Audio intent classification project

1st Emanuele De Leo
Politecnico di Torino
s318658
s318658@studenti.polito.it

2nd Giovanni Grossi
Politecnico di Torino
s318690
s318690@studenti.polito.it

***Abstract*—The Audio Intent Classification project proposes to develop a machine learning model able to classify a collection of audio files based on the intent of expressed by speakers. Each file is drawn from an audio data-set and plays one sentence belonging to a specific person speaking. The main objective of the research is to accurately identify the final intent expressed by the audio file, and therefore to understand the speaker's intention**

I. Problem overview

The project starts by collecting a series of audio recordings from different speakers. The files express the speakers' intentions concerning daily actions to undertake such as turning up the volume, activate lights, increasing the heating and so on. The goal of the project is to correctly identify the requested action uttered in each recording, Our data-set is splitted in:

- development.csv a data-set containing metadata about audio files. The data-set includes almost 10000 rows including the tagged actions that need to be analyzed
- evaluation.csv a smaller data-set with the same structure of the previous one without the intent classes .

The files include attributes regarding the speaker such as gender, range of age and language spoken. By deepening our analysis, we can notice that the database is not distributed in an equal manner. The total of possible actions displayed are seven, and they do not repeat with the same frequency during the day. Table 1 shows the distribution of target labels in development file. In addition, the number of audios recorded by each speaker appears to be different. As a result, it can be inferred that some actions and speakers are deemed to occur and be requested more frequently than others.

Another meaningful observation for the analysis concerns the features characterizing each speaker, indeed the speakers are numerous, and they differ in gender, age and accent since they all speak different native languages. Such features affect the tonality and frequency of the voices, and consequently have a strong impact on the final performance of the audio file.

By further analyzing the audio files, we noticed that the duration of the recordings was variable, displaying a peak of 20 seconds and a mean around 2.8 seconds.The significant variability of duration nudged us to modify the original data by cutting the long periods of silence precedent and subsequent the verbal pronunciation in order to reshape the values and to render them more suitable for our classifications' algorithms. To better depict the typologies of files we are working with, it could be useful to visualize them graphically. In Fig.1 we plotted the audio in a time domain, while Fig.2 represents a spectogram of an audio file, therefore his amplitude over time and frequency domain.

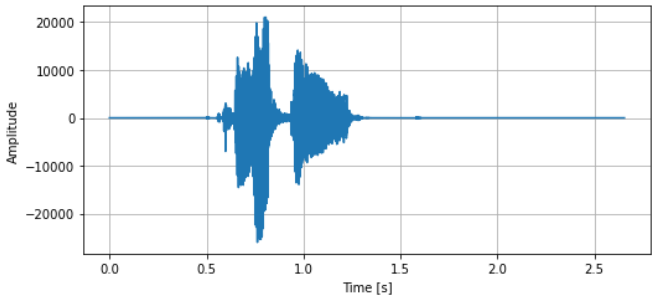

Fig. 1. amplitude over time representation

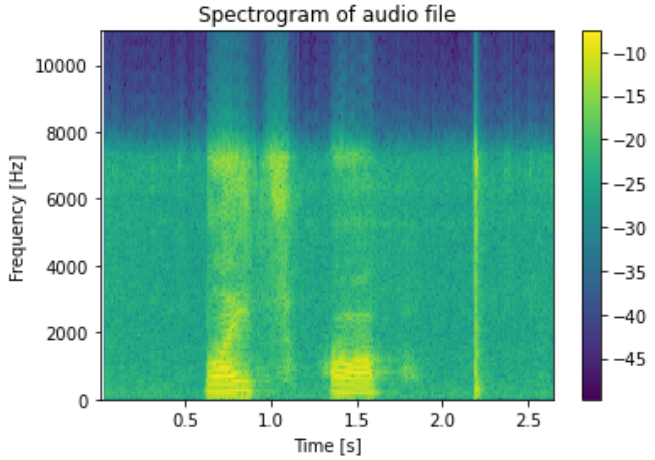

Fig. 2. amplitude over time and frequency

II. Proposed approach

A. Data preprocessing

The first action to undertake was to render the audio files in terms of equal length using the same sample rate, then we re-sampled each audio to 22.5kHz. Afterwords, we used two different approaches depending on whether the individual means exceeded or not the overall duration mean of the data-set. Therefore, for what concerns the audio files having a length higher than the overall mean, we removed the segments

TABLE I
FREQUENCY OF SPOKEN SENTENCES

requested action	n. of files
change language	1113
activate music	791
deactivate lights	592
increase volume	2614
decrease volume	2386
increase heat	1209
decrease heat	1189
total	9884

of audio where the amplitude resulted equal to zero and we added the remaining ones having a lower length. This method allowed us to collect the same number of features for each file.

