Speaker Age Estimation

Simone Francesco Licitra
Romeo Vercellone
Politecnico di Torino
Student id: s344643
Student id: s341967
s344643@studenti.polito.it
s341967@studenti.polito.it

***Abstract*—This study proposes a solution for estimating individuals' age by analyzing their speech audio recordings. We evaluate various regression models, initially, trained using only the basic data provided, subsequently incorporating additional features extracted from the audio samples. Furthermore, we develop a custom ensemble model that integrates three distinct sub-models to account for gender-specific characteristics while maintaining a generalized approach to the task. The performance, advantages and limitations of the proposed methods and models are thoroughly analyzed and compared.**

I. Problem Overview

This task consists of the estimation of people's ages by training a regressor model on features extracted from the audio of the speakers. The dataset is composed by 3624 samples, each sample is described by some features already extracted by the authors of the dataset, for example $gender$, $jitter$, and $shimmer$, and a column containing the name of the audio related to said sample. Specifically, we are given a dataset divided in:

- two $.csv$ files:
  - development.csv (development set of 2933 samples containing also the target age).
  - evaluation.csv (evaluation set of 691 samples).
- two folders with audio recordings of the speakers.

Let us have a look on how data is distributed in both datasets. In particular, we analyze age, gender and ethnicity, meanwhile for the evaluation set we can see only the last two.

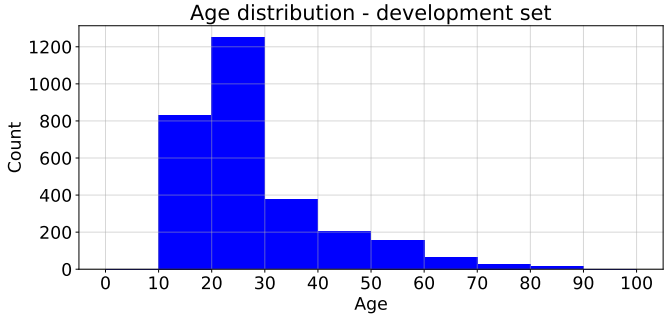

Fig. 1. Age distribution of the development set

By looking at these figures, we can make several statements. In Figure 1 we can clearly see how imbalanced the age distribution is. In fact the median is located at the age of 23. This could be a problem if our evaluation set contains a different distribution, it could cause our model to underestimate the target, increasing the error.

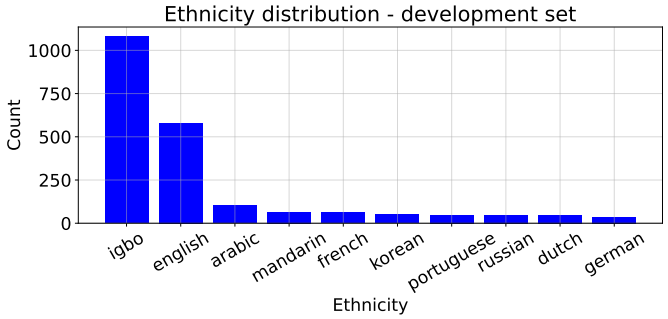

Fig. 2. Ethnicity distribution - development set

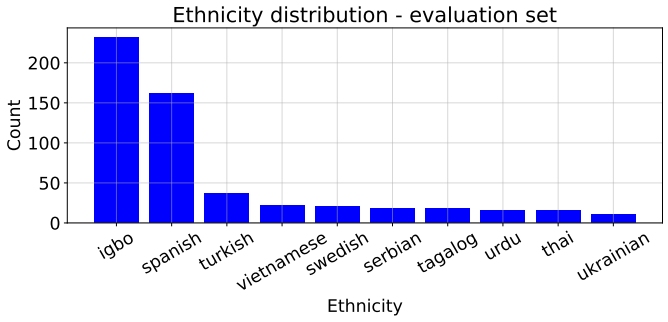

Fig. 3. Ethnicity distribution - evaluation set

Figure 2 and Figure 3 show only the ten most frequent ethnicities. We can clearly see that "igbo" people are the most frequent for both dataset, but on the other hand, all other most frequent ethnicities do not match at all. So we can predict that this feature will not be useful during our model validation phase.

Finally, we have also looked at the "gender" feature, and fortunately people are almost split evenly. We only show the distribution of "gender" of evaluation set because we detected an outlier (misspelled word) Figure 4, in fact one person is a 'famale' and we will handle her later in preprocessing step.

