ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

ATHENA Research & Innovation Information Technologies

M906 Programming for NLP II

Final Project - Sound

Report

Christina-Theano (Theatina) – LT1200012

Zafeiri Anthi – 7115182100007

Table of Contents

Introduction

Task Description

In this assignment, the separability of the emotions **calm / angry** is being researched. Multiple classifiers are trained and evaluated, the scores of which are later presented. One of those classifiers is selected for further **evaluation** in real time setting by using **audio files**. Also, while testing the classifier by using the **microphone input**, the **MFCC** features of a window are depicted as well as the model's prediction on the **emotion** of the speaker.

Structure

The .zip file attached, includes also the directories and files :

1. **Code**/ : the python files (.py) that contain the code for the questions which are the reference files modified for the task(s)
    a. **Q1_train**.py : classifier experiments, training and cross validation
    b. **Q2_evaluate_RT_files**.py : real time file classifier evaluation supported by audio window MFCC features visualization
    c. **Q2_evaluate_RT_EvalNFiles**.py : same functionality as b, with no visualization
    d. **Q2_evaluate_RT_N_files**.sh : bash script for classifier evaluation on N files using the Q2_evaluate_RT_EvalNFiles.py file
    e. **Q3_evaluate_RT_mic**.py : real time microphone input classifier evaluation supported by the respective audio MFCC features visualization
    f. **functions**.py : functions used by the files above for cleaner code practices
    g. **modelEval_results**.txt : results of classifier experiments from file Q1_train.py
    h. audio file representation reference .py files

2. **Data**/ : the dataframe with the necessary pre-processed data
    a. **AudioFiles**/ : the audio files from which N are selected to be used for evaluation

3. **Models**/ : models of trained classification algorithms

4. **Results**/ : Q1, Q2 & Q3 results such as logfiles, evaluation classification reports & real time model testing screenshots / videos

Syntax

In order to produce any of the questions results, the syntax to be followed is:

1. **Q1_train**.py < classification algorithm > (default: “LogReg”)

   Classification algorithms:

   a. “**HistGB**” : HistGradientBoostingClassifier ( )
   b. “**SVM**” : SVC (C=1.0, kernel="rbf")
   c. “**RF**” : RandomForestClassifier (n_estimators=150, warm_start=True)
   d. “**LR**” : LinearRegression ( )
   e. “**KNN**” : KNeighborsClassifier (n_neighbors=9, weights="distance")
   f. “**LogReg**” : LogisticRegression (C=1000.0, solver="liblinear", penalty="l2", max_iter=1000)

   In the following .py files, the default model which is used and evaluated is “Models/**LogReg_CalmAngry**.model”, the logistic regression classifier.

2. **Q2_evaluate_RT_files**.py < audio file path >

3. **Q2_evaluate_RT_N_files**.sh < evaluation audio files directory > < number of files >

4. **Q3_evaluate_RT_mic**.py

Implementation

Data Pre-Processing

Data processing before training is automated and performed by previous steps not involved in the project at hand. As it is discussed in question 4, suggestions and further steps are proposed aiming to optimize the classifier's performance.

Data Scaling

Some classifiers performed better with the addition of scalers. More specifically, the classes **MinMaxScaler** and **StandardScaler** have been used in the experiments.

MinMaxScaler transforms features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. (Unit variance means dividing all the values by the standard deviation.)

Features

The features used for the audio representation were the MFCCs (n=20).

MFCC

Mel **F**requency **C**epstral **C**oefficients (here n=20 – librosa default) are coefficients that collectively make up an MFC (**M**el-**F**requency **C**epstrum), which is a representation of the short-term power spectrum of a sound. MFCCs are commonly used as features in speech recognition systems as well as in MIR (**M**usic **I**nformation **R**etrieval) applications such as genre classification.

Results

Below, the tables and figures depict the results of the process pipeline, which comprises classifier **training**, **validation** (**10 Fold** Cross Validation), real time **testing** and real time **evaluation**. The performance is measured in **mean**, **median** and **std** of the classifiers' score values in 10 Fold CV.

Q1: Classifier & Training

In this question, experiments on the classifier architecture were carried out such as the model type, manual parameter fine tuning and data scaling.

The scores of the training process are shown below :

Classifier Type	**Mean**	**STD**	**Median**
Histogram-Based Gradient Boosting	90.82 %	**4.41** %	**90.79** %
C-Support Vector	89.48 %	7.54 %	92.11 %
Random forest	88.72 %	6.33 %	89.47 %
Linear Regression	88.42 %	8.17 %	88.16 %
k Nearest Neighbors	90.55 %	7.10 %	89.61 %
Logistic Regression	**91.10** %	**5.00** %	**90.79** %

Logistic Regression model, outperforms the rest of the classifier types, fine-tuned in the specific settings (in Q1_train.py file) for the emotion separability task.

