Report exam winter 2023

Emanuel Buttaci
Data Science and Engineering
Politecnico di Torino
Turin, Italy
s308589@studenti.polito.it

Giacomo Rosso
Data Science and Engineering
Politecnico di Torino
Turin, Italy
s309273@studenti.polito.it

***Abstract*—In this report we introduce an approach to classification of audio recordings, specifically commands for voice assistants. Our approach relies on the extraction of statistical features, time domain features and frequency domain features from audio recordings, subject to proper preprocessing. The choice of machine learning models mainly fall onto ensemble classifiers and neural network, which indeed turn out to be the outperformers among all classifiers, when properly tuned. In the end, our strategy is capable of delivering a score ranging from 79% to 94% on the provided evaluation set, which highlights the successfulness of our approach.**

I. Problem overview

Audio classification has been a much investigated topic in the last decades. In this project we are dealing with a dataset of audio recordings of several people with different characteristics. These recordings represent command for a voice assistant like Alexa. The objective is to recognize which commands are expressed in each audio recording, that is a classification problem. The dataset is divided into two parts.

- **development set**, which contains about 9854 **labeled** records and will be used to train our model.
- **evaluation set**, which contains about 1455 **unlabeled** entries and will be subject to our classification pipeline.

II. Proposed approach

A. Preprocessing

Before diving into our problem approach, we need to explore how data is structured.

Missing values No missing value is present and we do not require any imputation strategy.

Evaluation problem considerations When exploring the distributions of attributes emerges there are different current language spoken, where English appears to be the most widely spoken, followed by French. Native speakers are the vast majority. Furthermore, we can count up to 87 different speakers, but the recordings are not evenly distributed among them. Concerning gender, there is an almost uniform distribution as expected, while age range distribution is more sparse as speakers are older, which means most of the speakers are young people. However, exploration of evaluation set is very relevant, since it gives us an idea of how our problem is constrained with respect to the dataset attributes. Surprisingly, as we can see from data exploration, the evaluation set is composed only of native English speakers, which is a strict constraint on our problem. Anyway, this could be an advantage to us, since native speakers will not suffer any foreign accent in speaking English. Thereby, we take our first decision, which is to train whatever model only on English speakers, thus removing all foreign language speakers from development set. Potentially, in case there would be multiple languages in the evaluation set, we may think of a strategy which trains multiple classifiers, each one corresponding to a different language. In our case, we discarded such approach since English is the unique language appearing in the evaluation set.

Prior features removal By removing foreign speakers, we lose any discriminant information about the language spoken, which is uniquely English. Therefore we also decide that any attribute about language will be discarded since it is irrelevant in our approach. Using the same consideration, we also ignore the speaker level information, which is unique in the evaluation set and therefore constant. Anyhow, we also decide to keep all kinds of speakers from training set since they may capture different accents within the English language. Evidently, none of the speakers from evaluation set is also part of the development set, which leads us to indeed discard the ‘speakerId‘ information, which cannot relate any information between training set and evaluation set. Only ‘gender‘ and ‘ageRange‘ remain, which could be easily encoded by means of one hot encoding. Yet, we think that spectral information extracted from each voice recording may encode the difference between voices belonging to both genders and different age ranges. Thus, we also eliminate ‘gender‘ and ‘ageRange‘ attributes. Indeed, by training any model using such attributes we had proved how scores get worse when they are included.

Audio recordings exploration This approach leaves us only the audio recording, which, in fact, will be the effective object of feature extraction. We exhaustively motivated why we believe that audio recording may encode all meaningful information regarding our problem. These are encoded using wav format, namely sequences of 16 bits integers holding amplitudes in time domain. Some minority of recordings belonging to the development set share a higher sample rate of 22050 Hz, which has to be adjusted in order to have uniform recordings. The strategy which is going to be employed is a down resampling of such recordings to the mostly spread sample rate of 16000 Hz. Also, the lengths of audios are quite different, ranging from roughly 10k samples to 441k samples, though the distribution of lengths is stronly centered towards shorter lengths. This could be a clue suggesting the presence of long leading and trailing silences in some longer recordings. However, the 95% of audios' lengths fall within about 57k samples.

Target Inspection In the classification problem the target label (speakers' commands) is made by labels ‘action‘ and ‘object‘. Since we think that some combinations of ‘action‘ and ‘object‘ may be semantically meaningless, for example ‘change language‘ + ‘volume‘, we decide to combine from the very beginning both targets into a single target which directly encodes the intent. The encoding follows the same rule that is expected in the evaluation. Therefore, we are going to construct the overall target by means of a string concatenation, namely the intent. We end up with 7 different classes, which constitutes our final target for the classification problem.