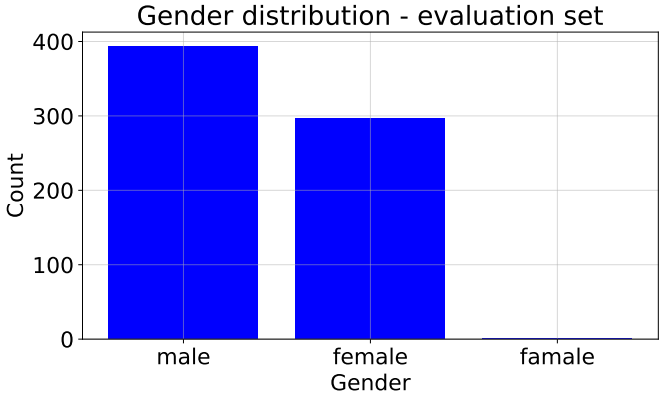

Fig. 4. Gender distribution - evaluation set

II. Proposed approach

A. Preprocessing

Let us have a closer look to the dataset and its features. We can eliminate the column 'sampling rate' since it presents the same value of 22050 Hz for all samples. Secondly, we have to handle categorical data and transform it as a decimal number. Our rules for encoding are:

- 'gender':
  - $male \rightarrow$ +1
  - $female \rightarrow$ -1
  - $famale \rightarrow$ -1
- 'tempo': from string to float by removing the square brackets.
- 'ethnicity': eliminated from the dataset.

The second step is a log_{10} transformation to help distribute a feature's value more evenly in it's new support, the motive surged when a distribution had a positive skew, like in Figure 7, and the resulting distribution in Figure 8. But it can help even when most of the value are concentrated in the lower part of the support, depicted in Figure 5. The effect of the transformations is in Figure 6.

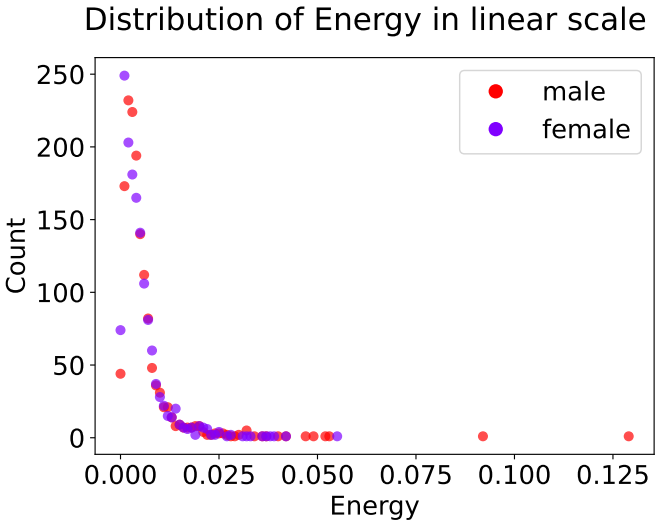

Fig. 5. Distribution of energy in linear scale

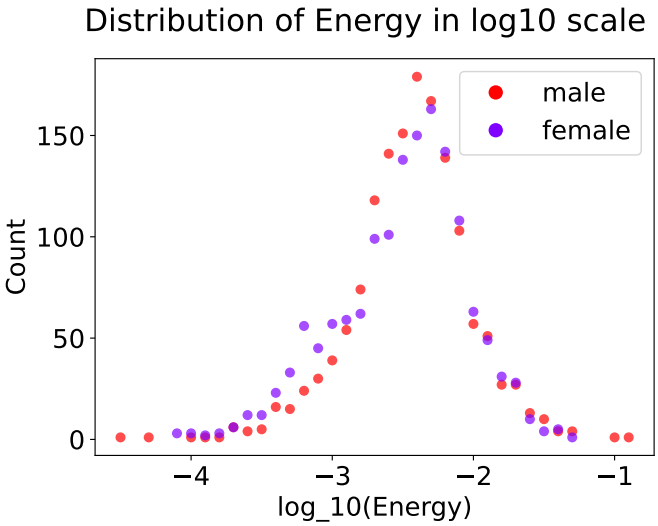

Fig. 6. Distribution of energy in logarithmic scale

The result is that we applied a log_{10} transformation to the columns: 'mean_pitch', 'jitter', 'shimmer', 'energy', 'zcr_mean', and 'spectral_centroid_mean'.

