Age Estimation by Voice: Speech Analysis Approaches

Alice Banaudi
Carla Finocchiaro
Politecnico di Torino

Student id: s347490
s347490@studenti.polito.it
Student id: s337024
s337024@studenti.polito.it

***Abstract*—In this report, we presented our methodologies for speech analysis aimed at age estimation. To achieve this, we extracted mel spectrograms from the audio recordings and computed statistical features in the time domain to enrich the dataset for model training. Various regression models were evaluated based on their Root Mean Squared Error (RMSE) performance. Among the models tested, Random Forest and Extra Trees demonstrated the best results. After hyperparameter tuning, the Extra Trees model outperformed Random Forest during the testing phase, delivering superior accuracy in age estimation.**

I. Problem Overview

Our project focuses on a regression problem using a provided set of audio recordings and their corresponding dataset, which includes pre-extracted acoustic characteristics. The objective is to estimate the speaker's age by leveraging various features, some of which were already provided, while others were extracted during our analysis. The dataset is divided into two parts:

- a **development** set , containing 2,933 samples, each having 20 features, including the file path to the corresponding audio and the age label
- an **evaluation** set, containing 691 samples, having the same features as the development set, except for the age label.

First, we analyzed the distribution of our target variable. As shown in Figure 1, the majority of speakers are in their twenties. This suggests that our model will likely perform better at predicting the age of younger individuals.

All audio files in the dataset share the same sampling rate, although their durations vary. This was addressed during preprocessing by computing statistical features across the time domain, ensuring that each speaker has the same number of features, regardless of the length of the audio. The features extracted from the audios are categorized into both the time and frequency domain. Figure 2 displays the audio recording in the time domain, where we observe that the audio lasts for 35 seconds. Several moments show the amplitude approaching zero, which corresponds to pauses made by the speaker during their speech. According to the dataset, the speaker takes 39

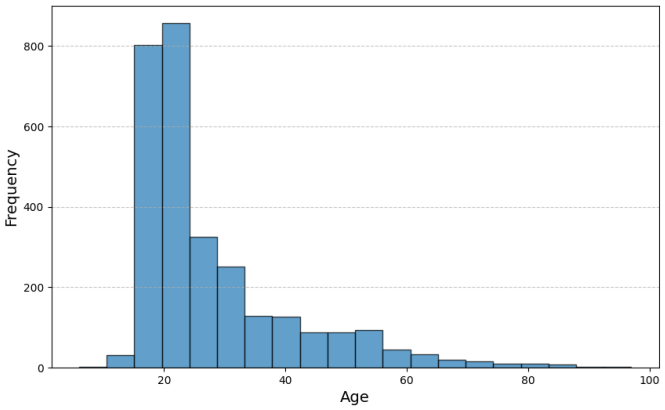

Fig. 1. Distribution of our target variable:age.

pauses throughout the recording. This kind of observations could be valuable for our analysis, as the frequency of pauses may provide insights into the speaker's age—we observed that older individuals tend to take more pauses while speaking.

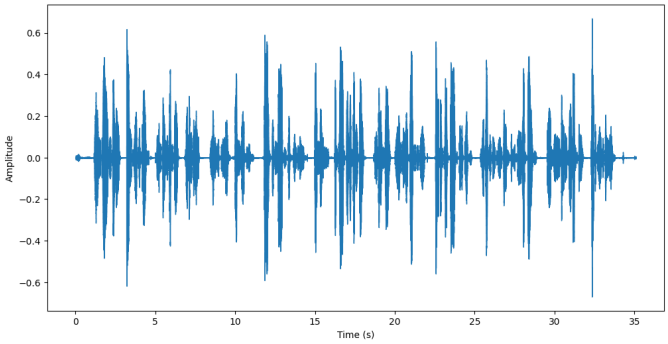

Fig. 2. Audio represented in the time domain .

To capture the fundamental characteristics of sound, we extracted the mel spectrogram for each audio file. This type of spectrogram is specifically designed to align more closely with how humans perceive sound, providing a more relevant

representation. Figure 3 is the mel-spectrogram of the same audio analyzed before: the areas that tend to darker shades are the ones with the lowest energy, with the ones in black indicating silence. We will use this representation of the audios to extract many features in our analysis.

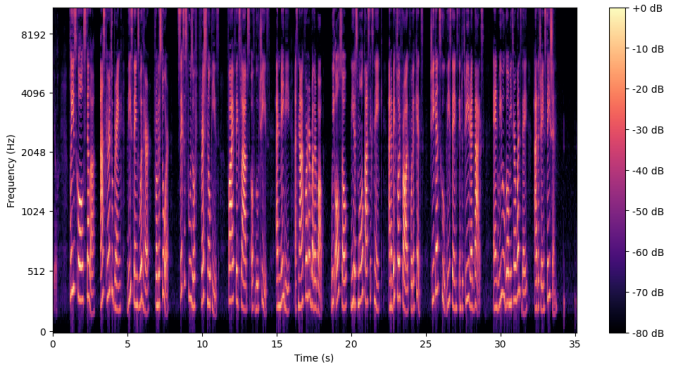

Fig. 3. Mel spectrogram of an audio.