Although the Histogram-Based Gradient Boosting classification algorithm has also output a good scores std and median, logistic regression was selected for having scored higher in the mean scoring metric.

Q2: Real Time Evaluation (files)

The evaluation was performed on the audio files' windows and not on the whole file, for more detailed scores.

2 Files	**Calm** (class 0)	**Angry** (class 1)	**Macro Avg**
Precision	.61	**.81**	.71
Recall	**.88**	.48	.68
F_1	.72	.61	.66
Accuracy			**.67**

51 Files	**Calm** (class 0)	**Angry** (class 1)	**Macro Avg**
Precision	.69	**.88**	.78
Recall	**.91**	.62	.76
F_1	.78	.73	.76
Accuracy			**.76**

67 Files	**Calm** (class 0)	**Angry** (class 1)	**Macro Avg**
Precision	.66	**.88**	.77
Recall	**.90**	.63	.76
F_1	.76	.73	.75
Accuracy			**.75**

"Calm" class shows better recall than class angry, while the latter outputs a higher precision score than the first. This could be a seemingly sensitivity **bias** over label "calm" created due to class 0 (calm) been used as the **default** class (such as for **pauses** or **low decibel parts** in the beginning / end / in-between the utterances – even the angry ones)

Also, starting from a low number of evaluation files to a higher one, we can conclude that there is an **accuracy** and **macro**-scoring convergence to ~ **75** % .

Q3: Real Time Testing (microphone)

Figures of **real time classifier testing** while speaking **calmly** and **angrily** respectively:

