class 1 and 92 data points are with low valence or class 0.

The performance of the model was evaluated at different stages by progressively including more features:

Self-reported features :



```
Classification Report :
              precision    recall  f1-score   support

           0       0.75      1.00      0.86        92
           1       1.00      0.54      0.70        68

    accuracy                           0.81       160
   macro avg       0.87      0.77      0.78       160
weighted avg       0.86      0.81      0.79       160
```

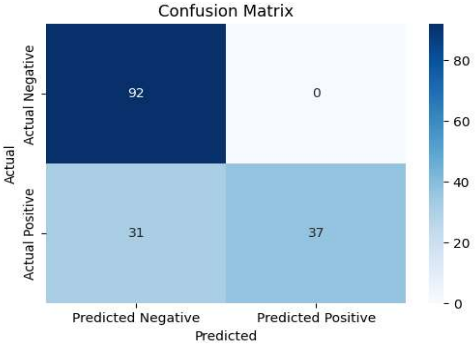
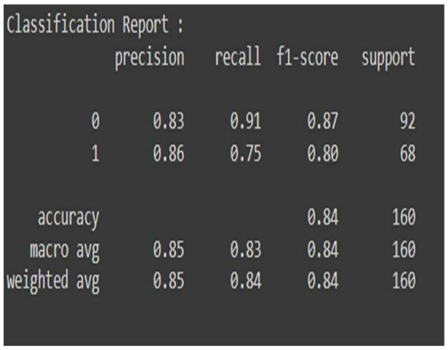


Fig : Classification report for self-reported features

The model shows a perfect recall for class 0, indicating it identifies all instances of class 0 correctly. Precision and F1-score for class 1 are high,but recall is relatively low. The accuracy is decent,but the imbalance in recall between the classes suggests the model struggles with high valence.

Self-reported features and facial feature :



```
Classification Report :
              precision    recall  f1-score   support

           0       0.83      0.91      0.87        92
           1       0.86      0.75      0.80        68

    accuracy                           0.84       160
   macro avg       0.85      0.83      0.84       160
weighted avg       0.85      0.84      0.84       160
```

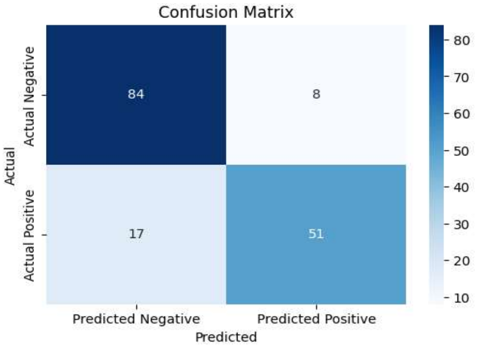
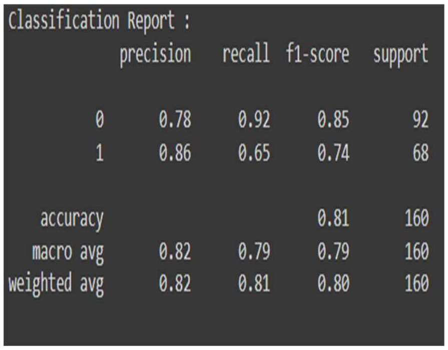


Fig : Classification report for self-reported and facial features

Incorporating facial features improves precision, recall, and F1-score for both classes.The model shows better balance between precision and recall for both classes. Overall accuracy improves to 84%, indicating a better general performance with facial features included.

Self-reported features and audio feature :



```
Classification Report :
              precision    recall  f1-score   support

           0       0.78      0.92      0.85        92
           1       0.86      0.65      0.74        68

    accuracy                           0.81       160
   macro avg       0.82      0.79      0.79       160
weighted avg       0.82      0.81      0.80       160
```

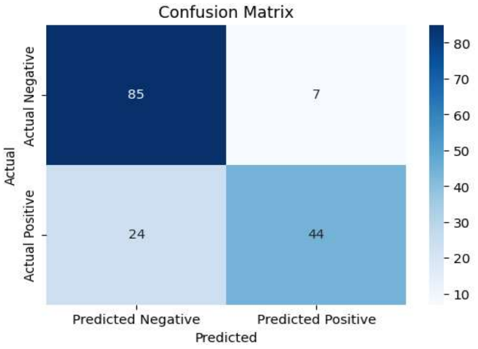
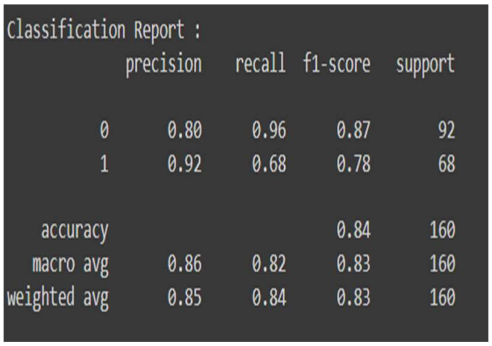


Fig : Classification report for self-reported and audio features

Adding audio features improves recall for class 0 but reduces recall for class 1 compared to using facial features. Precision for class 1 remains high, but the F1-score is lower than the model with facial features. The overall accuracy is 81%.

