# Building a speech emotion recognition model

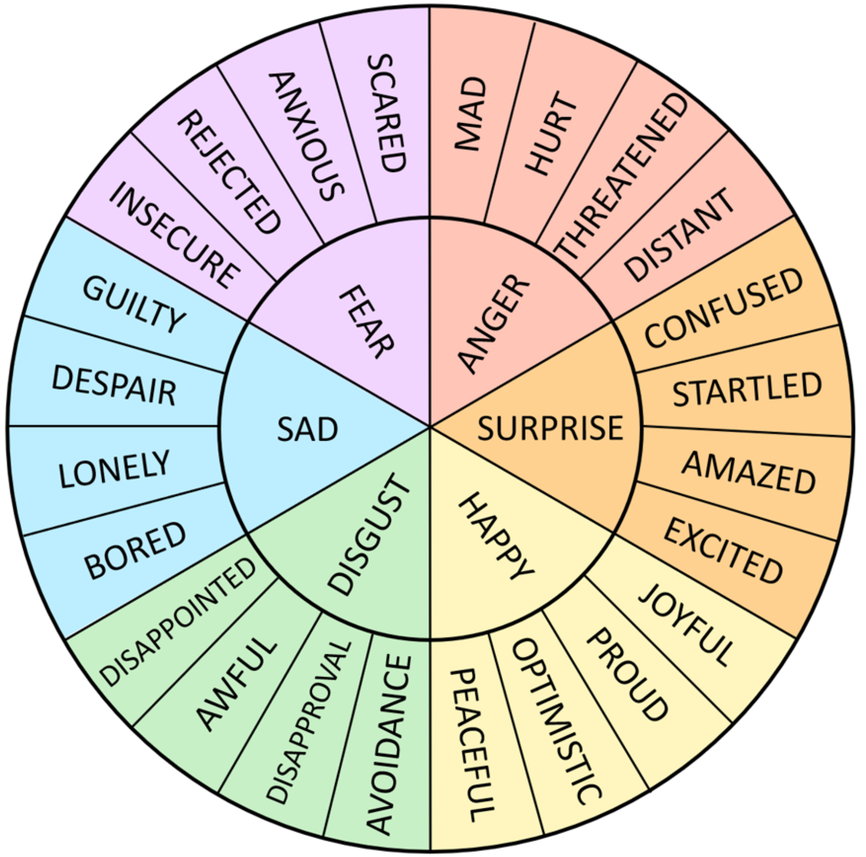
## Capstone Project

## EPFL Extension School - Applied Data Science: Machine Learning

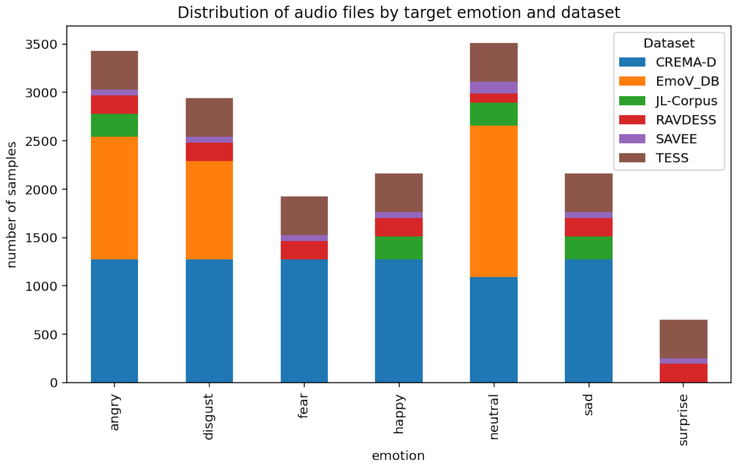
**Alessandro Zanette**

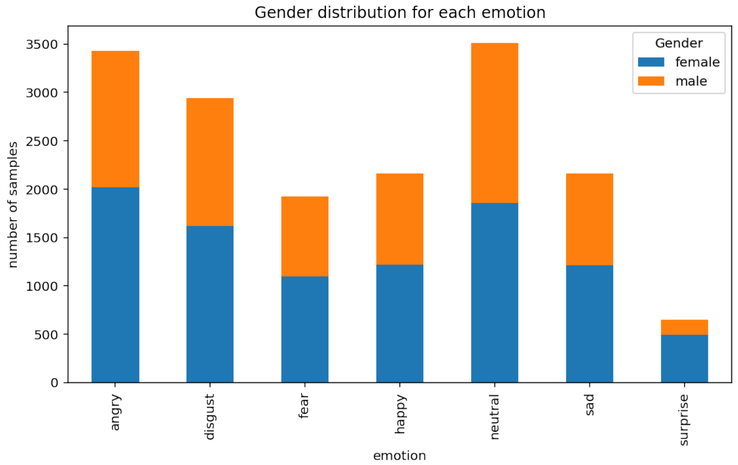
# The data

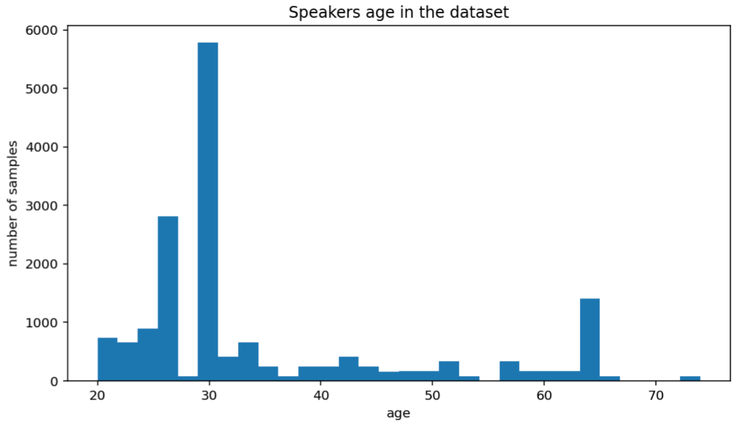
- Gathered few available datasets on emotions: CREMA-D, RAVDESS, SAVEE, TESS, EmoV-DB, JL-Corpus
- ~ 17000 audio samples in total
- Choice to focus in the classification of six cardinal emotions: anger, disgust, fear, happiness, sad, surprise
- Adding the “neutral” feeling, so a total of 7 targets