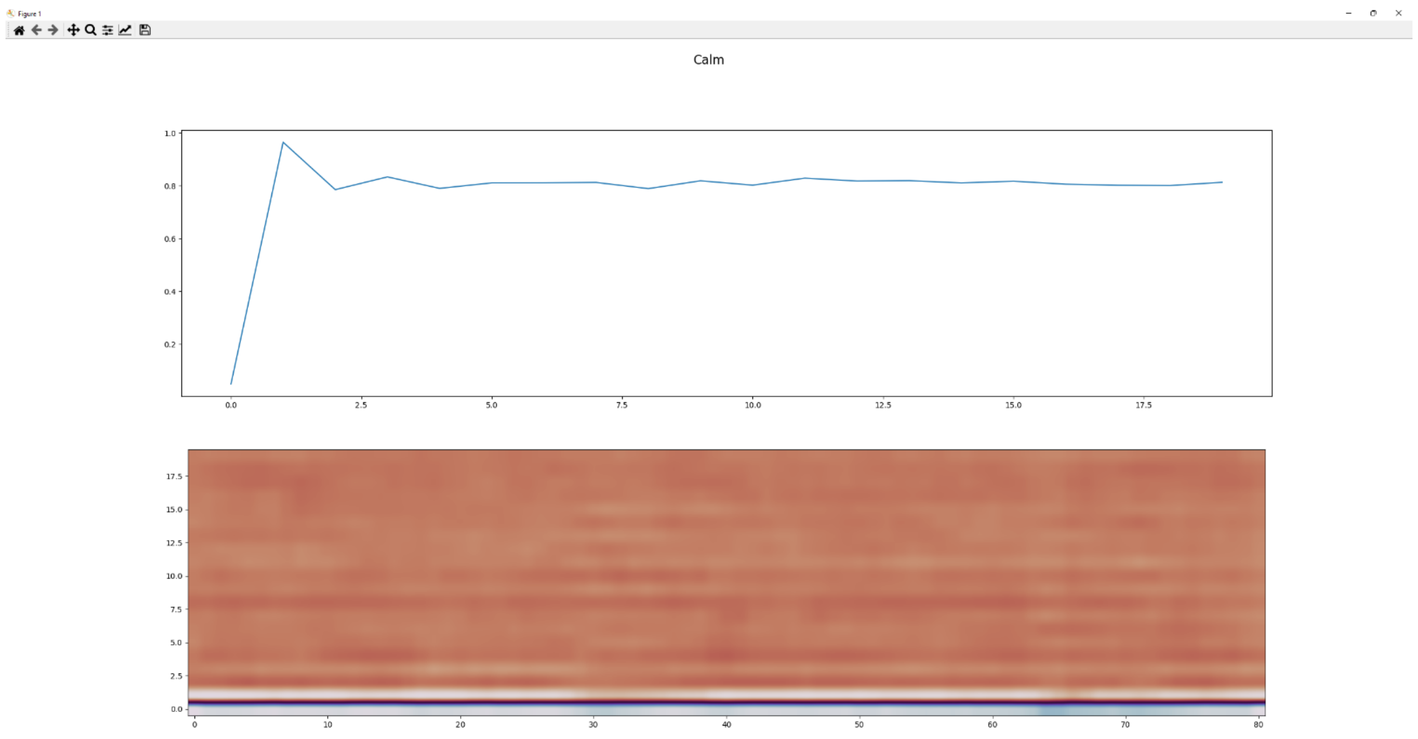

Figure 1. MFCCs of a **calm** utterance

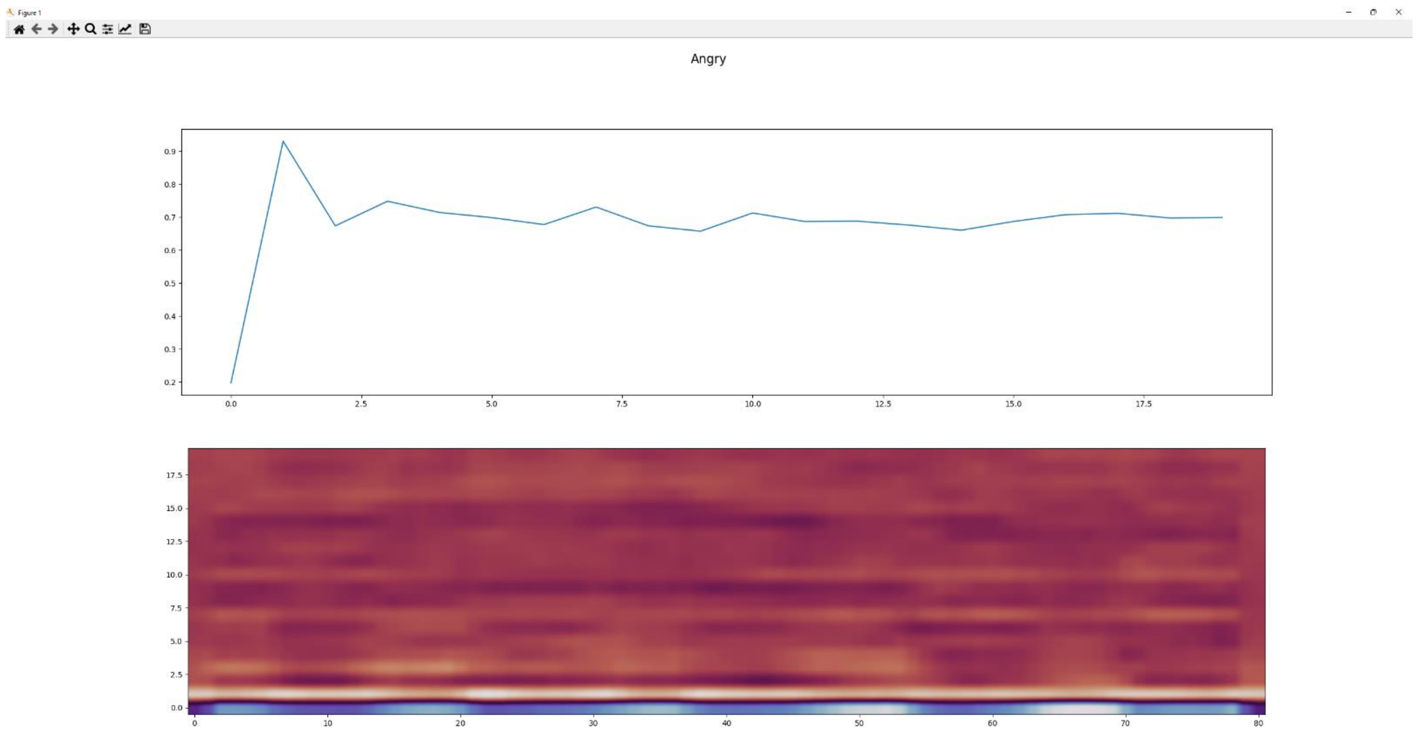

Figure 2. MFCCs of an **angry** utterance

Q4: Discussion

Multiple additional steps could be followed in order to experiment further, aiming at the classifier’s performance optimization.

Data pre-processing could be enriched with more data scaling (e.g. normalization, regularization) tests, data cleaning techniques, sound editing, audio files pause trimming, noise removal, decibel restoration, etc.

More data could be created in lab conditions (no noise, clear, equal decibel scale fluctuating only depending on emotion, etc.)

Feature extraction techniques could also be applied on the data to investigate their contribution to the classification task, such as dimensionality reduction on audio file features, spectral feature extraction (spectrogram, centroid, bandwidth), root mean square, band energy ratio, chroma and tempogram.

Finally, as far as the label “calm” sensitivity is concerned, a neutral label could be added to declare the pause / low db regions as mentioned in question 2 above, the predictions could be trimmed in those specific regions or the regions could have been trimmed in previous data processing steps proposed above.