Self-reported features, facial-feature and audio-feature :



```
Classification Report :
              precision    recall  f1-score   support

           0       0.80      0.96      0.87        92
           1       0.92      0.68      0.78        68

    accuracy                           0.84       160
   macro avg       0.86      0.82      0.83       160
weighted avg       0.85      0.84      0.83       160
```

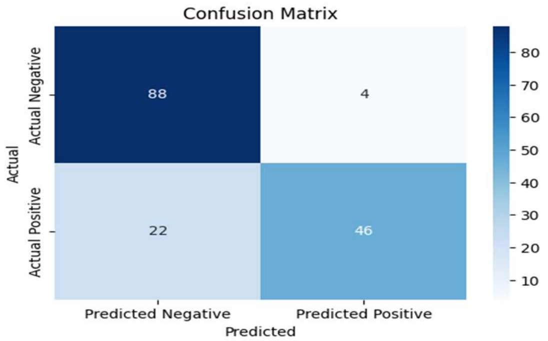
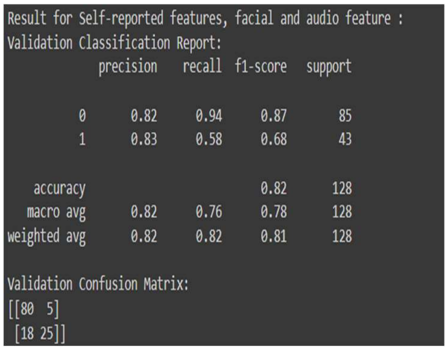


Fig : Classification report for self-reported, facial and audio features

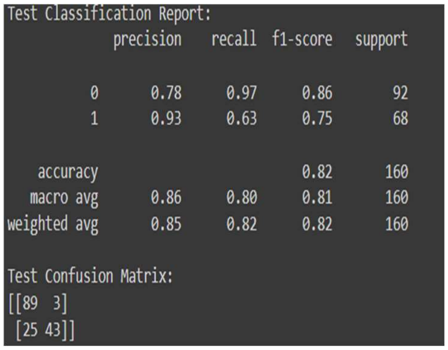
Combining all features(self-reported, facial and audio) results in the most balanced model with the highest precision for class 1 among all configurations. The overall accuracy, precision,recall and F1-scores are all strong, indicating that the model benefits from using the full set of features

By integrating self-reported, facial, and audio features, the model demonstrates improved and balanced performance. The addition of facial features significantly enhances the model's ability to predict both classes. Combining all features results in the highest precision and recall, indicating the effectiveness of a multimodal approach in emotion prediction tasks.

We have also performed cross validation by splitting the training set into a train set and a validation set, then assessed the model's performance on the unseen test set. This approach helps in identifying potential overfitting or underfitting.



```
Result for Self-reported features, facial and audio feature :
Validation Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.94      0.87        85
           1       0.83      0.58      0.68        43

    accuracy                           0.82       128
   macro avg       0.82      0.76      0.78       128
weighted avg       0.82      0.82      0.81       128

Validation Confusion Matrix:
[[80  5]
 [18 25]]
```

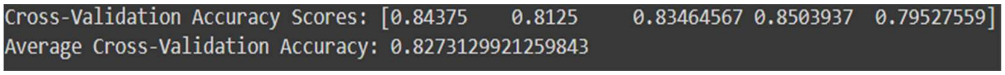


```
Test Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.97      0.86        92
           1       0.93      0.63      0.75        68

    accuracy                           0.82       160
   macro avg       0.86      0.80      0.81       160
weighted avg       0.85      0.82      0.82       160

Test Confusion Matrix:
[[89  3]
 [25 43]]
```

Fig : Classification report for validation and test set

The model's performance was evaluated using validation and test datasets, as well as cross-validation techniques. The results indicate that the model is neither overfitting nor underfitting, as evidenced by the following points:

- The validation accuracy (0.82) is consistent with the test accuracy (0.82), demonstrating that the model generalises well to unseen data.
- The precision, recall, and F1-scores for both classes are stable across validation and test sets, suggesting reliable performance in classifying both classes.



```
Cross-Validation Accuracy Scores: [0.84375    0.8125     0.83464567 0.8503937  0.79527559]
Average Cross-Validation Accuracy: 0.8273129921259843
```

Fig : Averaged cross validation accuracy.

- Cross-validation accuracy scores are consistently high, with an average of 0.8273, further supporting the model's robustness and generalisation capability.

In summary, the model exhibits stable and reliable performance across different evaluation metrics, indicating a well-balanced approach to learning from the training data without overfitting or underfitting.

Tools and Technologies used:

- Python
- OpenCV
- AWS Rekognition
- Librosa
- Kernel Principal Component Analysis

Conclusion :

This project has been a comprehensive exploration of developing a contextual multimodal emotion dataset. By integrating facial analysis through AWS Rekognition and audio feature extraction using librosa, we have built a robust pipeline for generating dataset reflecting the complexities of human emotions. The incorporation of self-report data to map emotions with valence and arousal, along with influence and emotion sequence length calculations, has enhanced the accuracy of emotion prediction models.

The project has provided valuable insights into machine learning, data analysis, and emotion detection technologies. The guidance and support received were instrumental in achieving the project's objectives.

References :

1.detect_faces api - facial recognition software :
https://aws.amazon.com/rekognition/?nc=sn&loc=0

2. SELFI: Evaluation of Techniques to Reduce Self-report Fatigue by Using Facial Recognition of Emotion - research paper.

3. https://www.kaggle.com/code/shivamburnwal/speech-emotion-recognition