Μάθημα: Μ906- Προγραμματισμός για Γλωσσική Τεχνολογία ΙΙ (Εαρινό Εξάμηνο 2023)
Όνομα: Κλεοπάτρα Καραπαναγιώτου
Α.Μ: 7115182200010
Τμήμα: Πληροφορικής και Τηλεπικοινωνιών (ΔΠΜΣ Γλωσσική Τεχνολογία)

Final Project Sound

Introduction

Emotion Recognition/Classification

The overall objective of an emotion classifier is to bridge the gap between human emotional expressions and computational understanding. By accurately recognizing and categorizing emotions, these classifiers enable machines to perceive, interpret, and respond to human emotions, leading to improved interaction and communication between humans and technology.

Emotion classifiers are utilized in various domains, including affective computing, human-computer interaction, virtual agents, sentiment analysis, personalized recommendation systems, and healthcare. They enable systems to adapt and respond to human emotions, enhance user experiences, and provide personalized services based on emotional states.

Some specific applications and fields of innovation in which emotion classification can play a leading role are:

1. **Healthcare** (Mental Health Research, Autism Support, Telemedicine)
2. **Social Robotics** (Retail Assistance, Healthcare companion, recommender systems: recommending products/services based on the user's mood)
3. **Education** (Self-paced learning Insight, Adaptive Learning content)
4. **Media and Advertising** (Ad Content optimization based on consumer's emotional engagement, TV Character Analysis: Determining if new characters on a show build rapport with viewers and if the interplay is successful)

Emotion classification does not solve a problem in a typical sense, but rather reveals problems we didn't realize they exist.

Task Description

Construct and evaluate a real-time audio processing system that recognizes and classifies two emotions (calm/angry) both from audio files and from the microphone.

Structure

The zip file contains the following .py files, which should be run in the following order.

visualisations.py: Contains the visualization of several spectral features, to decide, which features separate our two emotional classes (act_01= 'calm', act_02='angry') the best possible way.

audio_representation.py: To initiate the AudioRepresentation() class, which will enable the extraction of the useful features from the area that contains sound and ignores the silence. In our case, the useful features are the MFCCs, the effect of which will analyzed later in detail.

0_train_model_and_save.py: We are experimenting with several machine learning models, which are known for their good performance in binary classification tasks. We print the mean, std and median accuracy in 10-fold cross validation for each of the chosen models and based on the results, we choose and save the model that gave us the best mean accuracy. In our case it was the Random Forest classifier.

1_file_test_realtime_processing.py: This code reads an audio file, processes it in audio blocks (only the last 20), extracts MFCC features, and uses the trained Random Forest model to classify the emotions present in the audio. The classification results are displayed, plotted, and a classification report with the results for each of the 20 audio blocks is generated for evaluation.

2_mic_realtime_processing.py: This code follows the exact same process with the previous one. The only difference is that the input received real time is from the microphone.

majority_class_detection.py: Code to discern the majority class for each file and microphone session based on the classification results found in the folder **Classification_results.**

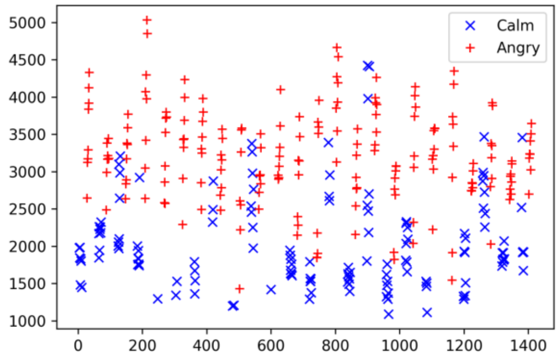
Implementation

The source of the data used for this project is the Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess), which can be found in Kaggle under the title "RAVDESS Emotional Speech Audio"[1]

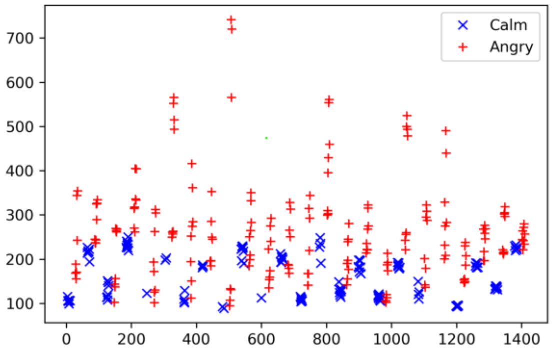
Data Pre-processing

The data were loaded from the "prepared_dataframe" pickle file which contains all the data from all speakers. For our task we did not perform an initial discrimination of the files based on gender, but rather only based on emotion, so only the files denoting calm or angry emotion were kept, irrespective of the gender class.