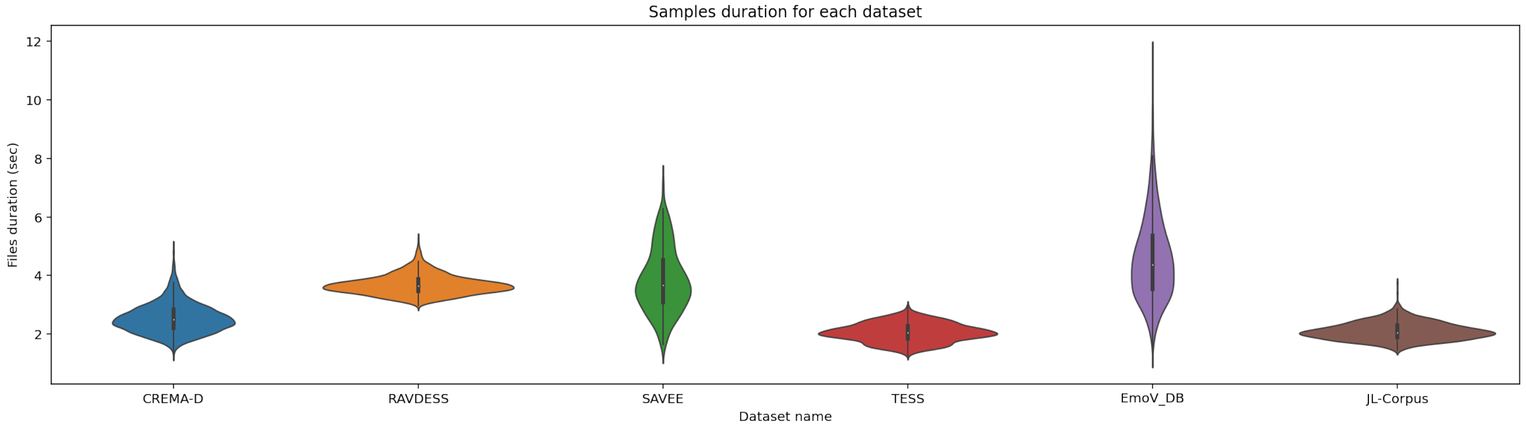


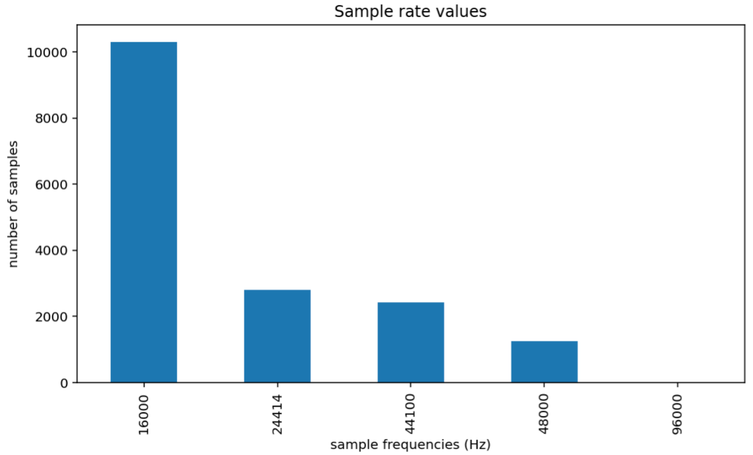
# Ensuring data compatibility



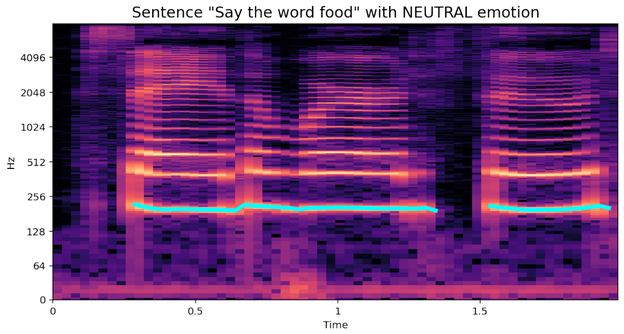



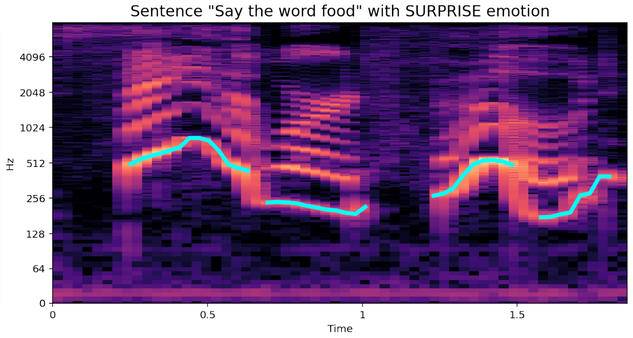



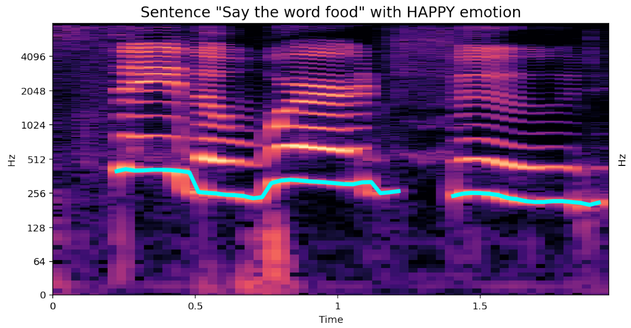



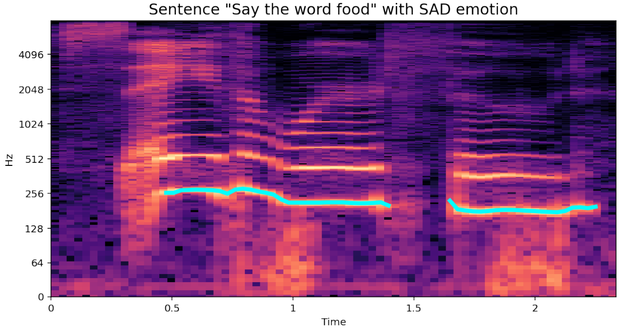
# First intuition

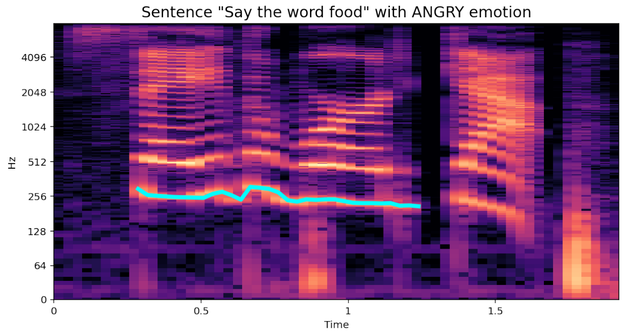
## Comparing the same sentence from the same speaker with different emotions



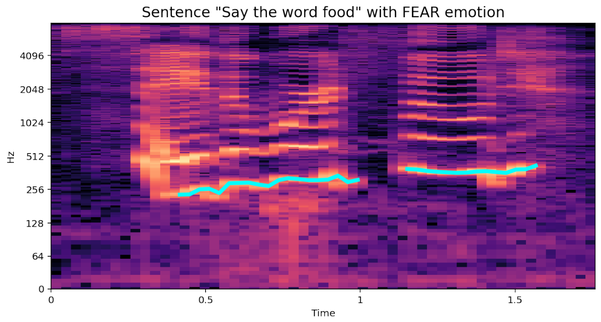







# Cleaning data

- Resample all files to 16000 Hz
- Trim the initial/ending silences
- Reduce stationary noise (-10%)
- Save cleaned samples in a new folder

# Feature extraction

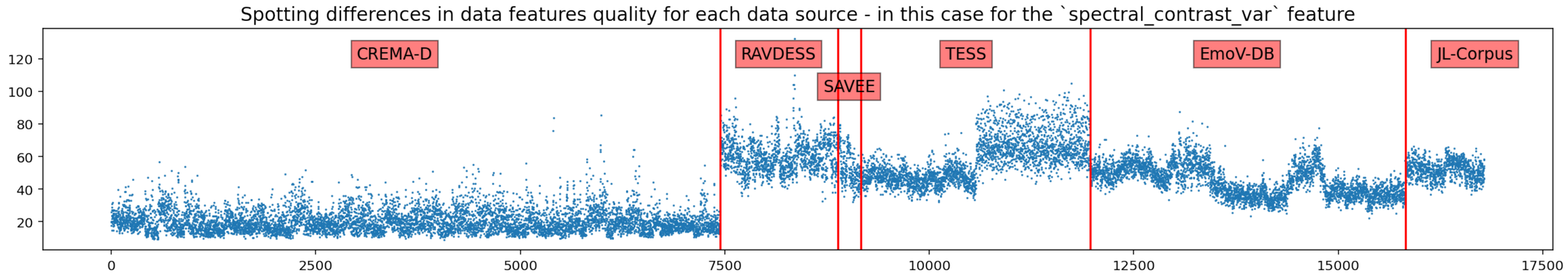
## *librosa*: a python package for music and audio analysis

- Fundamental frequencies (f0 mean, std, …)
- Zero-crossing rate
- Spectral centroid
- Spectral contrast
- Spectral flatness
- Harmony
- Root-mean-square (rms)
- Chroma feature
- Chroma cqt
- Chroms cens
- Rolloff
- Mel-Frequency Cepstrum Components (MFCCs): mean and variance of 30 extracted components