We extracted several extra features related to acoustic properties of the voice from the audio files in order to increase the quantity of information and to find all possible ways to improve our model.
They can be divided in macro-argument based on which fundamental measurement is used:

1) Time domain: 'audio_length', 'mean_absolute_slope', 'pitch_iqr', 'voiced_frames', 'number_of_frames', 'sm', 'sv', 'ss', 'sk', 'Time 5%', 'Time 25%', 'Time 50%', 'Time 75%', 'Time 95%', 'duration_50', 'duration_90' That relates to how long is the audio, the average of the absolute changes in amplitude, how much time is spoken during the audio, some statistical analysis of signal mean ('sm'), variance ('sv'), skew ('ss') and kurtosis ('sk'), and finally analysis of energy distribution, 'Time 25%' represent the time required to reach 25% of total energy.
2) Frequency domain: 'dominant_frequency' is the frequency with the highest amplitude when the Fourier series is applied to the audio.
3) Spectrogram: 'spect_overall_mean', and 32 'spect_frequency_mean' each one related to the mean of a part of the temporal frequency spectrum
4) MFCC (Mel-frequency cepstral coefficients): 35 'mfcc_frequency_mean' which represent the mean of a mfcc at every part of the spectrum
5) PolyFeature: 'mean_coeffs', 'std_coeffs' which capture the mean and standard deviation of the term of first grade

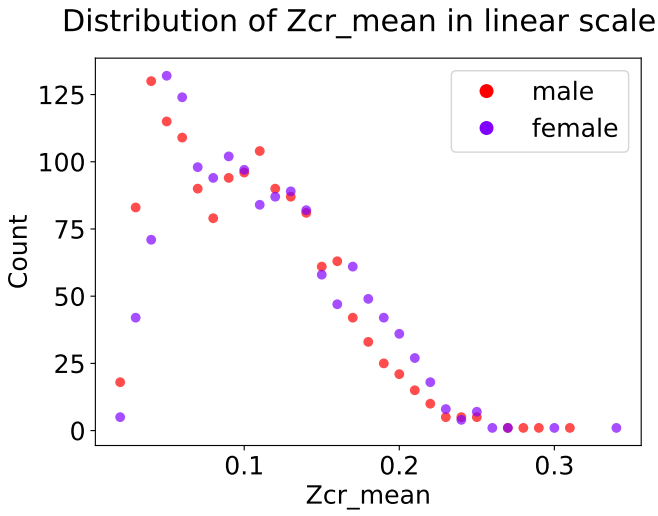

Fig. 7. Distribution of zcr_mean in linear scale

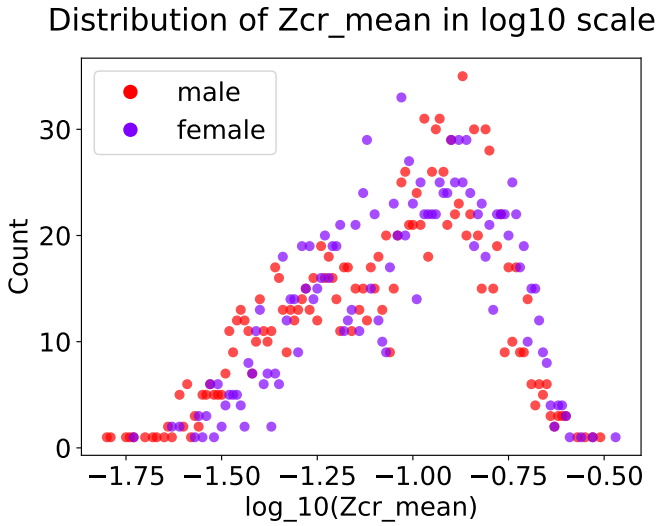

Fig. 8. Distribution of zcr_mean in logarithmic scale

of a polynomial describing the shape of the spectrum

6) Formants: 'f0_mean', 'f1_mean', 'f2_mean', 'f3_mean', 'f4_mean', 'f0_var', 'f1_var', 'f2_var', 'f3_var', 'f4_var'. The mean and variance of the first 5 formants of the voice captured
7) Temporal Median: 'temporal_median' that is the middle value in a sorted set of amplitudes, providing a measure of the central tendency of the audio.
8) Entropy: 'temporal_entropy', 'frequence_entropy', 'mean_spectral_entropy' which respectively evaluates whether the energy is evenly distributed throughout the duration of the audio, in the spectrum, and entropy of the mean spectrum over time.
9) 'words_per_second', 'char_per_second' which describe the number of words and characters spoken per second

There are some features that we calculated but in the end have not used, so we will avoid explaining them. Finally, we decided to aggregate number of words spoken and characters detected to the audio length, this will bring an improvement on the score.

B. Model selection

Initially, we trained several regression models available in *scikit-learn* without tuning hyperparameters, so we just tried basic versions to have an idea of which of them could work better for this task. In order to try different models, we need to standardize the data.
The evaluation metric considered to be minimized in this problem is the root mean squared error defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

where:

- n number of samples,
- y_i real target,
- $\hat{y}_i$ predicted target.