To clarify which features among centroid, mfcc and f0 will contribute to our classification the best, we generated their plots:

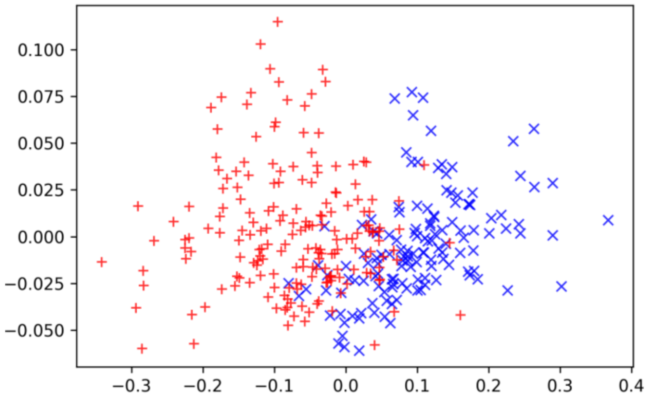


Mean_centroid



Mean_f0

[1] https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio



MFCCs in PCA

Features

MFCC (Mel frequency cepstral coeficients)

From the above plots we observe that the MFCCS and the mean centroid give us a nearly clear separation of the two classes.

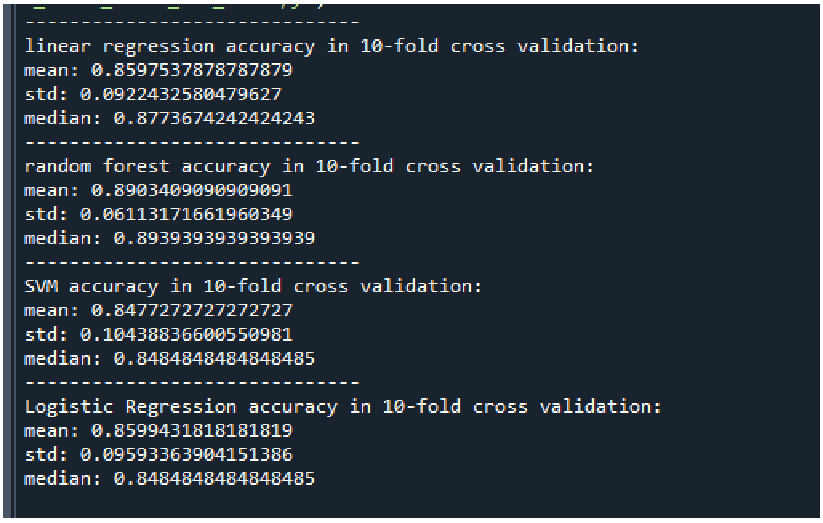
The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies.

On the other hand, the spectral centroid represents the center of gravity of the scale invariant Fourier transform (SIFT) spectrum and provides the spectral shape characteristic of the speech signal.

We will continue with the MFCCs for this task, because incorporating the Mel scale makes our features match more closely what humans hear and perceive.

Results

Classifier - Training



```
-----------------------------
linear regression accuracy in 10-fold cross validation:
mean: 0.8597537878787879
std: 0.0922432580479627
median: 0.8773674242424243
-----------------------------
random forest accuracy in 10-fold cross validation:
mean: 0.8903409090909091
std: 0.06113171661960349
median: 0.8939393939393939
-----------------------------
SVM accuracy in 10-fold cross validation:
mean: 0.8477272727272727
std: 0.10438836600550981
median: 0.8484848484848485
-----------------------------
Logistic Regression accuracy in 10-fold cross validation:
mean: 0.8599431818181819
std: 0.09593363904151386
median: 0.8484848484848485
```

Based on the above results we choose Random Forest as our classifier for this task.

Realtime evaluation (files)

For the real-time evaluation on files, we chose 8 random files from our training set, 4 for each gender, 2 per emotional class, 1 per emotional intensity. In the folder with name **“files_results**”, we can inspect the screen recording and the classification results for each chosen .wav file.

Majority Class Detection for each file

Emotion Class(intensity)/ Gender	**Calm (low)**	**Calm (high)**	**Angry (low)**	**Angry (high)**
Male	Angry (21.05%) **Calm (78.95%)**	Angry (40.00 %) **Calm (60.00%)**	Angry (50.00%) Calm (50.00%)	**Angry (66.67%)** Calm (33.33%)
Female	Angry (31.82%) **Calm (68.18%)**	Angry (1/20) **Calm (95.00%)**	**Angry (52.38%)** Calm (47.62%)	**Angry (68.42%)** Calm (31.58%)

From the results above several interesting information can be derived:

- Calmness, expressed in low emotional intensity is sufficiently distinguishable for both genders and especially for males.

- Calmness expressed in low emotional intensity is almost absolutely distinguishable in females, whereas less, but still sufficiently distinguishable for males.
- Anger expressed in low emotional intensity seems to be hardly distinguishable for both genders.
- Anger expressed in high emotional intensity seems identically distinguishable in both genders.

Realtime evaluation (microphone)

For the evaluation on mic 2 recordings, 1 per emotional class (irrespective of emotional intensity), were generated.