# Exploratory Data Analysis (EDA)

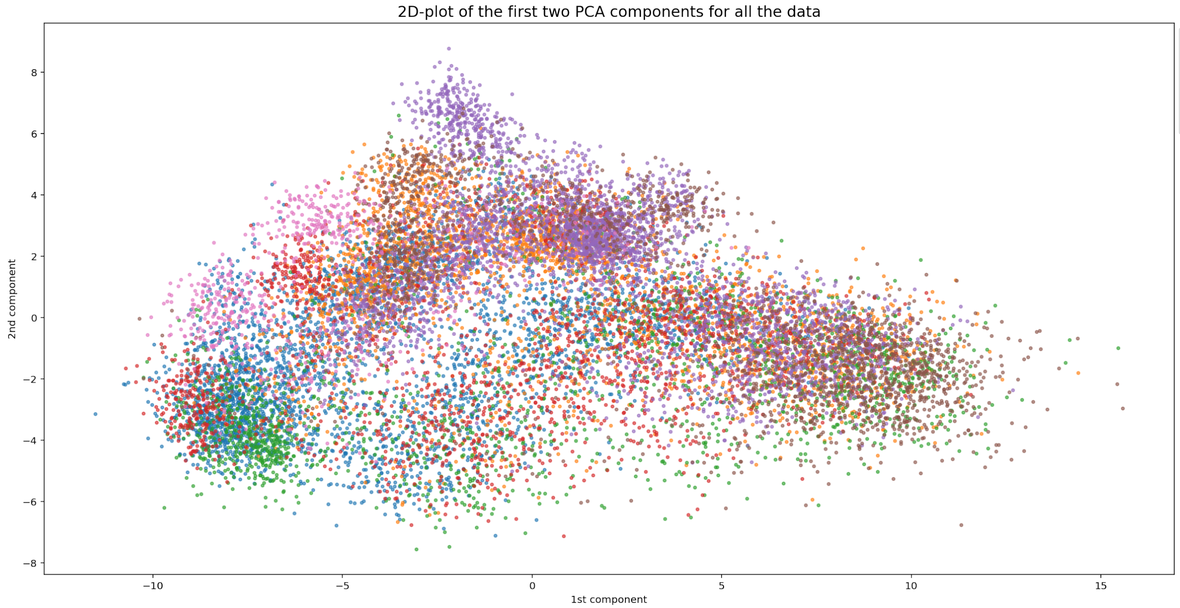
## Problem with Crema-D data

- ~500 fundamental frequency not retrieved from the files of this source
- Most features with remarkable gaps compared to the other sources
- Reviewing the documentation several differences emerged regarding the origin and purpose of the dataset



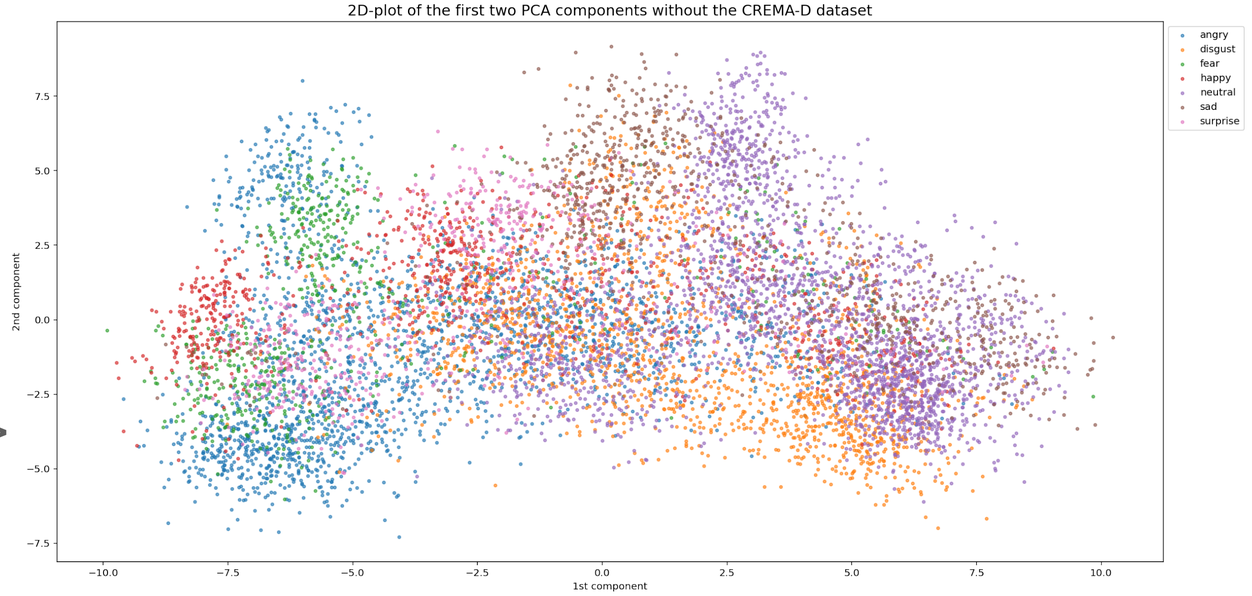
# Exploratory Data Analysis (EDA)