II. Proposed approach

A. Preprocessing

The first step involved verifying that there were no missing values in the dataset and exploring the categorical features. During this process, we identified an issue in the gender feature, where one of the labels was misspelled. This error was corrected to ensure consistency in the data. Subsequently, we applied label encoding to this feature to convert it into a numerical format suitable for modeling. We also had a categorical feature representing the ethnicity of the speaker. However, we decided not to convert this data or include it in our models, as we did not find it relevant for distinguishing someone's age.

We already had several features extracted and included in our dataset, such as measures of voice stability and variability (jitter, shimmer), along with parameters related to the signal's energy. Other features in the dataset were the zero-crossing rate (zcr), the mean spectral centroid, and the speaking rate (tempo). Voice quality was represented by the harmonic-to-noise ratio (hnr). Additionally, the dataset contained information about the number of words, characters, pauses, and the total duration of silence in the signal. We extracted the following features:

- **MFCC**. Many studies found that the Mel frequency Cepstrum Coefficient (MFCC) is an optimal feature to use in speech recognition. MFCC is derived from how the human ear perceives sound. The human auditory system does not interpret frequencies in a linear manner. Instead, for each sound with a given frequency (measured in Hz), there is a subjective pitch that is represented on the Mel Scale [1] . In particular, we extracted the first 13 coefficients and we computed the mean for each of them, reducing the dimensionality of the features.
- **Spectral rolloff**. The spectral rolloff indicates the frequency point at which a specified percentage of the total spectral energy is concentrated in the lower frequencies. This feature can be useful for distinguishing differences in vocal characteristics : as a person ages, alterations in the vocal tract shape and size of their vocal tract that can be captured by the spectral rolloff feature [2]. The mean and standard deviation of the spectral rolloff across the entire signal were computed.
- **Spectral bandwidth**. The spectral bandwidth measures the range of frequencies surrounding the spectral centroid that encompasses a specified percentage of the total spectral energy [2]. Similarly to the rolloff it can be useful to train our model. Same as before we computed the mean and standard deviation.
- **Rate of speech**. The rate of speech is calculated by dividing the number of words by the duration of the audio.

Moreover, we created a new feature using the maximum and minimum value of the pitch, calling it **pitch range**. We also decided to eliminate the sample rate feature, as it was the same for each recording. It is important to note that the same preprocessing operations were applied to both the development and evaluation sets to ensure consistency and comparability in the analysis.

In order to pass the processed data to the models we first split the dataset into training and test sets, with 80% of the data used for training and 20% for testing. Then we applied scaling: the transformed data was only used for models such as Linear Regression and Ridge.

B. Model selection

Several models, including linear algorithms (Linear Regression and Ridge) and pipelines combining linear models with polynomial features, have been tested. KNN and tree-based regression techniques, such as Random Forest Regressor and Extra Trees, were also firstly evaluated without any hyperparameters tuning in order to select the model that demonstrated the best predictive performance on the data.

Model	RMSE
linreg	10.1391
ridge	10.1381
rf	9.8243
extra_trees	9.9229
knn	12.0280
sin+poly5+linreg	12.9109
sin+poly5+ridge	12.9109

TABLE I
Performance of models with their RMSE values.

Based on the results presented in Table I, where tree-based models demonstrated superior performance, the focus was subsequently directed toward refining and analyzing these approaches.

- **Random Forest:** this algorithm builds multiple decision trees using random subsets of data and features. Each

tree independently predicts an outcome, and the algorithm aggregates these predictions through averaging (for regression tasks). By introducing randomness, it reduces overfitting and variance compared to individual trees. Random Forest performance depends heavily on its hyperparameters—tuning these can optimize both accuracy and efficiency for specific tasks. The algorithm doesn't require normalization or scaling of input features since it makes splits based on feature thresholds rather than magnitudes. [3]

- **Extra Trees**: this model builds upon traditional decision tree algorithms by introducing two primary sources of randomness:
  1) **Random Feature Selection:** At each node, a random subset of features is selected from the entire feature set.
  2) **Random Split Points:** For each chosen feature, split points are selected randomly within the feature's range, rather than searching for the optimal split.

  Unlike Random Forests, which select optimal split points, Extra Trees employs a randomized approach for both feature selection and split point determination. This dual randomization reduces variance and mitigates overfitting, especially in datasets with significant noise, while also improving computational efficiency due to its simpler algorithmic design. This model was selected due to its superior performance observed across various hyperparameter tuning scenarios. [4]

For both models, a grid search was conducted to identify the optimal hyperparameters, which will be discussed in detail in the next section.