Model	RMSE
Linear reg	11.006
Decision Tree	15.783
Random Forest	10.954
HistGB	10.858
SVR	11.394
KNN	12.315
MLP	10.736
Lasso	11.158
Ridge	11.001
TFR	12.163

TABLE I
DIFFERENT MODELS' PERFORMANCE COMPARISON

Model	RMSE
Linear reg	10.574
Decision Tree	14.837
Random Forest	10.378
HistGB	9.916
SVR	11.140
KNN	10.744
MLP	10.093
Lasso	10.865
Ridge	10.640
TFR	11.943

TABLE II
BASIC MODELS' PERFORMANCE WITH AUDIO FEATURES

As we can see in Table I, Random Forest, HistGradient-Boost, and MLP are the best with default hyperparameters, in fact we expected this result because:

- **Random Forest**: is a tree based model which builds several decision trees that are trained with different random sub sets of the dataset. So, every tree is different from the others, reducing overfit. this type of model could perform well also if the number of features increase greatly. And lastly, Random Forest works well also with not standardized data.

C. Hyperparameters tuning

In our case, Hyperparmeters fine tuning was mandatory since the resulting scores were unacceptable. We fine tuned two models, Random Forest and our own regressor, from now on called TripleForestRegressor or 'TFR'.
For the RandomForestRegressor we applied a Grid search algorithm where we tested:

- n_estimators: 500
- max_depth : [None, 10, 20, 30]
- min_samples_leaf: [1, 2, 4]
- random_state: 341967

Our model is an ensemble of three RandomForestRegressor, one trained on the whole dataset, while the other two are gender specific forests.
For the TFR we applied a incremental grid search algorithm, since the number of combination of hyperparameters was extremely high. So we first applied grid search on the forest regarding females, then on the one trained on males, and finally the one on the complete dataset. At each step the best result of the previous step is used, sadly so we do not guarantee the best configuration possible but just a local one.
The search space used for each forest in TFR is the one previously described for a simple RandomForestRegressor.

III. Results

We have finally chosen best hyperparameters for every model, and also for the sub-models inside the TFR one. In Table III are present the values of the hyperparameters obtained for the TFR regressor. In Table IV we show the validation score associated to the fine-tuned models and their associated score on the public dataset, Table V. Finally in Table VI we show the score on the public set when we just increase the number of estimators of all the RandomForestRegressor's to 500.

Model	Parameters	Best values
Full forest	*max_depth*	10
	min_samples_split	10
	min_samples_leaf	2
Male Forest	*max_depth*	None
	min_samples_split	2
	min_samples_leaf	4
Female Forest	*max_depth*	10
	min_samples_split	5
	min_samples_leaf	1

TABLE III
BEST HYPERPARAMETERS VALUES FOUND

Model	RMSE
Random Forest	10.226
TFR	10.258

TABLE IV
VALIDATION SCORE OF FINE-TUNED MODELS WITH AUDIO FEATURES

IV. Discussion

Finally we can amply discuss our model's performance. We have shown that RandomForestRegressor and an ensemble of multiple forests, can significantly improve the overall score both in the extracted validation set and on the public platform. We saw how even if at the start our model was not comparable with the others more simple models, after fine-tuning it decreased it's validation score by almost 2 points. It was not able to keep up with MLP or HistGradientBoostRegressor, but we preferred to avoid them and use more simpler models.
More optimizations, both on features extraction and their selection, could help the regressor substantially reduce the score. Furthermore, we believe we could have reached a better score in general if:

- features had the same distribution in both datasets,
- the development set contained enough information for all age groups, instead of being focused on younger people.

Finally we could have gone deeper on searching the best hyperparameters, although we would have to run a grid search with an unfeasible possible amount of configurations for the three different sub-models (for now the best configuration for each forest differ from one another), so it could have taken too much time.

Model	RMSE
Random Forest	9.725
TFR	9.652

TABLE V
FINAL PUBLIC SCORE WITH FINE-TUNING

Model	RMSE
Random Forest	9.834
TFR	9.796

TABLE VI
FINAL PUBLIC SCORE WITH BASIC MODELS

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
[2] M. Notter, "Age prediction of a speaker's voice." https://medium.com/epfl-extension-school/age-prediction-of-a-speakers-voice-ae9173ceb322, 2022.
[3] NowYSM, "Feature extraction from audio." https://www.kaggle.com/code/ashishpatel26/feature-extraction-from-audio, 2019.