## Dimensionality reduction with PCA



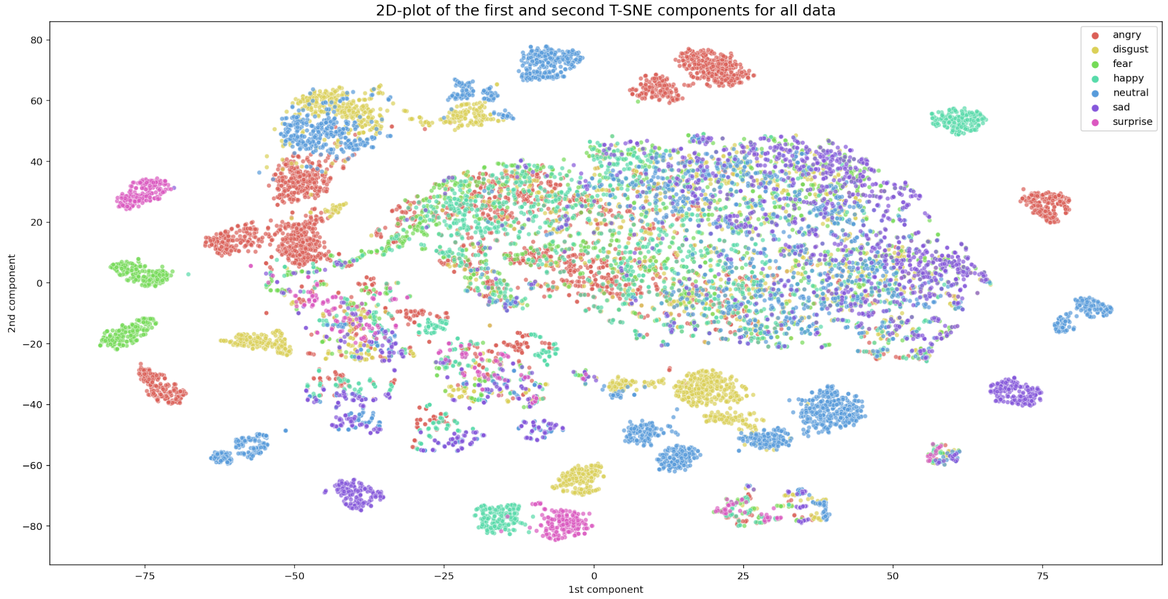

<— All the data

Without the Crema-D dataset —>



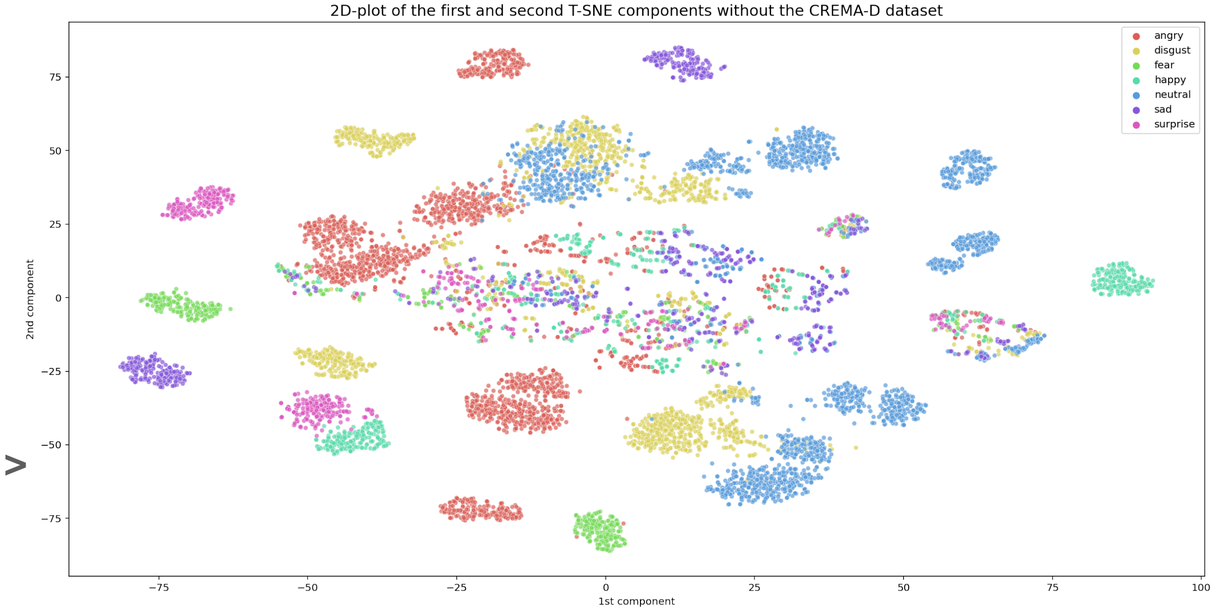
# Exploratory Data Analysis (EDA)

## Dimensionality reduction with T-SNE



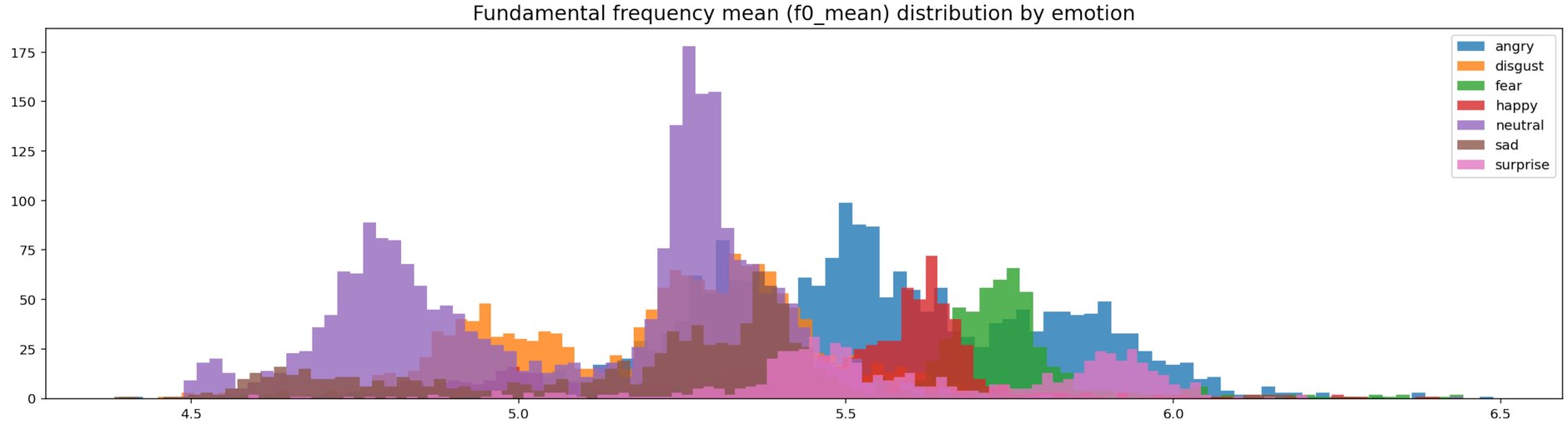

<— All the data

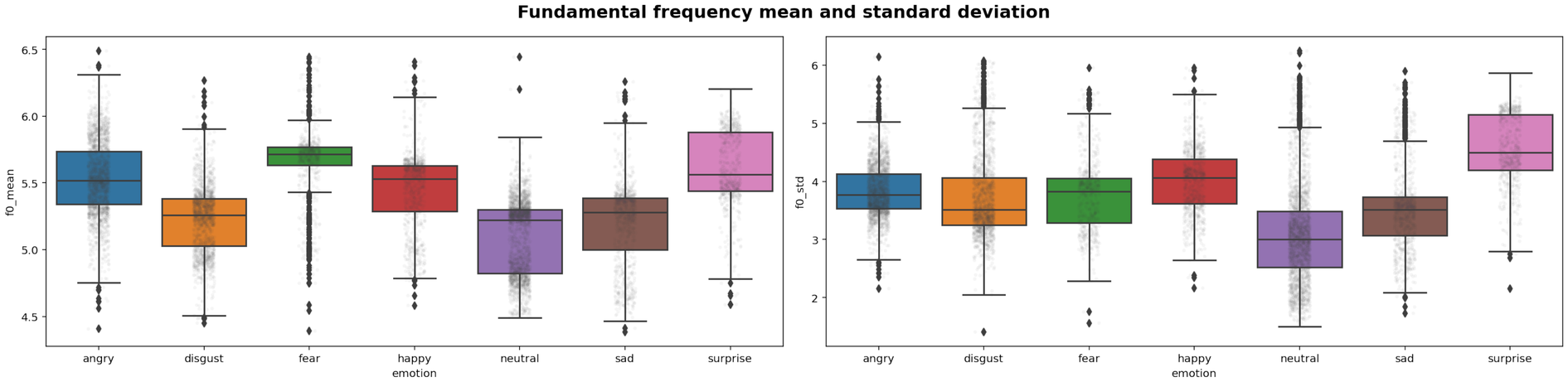
Without the Crema-D dataset —>



# Exploratory Data Analysis (EDA)

## Features with target

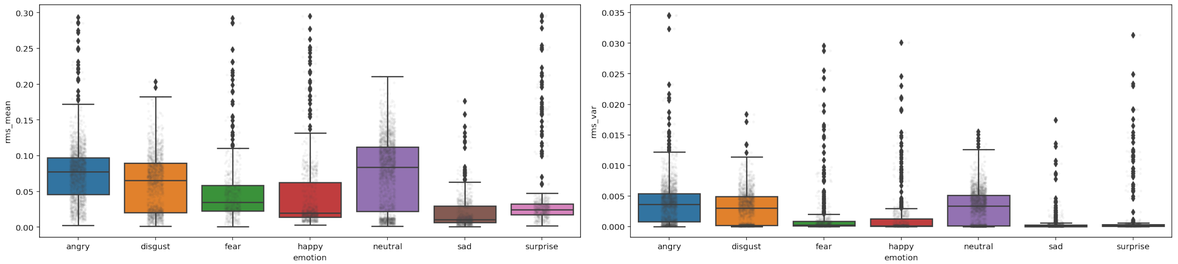


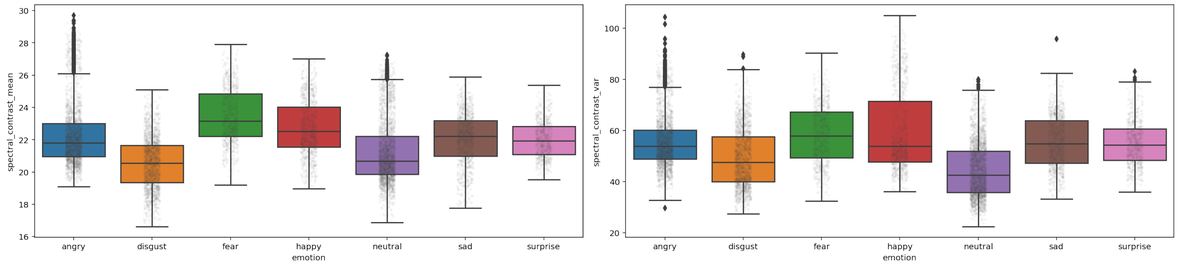
# Exploratory Data Analysis (EDA)