As mentioned, the audio files differed a lot from each other, therefore for the extraction process we needed a efficient range of features able to separate the information deriving from the speech and to capture only the relevant information for the analysis. After some analytical research we identified the main features that are most used to classify audios. Among these we found The Mel-Frequency Cepstral Coefficients(MFCCs), [1] commonly used for classifying audio data in the field of speech recognition and musical genre detection. Indeed, this set of coefficients represents the spectral content of an audio signal in a compact and meaningful way and contains information on how the energy bands changes over time in different frequency. The MFCC coefficient is interpreted in a way such that if it assumes a positive value, it indicates that that the majority of the signal's energy is concentrated at a lower frequency region. Conversely, if a MFCC coefficient has a negative value, it points out that most of the energy is concentrated at higher frequencies levels.In fig.3 a graphical representation of 20 MFCCs extracted from an audio of the collection. The results are expressed as numerical matrices, but let's see how these matrices are extracted [2]:

1) *Pre-emphasis*: The audio signal is passed through a high-pass filter to emphasize higher frequency components.
2) Framing: The signal is divided into overlapping frames, typically with a length of 20-40 milliseconds.
3) *Windowing*: Each frame is multiplied by a windowing function, such as a Hamming window, to reduce spectral leakage.
4) *Fourier Transformation*: The framed and windowed signal is transformed into the frequency domain using a Fourier Transformation
5) *Mel-frequency warping*: The power spectrum is transformed into the Mel-frequency scale, which is a non-linear scale based on the perceived pitch of the human ear.
6) *Logarithm*: The Mel-frequency spectogram is converted to a logarithmic scale.
7) *Discrete Cosine Transform (DCT)*: The log Mel-frequency spectogram is transformed into the ceptral domain using a DCT.

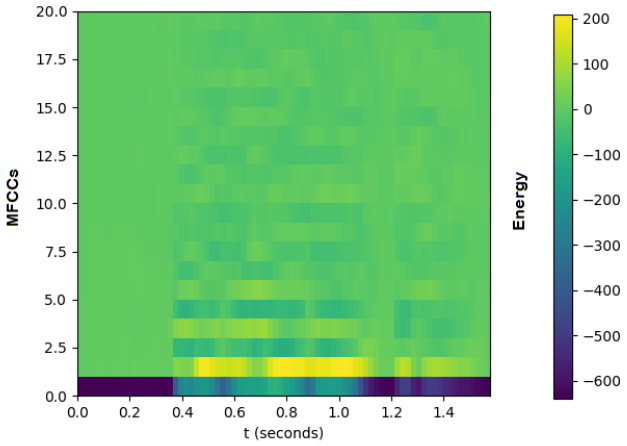

Fig. 3. figure of MFCC for one audio file

8) *Ceptral Coefficients*: The first few coefficients of the DCT are taken as the MFCCs, which represent the most important features of the audio signal.

Fig 4 visually represents the overall scheme for MFCCs' extraction.

Once this process had been completed we could finally study the results. Since the data values had a very wide range, we normalized those values with the min-max scaling technique before feeding them to the classification models. The MFCCs from each audio file correspond to a large number of values, so we had to compress them with an appropriate transformation of the data to avoid the overfitting on our classifier; because of this precaution, we deemed appropriate to use the PCA (Principal Component Analysis technique). The PCA reduces the dimension of the features' space by identifying the most important ones (the principal components) corresponding to a projection that captures the maximum variance in the data, while discarding the less important ones. Specifically, the PCA model transforms a n-dimensional matrix to a m-dimensional matrix ($m \leq n$) with respect to the variability of the original columns. This method can also be used to visualize the features' space in a lower-dimensional one, which can make our model easier to be understood and facilitate the interpretation of the data. Moreover, it can be useful for understanding the characteristics of the audio signal and their mutual correlations. In our case, the MFCCs correspond to different audio frames, and the use of PCA allows us to derive the best frames that describe all audios allowing the classification task.