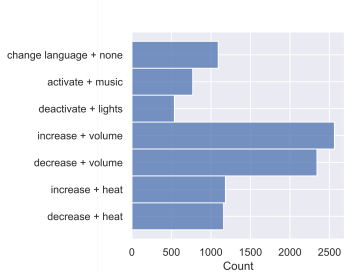

Fig. 1. Distribution of intent across development set

As it is seen in Figure 1 target categories are not uniformly distributed. Besides, each attempt to balance the problem, such as sampling the same amount of audio recordings with respect to each target category, returns worse results.

Wave signal processing As mentioned, the provided audio files are encoded in wav format, that is a sequence of 16 bits integers. Specifically, all the recordings have been sampled at 16000 Hz. Before proceeding, each audio recording is normalized in range [-1, 1] using 32768, which is the maximum feasible (absolute) value for any amplitude (largest 16 bits integer). The first step consists in cleaning each recording from irrelevant silence by cutting off those amplitudes which are below a certain threshold, set to 20 dB. The choice has been made in order to preserve the overall resemblance of the trimmed audios with respect to the original ones. Then, we have to uniform the amount of samples across all recordings, that is the length. To us, the simplest way to achieve this consisted in computing the index of the last non zero amplitude across all recordings, in order to exclude the trailing silence. Hence we compute the 95% percentile of these lengths. This value will be the final number of samples chosen for each recording. Longer recordings in time are cut (because of our choice they are 5% of all), while shorter recordings are filled with trailing zeros. Finally this results in roughly 30000 samples, corresponding to a duration of 1.9 seconds since the sampling frequency is 16000 Hz.

Feature extraction Since feature extraction is the most important step in the construction of our pipeline, we are going to extract three macro groups of features from audio recordings, each one capable of encoding relevant information when modelling each recording's wave signal.

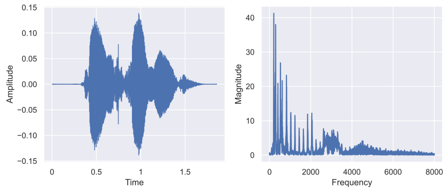

Fig. 2. Amplitude and Fourier transformed of original audio[1]

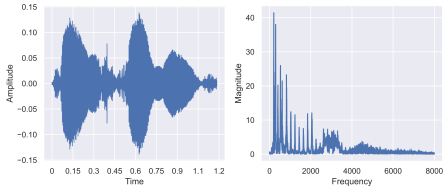

Fig. 3. Amplitude and Fourier transformed of processed audio

Statistical features Statistical features are information computed in time domain. Particularly, we divide each recording $x(t)$ in time windows of the same duration, that is 50 milliseconds, whose value was chosen after some attempts. So, each window will contain exactly $W = 800$ samples and each recording, due to a length of about 30000 samples, will be partitioned into less that 40 windows. Now, for each n-th window, we compute some statistical information.

TABLE I
STATISTICAL FEATURES

Feature	**Value**	**Granularity**
Mean	Average amplitude	window
Standard deviation	Deviation from the mean	window
Skewness	Asymmetry of wave signal	overall
Kurtosis	Tailedness of wave signal	overall

Time domain features Time domain features are another set of meaningful features related to the shape of each recording in the time domain, again computed with respect to each time window.

TABLE II
TIME DOMAIN FEATURES

Feature	**Value**	**Granularity**
Zero crossing rate	Axis traversal count rate	window
Root mean square	Signal energy	window
Peak-peak distance	Absolute total amplitude	window
Peak	Maximum absolute amplitude	window
Crest factor	Peak-energy ratio	window
Shape factor	Energy-absolute mean ratio	window
Impulse factor	Peak-absolute mean ratio	window
Margin factor	Energy-square root mean ratio	window

[1] Audio file is *0a3129c0-4474-11e9-a9a5-5dbec3b8816a.wav*

Frequency domain features Concerning about the spectral domain, each wave signal can be transformed from the time domain into the frequency domain. The transformation is performed through the discrete Fourier transform (DFT), which delivers a spectrum of magnitudes associated to each frequency (only real ones). Like for other features, we consider several time windows.

TABLE III
FREQUENCY DOMAIN FEATURES

Feature	Value	Granularity
MFCC	Mel cepstral coefficients	window
Mel spectrogram	Mel scale spectrogram	window
Spectral centroid	Weighted mean of frequencies	window
Spectral rolloff	85% percentile frequency	window
Spectral bandwidth	Variance w.r.t spectral centroid	window
Spectral flatness	Noisiness of wave signal	window
Spectral kurtosis	Frequency spectrum tailedness	window
Spectral skewness	Frequency spectrum asymmetry	window
Spectral flux	Power spectrum change rate	window
Spectral contrast	Energy peak-minimum difference	window
Spectral entropy	Power spectrum Shannon entropy	window

Mel-frequency cepstral coefficients are very popular in audio analysis and classification. Such coefficients are extracted from a complex computation involving Fourier transform and discrete cosine transform on logarithmic power spectrum in Mel frequency. Mel scale is a logarithmic scale, very close to the human perception of frequencies. Thus, the usage of Mel-frequency cepstral coefficients is a powerful tool in audio recognition. All other features are extracted from the spectrum of magnitude of frequencies or from the power spectrum, encoding information about the shape of frequency distributions within time windows.