## Features with target

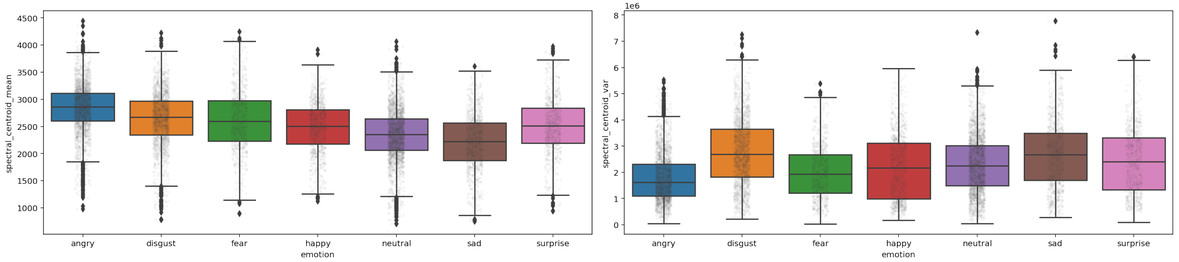
RMS mean and variance



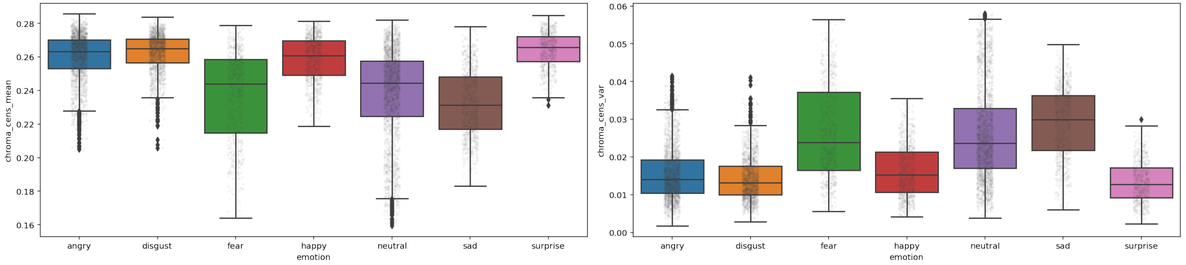

Spectral contrast mean and variance



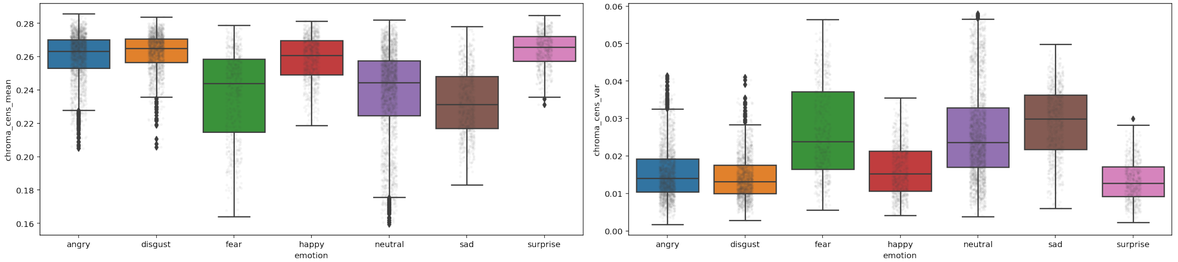

Spectral centroid mean and variance



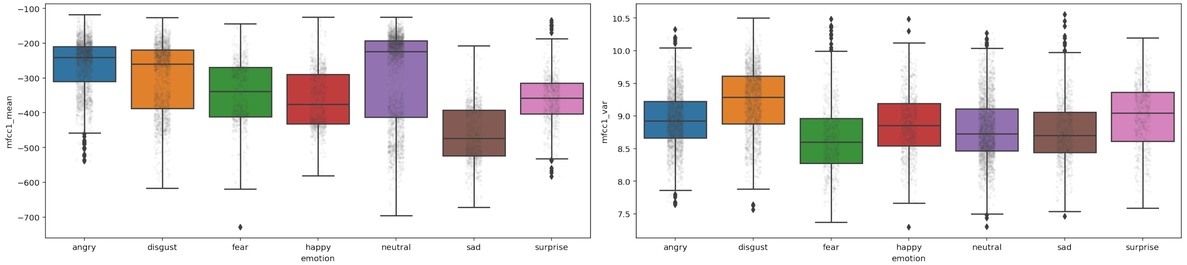

Chroma cens mean and variance




Chroma cens mean and variance

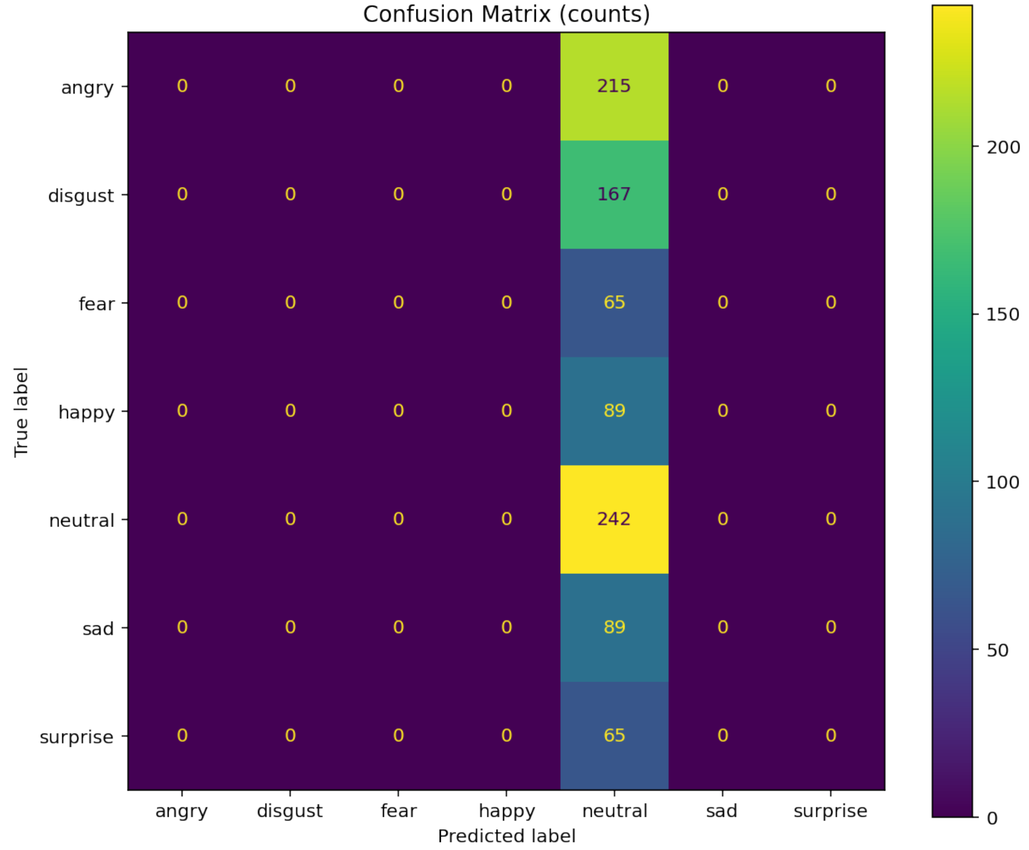


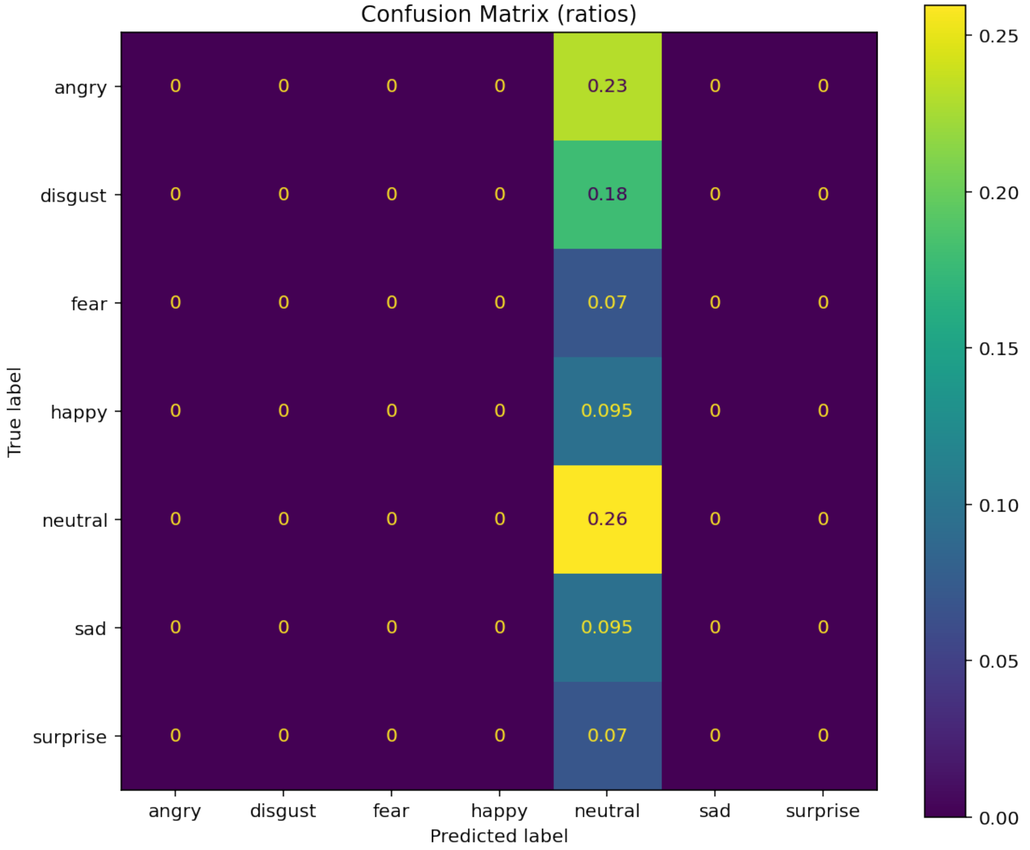

MFCC1 mean and variance



# Baseline

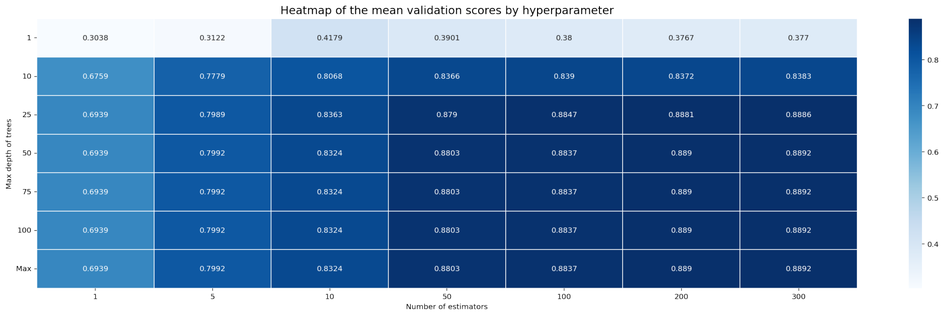
