Project Assignment Winter Call, A.Y. 2024/2025

Federico Cerbelli*, Niccolò Tozzi†
Politecnico di Torino
*Student ID: s346453,
†Student ID: s345809,
s345809@studenti.polito.it
s346453@studenti.polito.it

***Abstract*—With the continuous advancement of analysis techniques, speech recognition has become one of the hottest research topics. This technology enables programs to process and understand human speech, opening the door to numerous applications. In this project, we focus on audio analysis, specifically on developing a model capable of predicting the speaker's age. The following sections describe how we worked with the provided features and how we further analyzed the audio data to extract new meaningful attributes.**

I. Problem overview

The task of age prediction is challenging due to variations in speech caused by the speaker's physical characteristics (e.g. weight, height) and emotional state. These factors introduce variability and non-linearities, complicating accurate modeling [1]. A dataset of 3,634 samples is provided for this regression task, divided into two parts: the development set (2,933 samples) and the evaluation set (691 samples). The development set contains the target variable for each sample, making it suitable for model training. Several features have already been extracted (see Tab. I). Alongside for each sample we are provided with the corresponding audio file.

The speakers have different ethnicities which introduces variability in language and sentence structure. Additionally, the duration of the audio recordings varies across samples.

Feature	Description
Sampling rate	In Hz.
Gender	The gender of the speaker.
Ethnicity	The ethnicity of the speaker.
Mean, max, min pitch	In Hz.
Jitter	Pitch variation measure.
Shimmer	Amplitude variation measure.
Energy	Energy of the signal.
ZCR mean	Rate of singnal sign changes
Spectral centroid mean	Centroid of the spectrum.
Tempo	In beats per minute (BPM).
HNR	Harmonic-to-noise ratio.
Num words, num characters	Count of words/characters.
Num pauses	Pauses detected in the speech.
Silence duration	In seconds.
Path	Path to the audio.

TABLE I
Features briefly described.

II. Proposed approach

First of all, it is essential to analyze the dataset. At a glance, it is evident that there are two categorical features. These features must be processed before being fed into the model. Although there are no missing values, the numerical features also need to be examined to understand their distributions and identify any potential issues, such as outliers or inconsistencies in their domains.

After completing the necessary preprocessing steps, we incorporate additional features derived from audio analysis. These features will be explained in the next section. Once these key steps are completed, we split the development set into training and validation subsets. This allows us to fine-tune hyperparameters, build a robust pipeline, and effectively evaluate the performance of our regression models. Our approach involves training and optimizing multiple models to determine which one yields the best predictive performance.

A. Preprocessing

The dataframe includes columns that are irrelevant for training a model to predict age. Specifically, the $sampling\_rate$ column is constant across all samples, and the $path$ of the audio file does not provide any useful information for the prediction. The $tempo$ is stored as a string, which must be converted into a numerical format to be properly utilized by the model. Among all features, we observed a strong correlation between num_words and $num\_characters$, as expected. Given the statistical and logical correlation, we retained only the num_words column to eliminate redundancy. The $silence\_duration$ feature does not appear to provide a measure of periods when no one talks. However, subsequent tests revealed that the model heavily relies on this feature to make inferences. One possible explanation is that the magnitude of $silence\_duration$ is linked with the overall audio length, which can influence the context or structure of the speech, indirectly providing valuable information about the audio's content and patterns.

Another relevant set of features to examine are min_pitch and max_pitch. From the box-plot representation (Fig. 1), it is evident that for min_pitch, some potential outliers lie above the upper whisker. Similarly, for max_pitch, a significant number of points fall below the lower whisker. These values were often associated with short and noisy audio recordings. However, attempts to remove these observations resulted in worse performance. Since the vast majority of samples share similar min_pitch and max_pitch values, and given that the

most extreme values do not appear to be genuine, we decided to remove these two features.

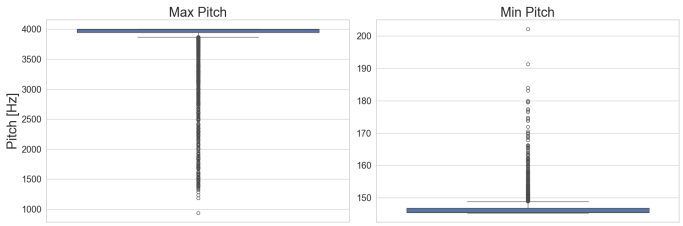

Fig. 1. Box-plot of max_pitch (left), Box-plot of min_pitch (right)

Another preprocessing step we adopted was extracting additional features from the raw audio signals. Given that many recordings were noisy, we applied a filter to the signals before feature extraction. To avoid losing or altering any frequency information of the speech, we used a low-pass Butterworth filter, which provides a flat passband with a 3dB attenuation at the cutoff frequency, set at 4500 Hz.