B. Model selection

Implementation of FFNN Leaded by the thought of not missing any opportunity in making more accurate predictions, we decided to implement a feed forward neural network by stacking different layers. The choice related to the number of layers, neurons and epochs has been made very practically by making different attempts in assessing the performance over the development set. Anyway, such investigation has been avoided here.

Model performance evaluation Classification is carried out on a representative sample of 1000 elements, in order to avoid using the whole dataset, which would result in huge computational times. The training set is 80% of the original sample (800 records). After audio preprocessing, a standardization step is prefixed to those models which otherwise would suffer the absence of a common scale within the data, namely the logistic regression, support vector machine, nearest neighbours and neural network. Though all scores are similarly aligned, ensemble models performed better than others.

TABLE IV
BASE SCORES ON MULTIPLE CLASSIFIERS

Model	Sample score
LogisticRegression	0.465
RandomForestClassifier (RF)	0.460
HistGradientBoostingClassifier (HGB)	0.520
SVC	0.455
KNeighborsClassifier	0.425
FeedForwardNeuralNetworkClassifier (FFNN)	0.500[2]

Feature selection The amount of features extracted from the audio recordings is relatively large, that is 2000 features for roughly 9500 records. Computational costs are to be taken into account. Therefore different strategies have been tested in order to decrease the dimensionality of our data. In the most simple case a variance threshold was used, but it did not improve the results significantly, since the selection criterion was based on a statistical property, namely the variance, rather than the true meaningfulness of features with respect to the target. Another employed strategy consisted in using f-scores from builtin ANOVA statistical tests. Relative performance was marginaly improved both on development set and evaluation set. However, the technique which delivered the most promising performance was based on features importances from a fitted ensemble model. In practice, feature selection using a trained tree based classifier, according to an importance criterion, did outperform other techniques partly on development set and mainly on the evaluation set. Indeed, in the end we employed an extra trees classifier with 750 trees and entropy criterion for the computation of features importances. This tree based ensemble is peculiar since the decisions on which attributes to split are made randomly instead of optimistically, as it happens for any random forest.

C. Hyperparameters tuning

We decide to investigate and improve the performance of three promising models, namely ‘RandomForestClassifier‘, ‘HistGradientBoostingClassifier‘ and ‘FeedForwardNeuralNetworkClassifier‘. Again, for computational reasons, we decided to select a subset of 1000 elements from the development set. Note that optimization is performed by selecting the best scoring model, averaged over a 5-fold partition.

Random forest The random forest model is a powerful ensemble model which trains independently several decision trees, each one on a random sample of the dataset. Moreover, each tree selects only a subset of features to evaluate each split. Therefore, resulting trees are decorrelated. This technique has the main advantage of being robust to overfitting. In our tuning we focus on the quantity of trees and the split criterion chosen at each node.

[2] The implementation using Keras library suffers from the lack of setting a random state for creating reproducible results. So, as it can be seen in practice, scores obtained using the neural network will often be similar tough not the same.

TABLE V
SEARCH GRID OF HYPER PARAMETERS

Model	Parameter	Values
RF	random_state	**0**
	n_estimators	100, 250, 500, **750**
	criterion	gini, **entropy**, log_loss
HGB	random_state	**0**
	learning_rate	0.075, **0.1**, 0.25
	loss	**log_loss**
	max_iter	100, 125, **150**
FFNN	activation	**relu**, tanh, sigmoid
	optimizer	**adam**, sgd, adadelta
	epochs	100, 150, **200**
	batch_size	**128**

Gradient Boosting In order to capture some relationships within the data in more depth, we decide to optimize the gradient boosting model. Like the random forest, it makes use of several trees. However the training process is completely different. Whilst the random forest independently fits each decision tree, possibly in parallel, instead the gradient boosting model trains each decision tree sequentially, thus it is not capable of parallelizing. Anyhow, its training benefits from a corrective process in which each new tree is trained on the residual error. Therefore this tree is constructed in order to correct the prediction error of previous trees with respect to the target. The residual error and correction step are applied by means of gradient computation and descent. The learning rate and number of epochs is investigated during optimization in this case.

Feed forward neural network The last model which we decide to optimize is the feed forward neural network, according to the architecture chosen for the implementation. A neural network has a very high capability of learning the non linear relationships from the data and the target. Such training is performed through a variant of stochastic gradient descent. In practice the network is trained in order to minimize the prediction errors and the neurons' coefficients are updated accordingly by means of gradient computation and descent. In this case we try different activation functions involved in computing each neuron's output.