- *DummyClassifier* with *most_frequent* strategy
- accuracy 26% (neutral category)

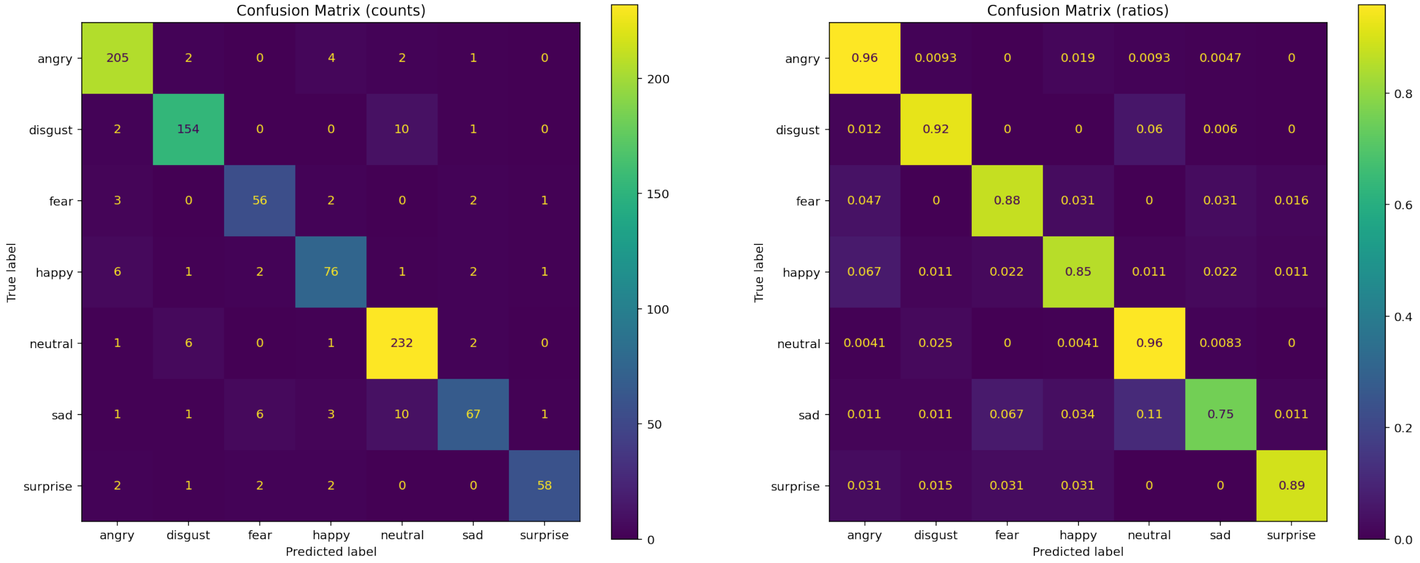


# Random Forest

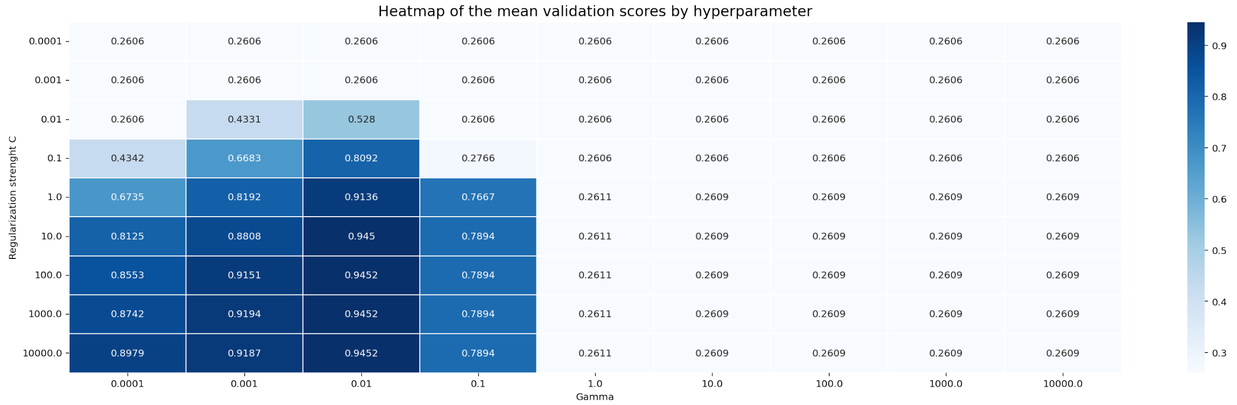
- Tuning hyperparameters with k-fold cross-validation grid search
- Max depth of trees: 50
- Number of estimators: 300
- Best features: root mean square (rms), mfcc1, fundamental frequencies data, chroma cens
- Worst features: zero-crossing rate, gender, spectral centroid, spectral flatness, some “mfccs” variances
- Accuracy on the test set: 91.6%