For feature extraction, we used *librosa* [2]. Specifically, we extracted:

- Frequency Cepstral Coefficients (MFCCs): We calculated 13 MFCCs using analysis windows of 16,384 samples. The features used are the mean of each extracted coefficient over all windows. MFCCs are widely regarded as effective features for speech analysis and age estimation, as they provide a compact representation of the audio's power spectrum. They are derived using a discrete cosine transform of the logarithmic power spectrum of the audio. Numerous studies have shown that MFCCs consistently help reduce prediction error in age estimation models [3].
- Chroma: Similarly, we extracted 12 chroma features. Chroma features, typically used to represent pitch class, capture the tonal content of an audio signal in a compact form. While most commonly applied in musical analysis, they have also been shown to enhance performance in speech processing models. [4]

With regard to the categorical features, the dataset includes Ethnicity and Gender. The Ethnicity column originally contained 165 unique values. This categorical values would need to be encoded. To reduce the dimensionality of the problem, we grouped ethnicities with fewer than 100 observations into a single category labeled "Other", except for English, Arabic, and Igbo, which had 579, 102, and 1081 observations, respectively. This approach was further motivated by the fact that in the evaluation set, we could encounter ethnicities that were not present in the training set. This way, we could easily map those ethnicities to "Other." We applied one-hot encoding to both Gender and Ethnicity to make them suitable for model processing.

Finally, we observed that the distribution of the target variable (age) is highly positively skewed ($\tilde{\mu}_3 = 1.79$), as shown in Fig. 2. The dataset contains a higher proportion of younger speakers, with fewer samples representing older age groups. We decided not to transform the target variable. While transformations can sometimes help normalize skewed distributions, we believe that preserving the original age distribution is important for maintaining intuitive and interpretable model predictions. However, we acknowledge that the imbalance in the target variable may introduce bias and hinder the model's ability to generalize effectively across all age ranges. Therefore, it is crucial to account for this issue when building the models to ensure fair performance across all age groups.

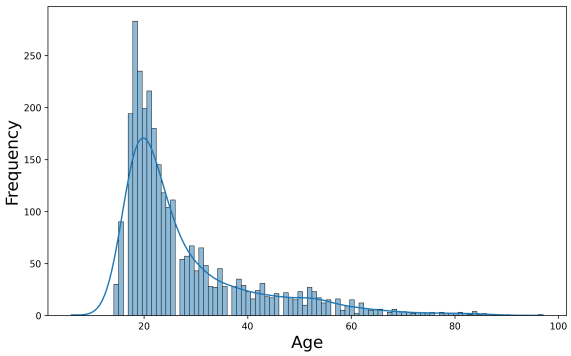

Fig. 2. Distribution of age in training data

B. Model selection

We decided to focus on several models that could be suited for handling the relationships in the data: Support Vector Regression (SVR), K-Nearest Neighbors (KNN) and Random Forest (RF).

- SVR, particularly with non-linear kernels, is effective at capturing non-linear relationships between features. Since SVR relies on distances between data points to make predictions, feature scaling is a necessary step. The regression problem can be seen as a generalization of the classification problem, where the model returns a continuous-valued output. This extension of Support Vector Machines (SVM) aims to find the ε-insensitive tube that best approximates the continuous-valued function, balancing model complexity and prediction error. A flexible tube is formed symmetrically around the estimated function, such that the absolute values of errors less than a certain threshold ε are ignored. Only the points outside the tube are penalized. The hyperplane in SVR is defined in terms of support vectors, which are the training samples that lie outside the boundary of the tube. As in SVM, the support vectors in SVR are the most influential instances that affect the shape and position of the tube. [5].
- KNN regressor is a simple yet powerful non-parametric algorithm that makes no assumptions about the underlying distribution or functional relationship between the variables. The algorithm classifies a new observation by identifying its k nearest neighbors from the training set and averaging the values for regression among these

neighbors. Its performance is highly dependent on the choice of distance metric, the number of neighbors (k), and appropriate feature scaling, as it is sensitive to differences in feature magnitude.

- RF is an ensemble learning method that constructs multiple decision trees and combines their predictions. RF enhances robustness through two key techniques. First, Bootstrap Aggregation (Bagging) creates multiple decision trees by randomly sampling the training set with replacement, helping to mitigate overfitting. Second, Feature Bagging (Random Feature Selection) ensures that each tree considers a random subset of features at each split, reducing correlation between trees and improving generalization. These techniques make RF particularly effective at handling noisy data and outliers. Since decision trees process each feature independently, they are unaffected by variations in feature range.