III. RESULTS

Once we found out the optimal configurations, let us compute the average score on the entire development set by means of a 5-fold cross validation strategy, which gives an idea of the overall performance of the optimized model. Besides, since feature selection gives better results, we opted for the extra trees classifier's features importances. Table VI shows the improvements achieved by implementing such technique. Evidently, the neural network outperforms the other tuned models, while the gradient boosting seems better than the random forest. Since the neural network uses gradient descent in order to minimize the error function when fitting the data, this could lead to a deeper interpretation of the relationships between features, particularly the non linear ones. The same applies to the gradient boosting classifier, apparently like a random forest, which is trained on the minimization of residuals. In fact, the random forest constructs many independent trees and such independence may be the cause of missed relationships in the data, which is solved by the gradient based mechanism behind the gradient boosting model.

TABLE VI
SCORES OF TUNED MODELS ON DEVELOPMENT SET[3]

Model	Score	Score after feature selection
RF	0.654 ± 0.054	0.661 ± 0.057
HGB	0.696 ± 0.049	0.688 ± 0.052
FNN	0.760 ± 0.057	0.766 ± 0.053

The highest submission scores on the leaderboard are possibly obtained by tweaking some hyper parameters after the optimization step, that is the only way we figured out to improve the scores. In particular the highest score for ‘**FeedForwardNeuralNetworkClassifier**‘ is **0.930** which is due to the randomness present in the training of the model. Therefore we made use of a **voting classifier** constructed with an odd number of neural network, 17 voters to be precise, for simulating a majority voting mechanism. In this way we could achieve the average performance of the neural network without much randomness. Indeed we got the highest score of **0.942**. Regarding ‘HistGradientBoostingClassifier‘, the peak obtained on the evaluation set was 0.854. Finally, the ‘RandomForestClassifier‘ touches its best score of 0.797 without further tweaking after optimization.

IV. DISCUSSION

We strongly believe that our approach and experimentation to this classification problem has been quite successful. Our belief mainly lies in the preprocessing and feature extraction steps, the core of our pipeline. Time and spectral features extraction turned out to be an exhaustive and meaningful way of performing feature engineering on audio recordings. Furthermore, the partitioning of each wave signal into time windows also yielded to a granular characterization of our audio recording. Overall, we ended up with about 2000 extracted features, which is a large amount, thus the implementation of features selection helped in distinguishing the most significant ones, roughly 950. Finally, using ensemble models and the much popular neural network, we were able to reach considerable accuracy scores varying from 61% to 76% on the development set and from 79% to 94% on the evaluation set, which again confirms that our approach was right in combination with such powerful classifiers. Possibly, other strategies may take place for further investigation, and we mainly were interested in the concept of wavelet transform, its coefficients and a more profound strategy of building the neural network inner structure. These could be some enhancements to our current approach presented in this project.

[3] Extra trees classifier with 750 trees and entropy split was trained onto development set and its features importances were employed for selection.

References

[1] Sim, J., Min, J., Kim, D., Cho, S. H., Kim, S., Choi, J.-H. (2022). “A python based tutorial on prognostics and health man- agement using vibration signal: signal processing, feature extraction and feature selection. Journal of Mechanical Science and Technology”, pp. 4083-4097, April 2022. http://doi.org/10.1007/s12206-022-0728-z

[2] P. Mahana, G. Singh (2015). “Comparative Analysis of Machine Learning Algorithms for Audio Signals Classification”, International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015.

[3] R. Lenain, J. Weston, A. Shivkumar, E. Fristed (2020). “Surfboard: Audio Feature Extraction for Modern Machine Learning”, May 2020. https://arxiv.org/pdf/2005.08848.pdf

[4] D. Gerhard (2003). “Audio Signal Classification: History and Current Techniques”, Technical Report TR-CS 2003-07, November 2003. https://www.uregina.ca/science/cs/assets/docs/techreports/2003-07.pdf

[5] Wikipedia contributors (2022). “Mel-frequency cepstrum”. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=1106625289

[6] Wikipedia contributors (2022). “Mel scale”. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Mel_scale&oldid=1117532023

[7] Wikipedia contributors (2022). “Spectrogram”. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Spectrogram&oldid=1127110588

[8] Wikipedia contributors (2022). “Discrete Fourier transform”. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Discrete_Fourier_transform&oldid=1130001170

[9] Pedregosa et al. (2011). “Scikit-learn: Machine Learning in Python”, JMLR 12, pp. 2825-2830.

[10] Chollet, F., others (2015). “Keras”. GitHub. https://github.com/fchollet/keras

[11] Alexey Natekin1 and Alois Knoll (2013). “Gradient boosting machines, a tutorial”, DepartmentofInformatics,TechnicalUniversityMunich, December 2013. https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021/full