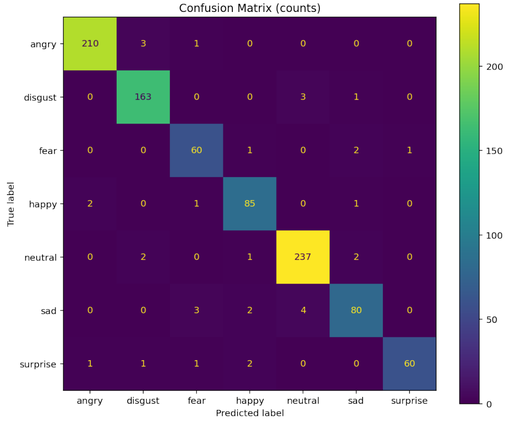


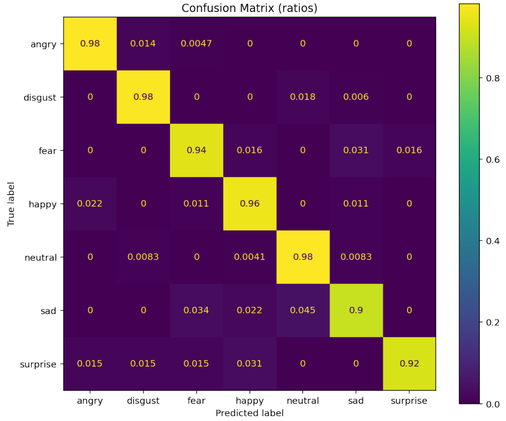



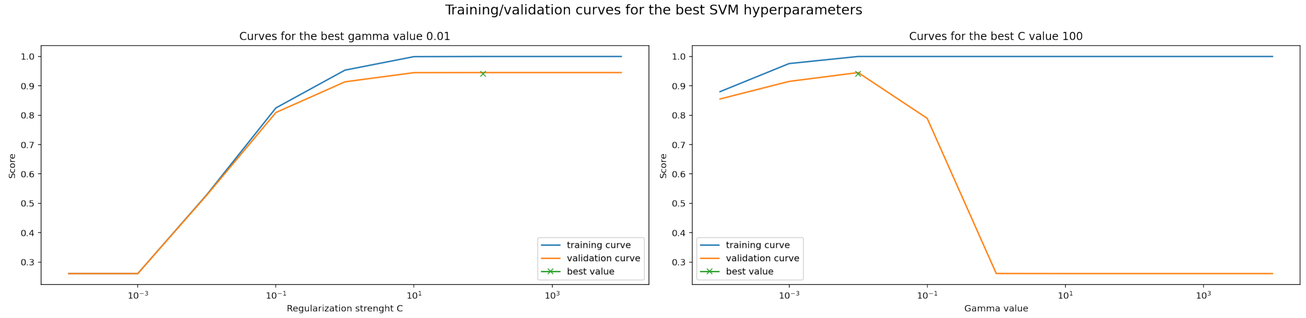
# Support Vector Machine

- Tuning hyperparameters with k-fold cross-validation grid search
- Regularization strength C: 100
- Gamma: 0.01
- Accuracy on the test set: 96.1%



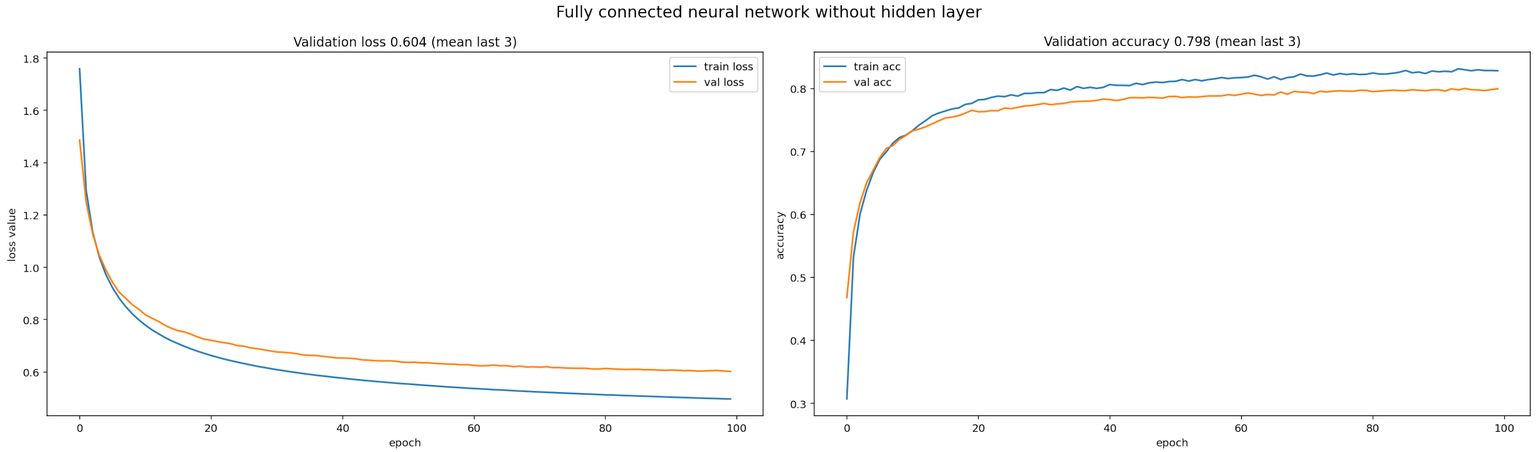
# Dense Network

## 1. Fully connected model without hidden layers

- (90 features * 7 categories) + 7 bias = 637 parameters
- Provided class weights to address samples imbalance
- EarlyStopping callback didn’t run even after 100 epochs
- Accuracy on the test set: 82.47%

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense (Dense) | (None, 7) | 637 |

```
Total params: 637
Trainable params: 637
Non-trainable params: 0
```



# Dense Network

## 2. Model with two hidden layers

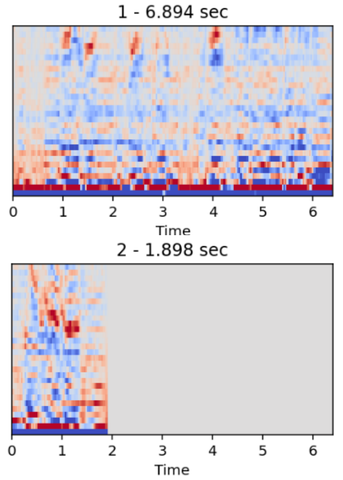
- Layer with 256 units and with 64 units for a total of 40199 parameters
- In all layers:
  - relu activation
  - variance scaling
  - L2 regularization
- Provided class weights to address samples imbalance
- Accuracy on the test set: 92.26%

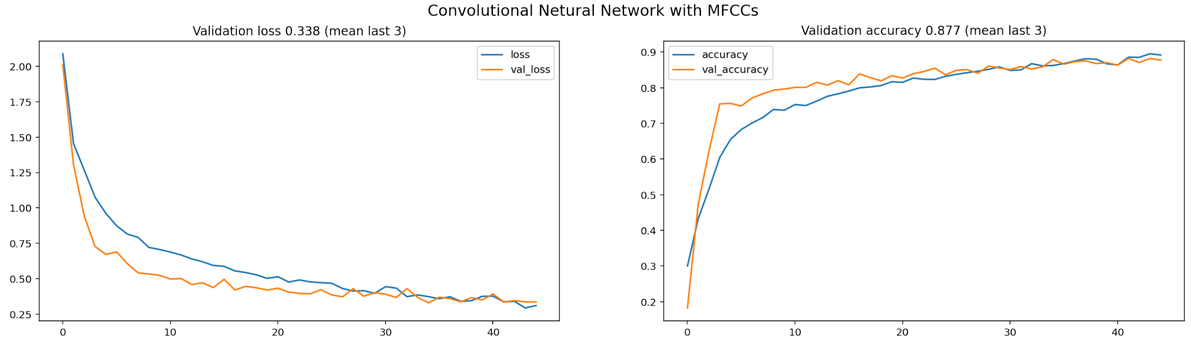
```
# Create model with hidden layers
model = Sequential()

# Hidden layer 1
model.add(Dense(units=256,
                activation=activations.relu,
                input_dim=90,
                kernel_initializer=initializers.VarianceScaling(scale=1.0, seed=0),
                kernel_regularizer=tf.keras.regularizers.l2(0.0001)))

# Hidden layer 2
model.add(Dense(units=64,
                activation=activations.relu,
                kernel_initializer=initializers.VarianceScaling(scale=1.0, seed=0),
                kernel_regularizer=tf.keras.regularizers.l2(0.0001)))

# Output layer
model.add(Dense(units=7,
                activation=activations.softmax,
                kernel_initializer=initializers.VarianceScaling(scale=1.0, seed=0),
                kernel_regularizer=tf.keras.regularizers.l2(0.0001)))

# Network summary
model.summary()
```