Consequently we will have to tune n number of MFCCs and m size of PCA to extract the most useful features from/by raw data in order to fit our classifier.

B. Model selection

We tested two different classifier algorithms

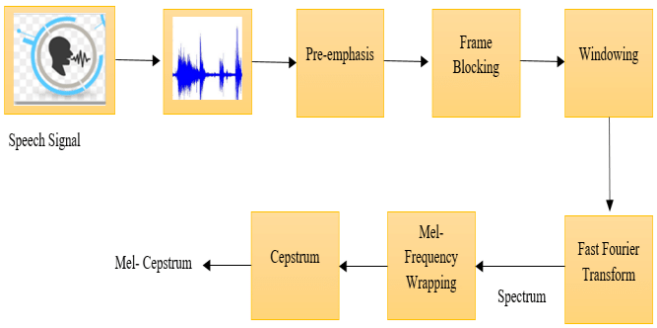

Fig. 4. extrction scheme of MFCCs

- The first one can be identified is a machine learning model that uses multiple decision trees to make predictions, that is Random Forest. The decision trees are trained on different portions of data and also on different subsets of features, this helps to avoid the overfitting problems that can occur with a single decision tree by the vote of many of it to assign the class labels. this algorithm provided good results in problems simalr to ours [3]
- The second one concerns the SVC (Support Vector Classifier) This algorithm transforms the data points using a possible non-linear transformation and identifies the maximum-margin hyperplane that separates the classes. SVM has also been successful in audio signal classification [4], that's why it is another option to consider despite the amount of time it takes managing a large amount of data.

C. Hyperparameters tuning

The parameters we considered to tune are the number n of MFCC extracted from each file and the number m of PCA to reduce the dimension of our data-set. We also had to tune the hyperparameters for the chosen classification algorithms. The papers of research we have consulted, performing this kind of task, commonly used around 13 MFCCs extracted [5], also we observed that for a larger number of coefficients the test results failed, so we will evaluate values in a range close to 13. To tune the number of PCA we will start from 1 and then increasing it until we reach stability in the accuracy score considering the different hyperparameters for models.
We splitted our train set in 80/20 to train and test the models and we ran a grid search to tune m and n. Following that, we used the same approach to tune the hyperparameters of our classification models by embedding the best values of m and n previously found. In Table 2 we collected the hyperparameters to be tuned. With regard to Random Forest we tried to improve the number of estimators to have more accurate results, so we set it to be 1500. Regarding SVC we understood that for different combination of hyperparameters it returned the same results, so we tested it with C=5, kernel=rbf and gamma=0.1.

TABLE II
HYPERPARAMETERS CONSIDERED

Model	Parameter	Vlues
Mfccs	n	1→20, step 1
Pca	m	7→60, step 2
Random forest	criterion max depth max features min impurity decrease	{gini , **entropy**} **None**→20, step 2 {**auto** , log2} 0.05, **0.1**, 0.2
SVC	kernel C gamma	{linear, sigmoid,**rbf**} **10**, 15, 20 0.2 ,**0.1**, 0.4

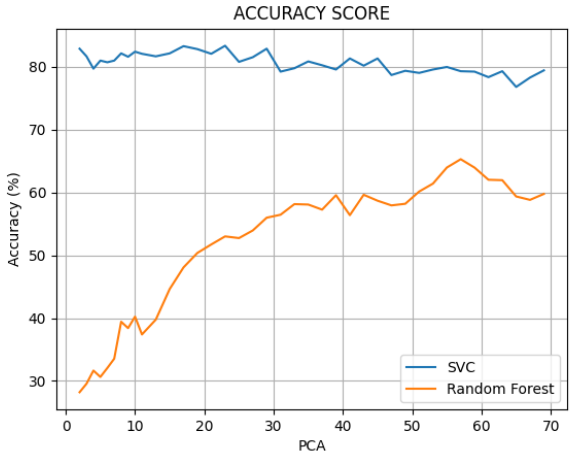

Fig. 5. accuracy RFC

III. RESULT

The validation of the models has been done splitting the train data in a stratified fashion based on the class labels:

- 80 percent as train-set to fit the model
- 20 percent as test-set to make the prediction