C. Hyperparameters tuning

Different configurations were tested, as shown in Table II, to optimize the performance of the **Random Forest Regressor** and **Extra Trees Regressor**. The tuning process employed *Grid Search* with 5-fold cross-validation, using *RMSE* as the scoring metric.

Model	Parameter	Values
Random Forest	*n_estimators*	{100, 200}
	max_depth	{10, 20, None}
	min_samples_split	{2, 10}
	min_samples_leaf	{1, 4}
	max_features	{sqrt, log2, None}
	bootstrap	{True, False}
Extra Trees	*n_estimators*	{100, 200}
	max_depth	{10, 20, None}
	min_samples_split	{2, 10}
	min_samples_leaf	{1, 4}
	max_features	{sqrt, log2, None}
	bootstrap	{True, False}

TABLE II
HYPERPARAMETERS CONSIDERED FOR RANDOM FOREST AND EXTRA TREES MODELS.

These configurations were used to train the final optimized models and evaluate their performance on the test set.

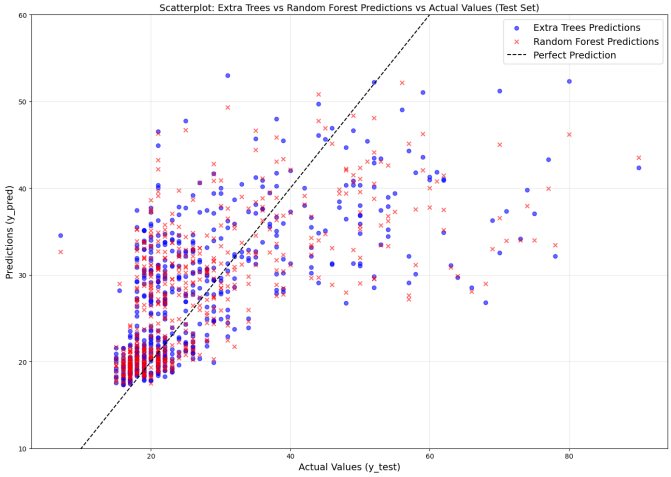

Fig. 4. Extra Trees vs Random Forest Predictions vs Actual Values (Test Set)

III. RESULTS

The best configuration for Random Forest was found for {'bootstrap': False, 'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 200} (RMSE score ≈ 9.93812), whereas the best configuration for Extra Trees was found for {'bootstrap': False, 'max_depth': None, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100} (RMSE score $\approx$ 9.77035). After conducting an extensive grid search, we manually selected the configurations that delivered the best RMSE performance, both on our local evaluation platform and the public scoring system. We trained the best performing Extra Trees and Random Forest regressors on the development data, then we used the trained models to label the evaluation set. The public score obtained is of 9.616 for the Extra Trees and of 9.855 for the Random Forest.

Both models exhibit similar performance on the test set, as shown in Figure 4. Extra Trees, however, performs slightly better, as evidenced by its tighter clustering around the perfect prediction line. Notably, the distribution of the target variable, age, is uneven, with both models demonstrating better accuracy when predicting lower age values. However, as the age variable increases, the models display greater uncertainty in their predictions.

IV. DISCUSSION

The superior performance of Extra Trees compared to Random Forest on the test set suggests that the increased randomness in its decision boundaries allowed for better generalization, particularly in handling the uneven distribution of the target variable. The dataset's skewed age distribution likely caused Random Forest to overfit to dominant patterns in the training data, resulting in reduced performance for underrepresented regions. In contrast, the broader variability introduced by Extra Trees reduced overfitting and made it

more effective at capturing the inherent noise and complexity of the data.

Some aspects worth considering for future improvements include:

- Applying PCA to MFCC feature data and MFCC delta coefficient feature data before combining them into a single matrix. This method has been shown to improve the accuracy of speech recognition systems from 86.43% to 89.29% [5].
- Feature extraction could be performed using VGGish, a convolutional neural network originally designed for visual recognition tasks. VGGish has shown exceptional capabilities in extracting hierarchical and representative features from audio signals, making it a powerful tool for audio analysis [6].

However, the results are highly promising, with Extra Trees demonstrating superior generalization on complex datasets. Future enhancements, such as PCA and VGGish, could further refine performance, but the current work already offers valuable insights.

REFERENCES

[1] V. Tiwari, "Mfcc and its applications in speaker recognition," 2010.
[2] D. Tran Duc, H. Tan, and S. Pham, "Customer gender prediction based on e-commerce data," 10 2016.
[3] L. Breiman, "Random forests," 2001.
[4] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," 2006.
[5] A. Winursito, R. Hidayat, and A. Bejo, "Improvement of mfcc feature extraction accuracy using pca in indonesian speech recognition," 2018.
[6] M. Diwakar and B. Gupta, "Vggish deep learning model: Audio feature extraction and analysis," 2024.