Majority Class Detection for each microphone session

	Calm	**Angry**
Mic(me=Female)	Angry (22.22%) **Calm (77.78%)**	**Angry (51.61%)** Calm (48.39%)

The results of the real-time evaluation in microphone seem compatible with results of the evaluation on files, and especially the female recordings. The predictions of the majority class are compatible with the true label-class for both emotional classes. The struggle in discerning anger, though, is also present in the real-time evaluation through microphone, but it is not clear if this time it's the emotional intensity causing this vagueness or other factors such as longer pauses, which are common in spontaneous speech.

Conclusions, further thoughts

Overall, the results of our trained classifier generated accurate results in both lab- and real-life conditions, excluding the emotion of anger when expressed with low emotional intensity. All results in the majority class detection are compatible with the true emotional class of the file and microphone session accordingly. Nevertheless, in order to address the above shortcomings and optimize the classifier's performance several steps can be taken in terms of data preprocessing.

Feature Engineering

- **Combining MFCCs with other informative features**

As indicated above, not only MFCCs make our 2 classes separable, but also the centroid. Further features that would be interesting to involve are the pitch contour in terms of the sound quality and the speaking rate in terms of duration.

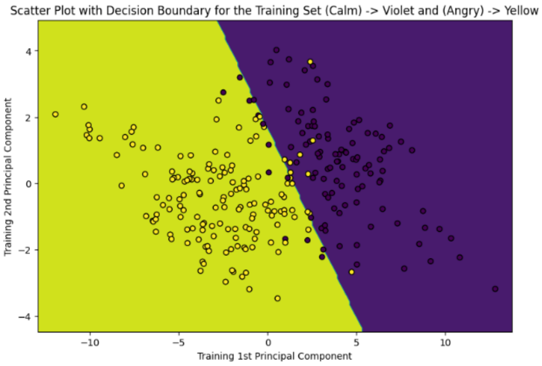
- **Feature Scaling**

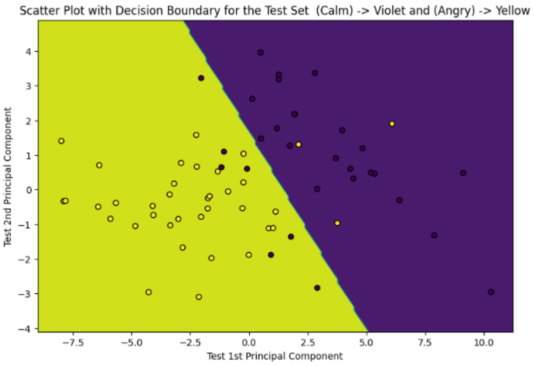
Combining MFCCs with other features, which have numerical values of different scale, lead to a biased model favoring the higher numerical values over the lower ones. That's why creating common range features with feature scaling is crucial before training the classifier.

- **Dimensionality Reduction (PCA)**
  Finding those features that contribute the most to our classification and being able to plot visualize them is also an important step in the classifier optimization. Nevertheless, reducing dimensions requires making adjustments in the real time visualization of the mfccs, since our code is expecting 20 features to work, whereas reducing dimensions to 2 lead to an error.

Hyperparameter tuning in Machine learning algorithms

In the file named **experiment_train_model_feature_extration_scale_pca_logistic.py** we extract not only MFCCs, but also mean +std centroid and mean+std bandwidth. We apply feature scaling and reduce dimensions to 2. We train a logistic regression classifier to which we apply 5-fold Grid-Search cross validation to fine-tune the regularization parameter C and then after finding the best C we re-train our classifier and plot the results. It was clear from the accuracy results, that hyperparameter optimized the performance of our classifier, reaching a mean accuracy in 10-fold of over 93%! The classifier's performance in both training and test set can be examined in the plotted decision boundary below.





Lab evaluation

For the lab conditions it would be interesting to experiment with noisy files of different SNR (Signal to Noise Ratio) e.g. 3,6,9 dB, in order to compare their results and decide, which SNR affects the performance of our classifier the least.

Mic-evaluation

During the mic session it was observed that speakers tended to make longer pauses, which potentially lead to less correct predictions. We could apply pause-trimming techniques or reduce the audio blocks kept for evaluation from 20 to 10, to enforce the classifier to generate less classification results.

References

Kaggle Notebooks
https://www.kaggle.com/code/shivamburnwal/speech-emotion-recognition
https://mikesmales.medium.com/sound-classification-using-deep-learning-8bc2aa1990b7

Gitlab/Github Notebooks

https://gitlab.com/language-technology-msc/programming-for-language-technology-ii-2022-2023/programminglangtechii_c03/-/blob/main/4_run_ML_tests.py

https://gitlab.com/language-technology-msc/programming-for-language-technology-ii-2022-2023/programminglangtechii_c05/-/blob/main/1a_file_playback.py

https://gitlab.com/language-technology-msc/programming-for-language-technology-ii-2022-2023/programminglangtechii_c05/-/blob/main/5_mic_spetrogram.py

https://github.com/vibhash11/Emotion-Recognition-Through-Speech

Dataset

"The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)" by Livingstone & Russo is licensed under CC BY-NA-SC 4.0.