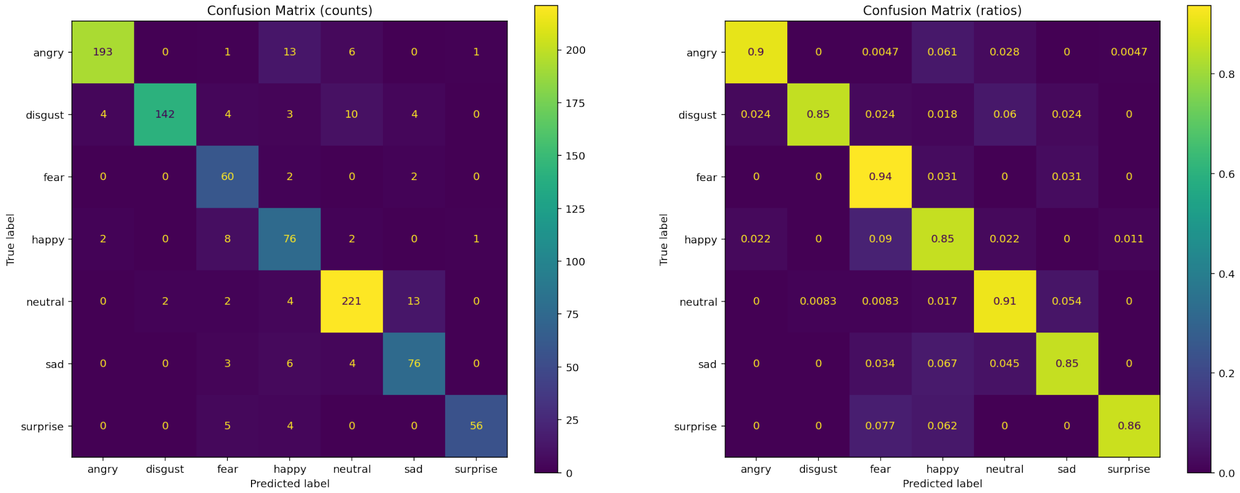
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 256)               23296

 dense_2 (Dense)             (None, 64)                16448

 dense_3 (Dense)             (None, 7)                 455

=================================================================
Total params: 40,199
```

# Convolutional Neural Network

- Extracted MFCCs data as 2D arrays
- Resized MFCCs “images” with the same size 30 x 200 (~ 6 seconds)
- Filters, max pooling, dropout: total of 918279 parameters
- Accuracy on test set: 88.49%

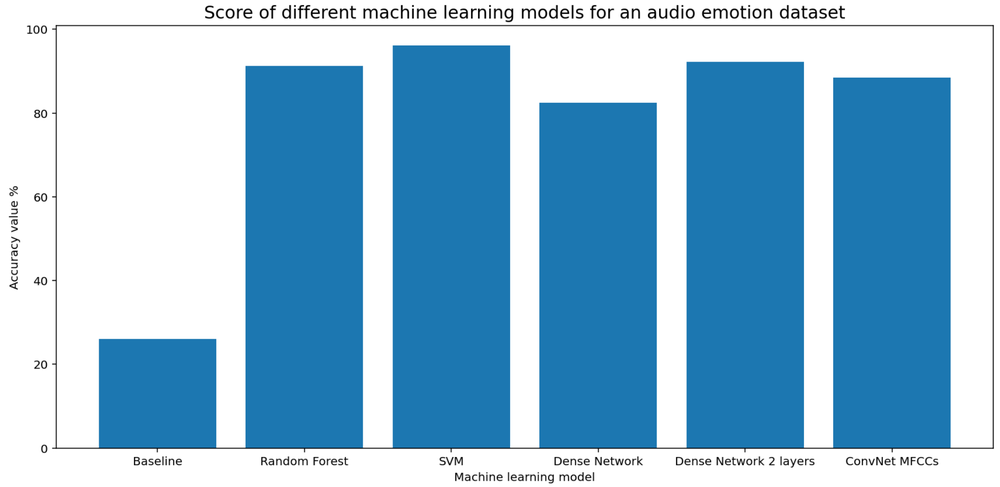


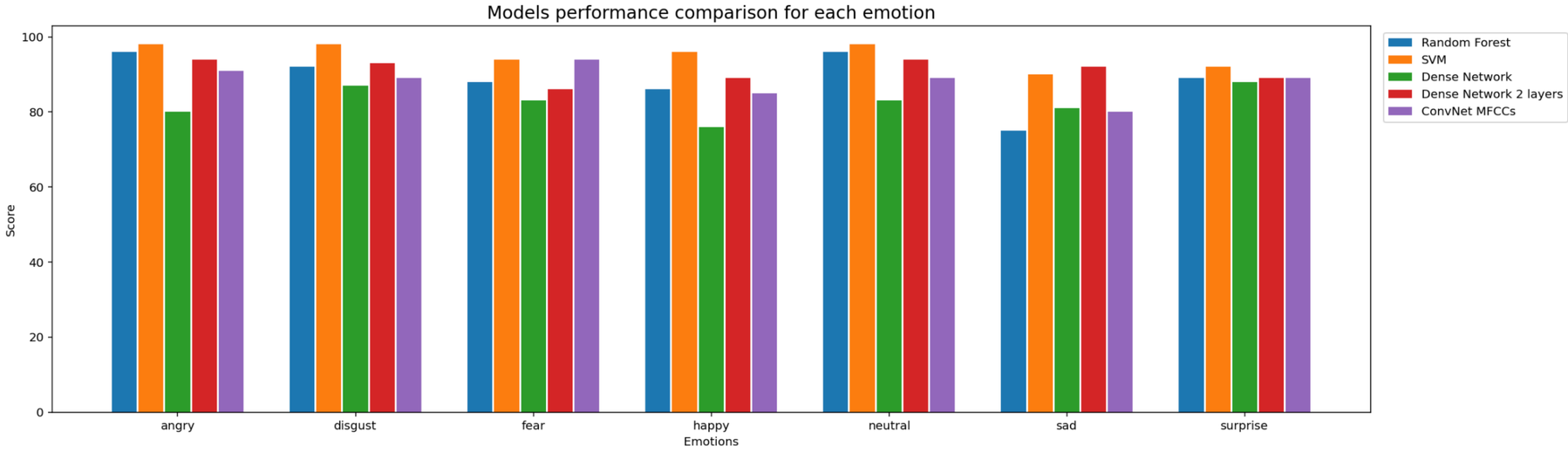



# Models performance comparison

| Model | Test Accuracy % |
|---|---|
| SVM | 96.10 |
| Dense Network 2 layers | 92.26 |
| Random Forest | 91.20 |
| ConvNet MFCCs | 88.49 |
| Dense Network | 82.47 |
| Baseline | 26.00 |




| | angry | disgust | fear | happy | neutral | sad | surprise |
|---|---|---|---|---|---|---|---|
| **Random Forest** | 96 | 92 | 88 | 86 | 96 | 75 | 89 |
| **SVM** | 98 | 98 | 94 | 96 | 98 | 90 | 92 |
| **Dense Network** | 80 | 87 | 83 | 76 | 83 | 81 | 88 |
| **Dense Network layers** | 94 | 93 | 86 | 89 | 94 | 92 | 89 |
| **ConvNet MFCCs** | 91 | 89 | 94 | 85 | 89 | 80 | 89 |



**“We are not thinking machines that feel; rather, we are feeling machines that think.”**
**Antonio Damasio**

# Thank you