In the case of SVC model has been done also the cross validation technique, because the model generated problems of overfitting. SVC learned too strictly from the train-set and when we fitted it with the evaluation-set the accuracy was very lower than the one obtained simply splitting the data-set. The cross validation has been done with 5 partitions. The accuracy of models has been computed considering the predictions on the test-set with its original labels. By doing so we did not resolved the proble

The tuning of the parameter n, corresponding to the number of components of the new dimension generated by the PCA transformation, is summarized in the line chart Fig. 5. It can be clearly seen that the best choice of n for Random Forest is around 55-60, and it is the best performing model on evaluation set despite its results during the validation. SVM seemed more accurate than Random Forest, we have achieved an accuracy around 84% during the validation phase. Unfortunately these values of SVC are not correct because of the overfitting mentioned before.

A random guess on this problem will provide a 15% accuracy, comparing our results with the others in the leader board is evident that we could have tried different solutions to approach the problem.

For comparison we tested a naive solution composed of the following steps :

1) Extraction of the spectogram of each file so dB width over time and frequency.
2) Splitting the spectogram matrix in chunks and compute mean and standard deviation for each of them.
3) Train Random forest and SVC with default values on the extracted features.

This approach resulted on 40% accuracy.

IV. DISCUSSION

Our approach provided a better solution then the naive baseline. We could say that MFCCs represent powerful features in a file audio, and selecting from them the most relevant ones looks promising to achieve even better results. We reached a satisfying accuracy score on a test-set of audio with different distortions and silences, and this result pointed out that we had done an adequate preprocessing step. There are some procedures that could improve our result:

- *Data augmentation* is a technique that slightly modifies the data to expand the train set for better feeding the classification algorithm. However we did not apply it due to time and computational problems, we believe that better results could be achieved applying this technique.
- The extraction of *others audio features* might improve the outcomes, indeed, trying different combinations could be crucial for a more accurate recognition. We tried features such as spectral-centroids, roll-of and spectogram, but with poor results.
- Another utility for the project would have been to tune *several parameters* on the MFCCs extraction such as the frame length [6].
- Another approach could be based on *neural networks* [7] and the automated extraction of the features which has proven to achieve excellent results in this field. cit6 We have tried to develop a sequential model for our purpose, but the complexity of a good structure in terms of coding and timing was too high.

The data preprocessing has been the main problem in terms of computational time, especially in this case since we used Librosa library, that is perfect for the purpose of process audio files, but its functions are quite slow.

The obtained results and the breadth of possibilities stimulate us to believe that we are close to achieve a fine resolution of the problem.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “Feature Normalisation for Robust Speech Recognition,” https://arxiv.org/pdf/1507.04019.pdf, 14 JUL 2015.

[2] TY - JOUR AU - Gupta, Shikha AU - Jaafar, Jafreezal AU - Wan Ahmad, Wan Fatimah AU - Bansal, Arpit PY - 2013/08/31 SP - 101 EP - 108 T1 - Feature Extraction Using Mfcc AUG 2013

[3] Khadar Nawas, Manish Kumar Barik, A Nayeemulla Khan “Speaker Recognition using Random Forest”https://www.itm-conferences.org/articles/itmconf/pdf/2021/02/itmconf_icitsd2021_01022.pdf,

[4] Jia-Ching Wang, Jhing-Fa Wang, Cai-Bei Lin, Kun-Ting Jian, and Wai-He Kuok , “Content-Based Audio Classification Using Support Vector Machines and Independent Component Analysis ,”https://www.researchgate.net/publication/220928088_Content-Based_Audio_Classification_Using_Support_Vector_Machines_and_Independent_Component_Analysi 24 AUG 2006.

[5] Amir Hossein Poorjam, “Why we take only 12-13 MFCC coefficients in feature extraction?,”https://www.researchgate.net/post/Why_we_take_only_12-13_MFCC_coefficients_in_feature_extraction/5b0fd2b7cbdfd4b7b60e9431/citation/download. 20018.

[6] Masoud Maleki, Diagnosis of COVID-19 and Non-COVID-19 Patients by Classifying Only a Single Cough Sound,https://doi.org/10.48550/arXiv.2102.04880 8 Feb 2021.

[7] Juncheng B Li, Zheng Wang, Shuhui Qu, Florian Metze, “Robustness of Neural Architectures for Audio Event Detection,”https://arxiv.org/abs/2205.03268 6 May 2022.