For all three models, the imbalanced distribution of ages could pose a challenge. To address this issue, we decided to apply class weights. For RF we divided the samples into two age groups: those with $age > t$ and those with $age < t$, where t stands for threshold. For each sample is assigned a weight calculated as $1 - f$, where f denotes the relative frequency of its respective group. To determine the partition threshold, we visualized all predictions in a 2D space, where each point was represented by its coordinates ($true\_age$, $predicted\_age$). With the ideal model, all points would lie perfectly along the bisector of the first quadrant. This visualization revealed that the model consistently underestimated older men/women years. Heuristically, we identified 45 as the age threshold. To confirm the threshold choice and evaluate the need for additional partitions, we conducted various tests. However, this specific configuration yielded the best results, and no further thresholds improved the model's performance.

In contrast, this configuration was less effective for the SVR, due to the significant disparity between the weights of age classes. To address this, we used four balanced age intervals: 0-20, 20-30, 30-45, and 45+. These intervals led to improved performance for the SVR.

For KNN, we did not assign weights based on age groups. Instead, we only experimented with distance-based weighting as a model hyperparameter.

C. Hyperparameters tuning

All three models require hyperparameter tuning to achieve optimal performance.

To fine-tune the models, we used GridSearchCV from scikit-learn, which performs an exhaustive search over a specified parameter grid. This method identifies the best configuration based on the cross-validation score. As the scoring metric we used the mean squared error.

- RF that was trained with 1000 estimators. The hyperparameters we tested are:
  - Minimum number of samples required to split an internal node, by increasing it we decrease the model complexity. Values: $[2, 5, 10]$
  - Maximum depth of the tree: This controls how deep each decision tree can grow. Limiting the depth helps prevent overfitting. Values: $[15, 20, 30]$
  - Minimum number of samples required to be at a leaf (terminal) node. A higher value could reduce overfitting. Values: $[3, 5]$
  - The fraction of samples to draw for training each base estimator: controls the size of the bootstrap sample used for training each tree (max samples). Adjusting this parameter influences the diversity among the trees. Values = $[0.5, 0.65]$.
  - The fraction of features, randomly sampled, than can be used for each split (max features). Values = $[0.5, 0.65]$.
  - Criterion for the split: defines the metric used to evaluate impurity at each node/split. We tested $Poisson$ and $SquaredError$.

- For KNN we tested:
  - The $Euclidean$ and $Manhattan$ distance.
  - Two configurations for the weights: one where each point in the neighborhood is equally important ($uniform$) and another where closer points have greater influence (weights are assigned as the inverse of $distance$).
  - Number of neighbors (K): The value of this hyperparameter determines the number of closest points that contribute to the prediction. A small K implies grater sensitivness to noise points, a large K could lead to underfit by oversmoothing the results. Values: $[10, 15, 20, 25, 30, 35]$.
  - Number of PCA components used to reduce high dimensionality. We selected a fraction of components that capture between 50% and 85% of the total explained variance to minimize information loss. Values: $[0.5, 0.65, 0.75, 0.85, None]$, where None means keeping all components.

- For the fine tuning of SVR we tested:
  - ε values: $[1, 4, 6, 8]$
  - The regularization parameter C. A larger value gives more weight to minimizing the error, but increase the risk of overfitting. Values: $[1, 5, 8, 10, 20, 30]$
  - Radial Basis Function (rbf) and Polynomial ($poly$) Kernels , treating the kernel choice as a hyperparameter. rbf and $poly$ further generalizes the approach by mapping the data into higher-dimensional spaces, allowing the model to capture non-linear relationships.
  - The degree of the polynomial kernel function. This parameter is ignored for rbf. Values: $[2, 3, 4]$

III. Results

We compared the models on a dedicated test set, a reserved partition of the development set that was not involved in

the training process. This test set allowed us to assess the models' ability to generalize to new, unseen data and to validate the stability and effectiveness of the hyperparameter optimization conducted during cross-validation. The optimal hyperparameters for each model are highlighted in Tab. II. To make the evaluation more robust, we tested the models using five different random states, each representing a different split of the training and test sets. The models' performance was measured using the Root Mean Squared Error (RMSE) over the predictions, a commonly used metric that quantifies the average magnitude of the errors between predicted and actual values, providing a clear indication of prediction accuracy. The results, including the mean performance, are presented in Tab. III.

Model	**Optimal Hyperparameters**
SVR	ε: 4, C: 30, Kernel: rbf, Degree: None
KNN	Distance Metric: $Euclidean$, Weights: $distance$, K: 25, PCA: None
Random Forest	Min Samples to Split: 5, Min Samples per Leaf: 2, Max Depth: 20, Max Features: 0.65, Max Samples: 0.65, Criterion: $Poisson$

TABLE II
OPTIMAL HYPERPARAMETERS FOR EACH MODEL, DETERMINED THROUGH CROSS-VALIDATION.

Model	**RS 42**	**RS 10**	**RS 100**	**RS 2**	**RS 345809**	**Mean**
Random Forest	9.53	10.46	9.68	10.12	9.83	9.92
SVR	9.39	10.00	9.36	9.92	9.61	9.66
KNN	10.03	10.77	10.23	10.74	10.25	10.40

TABLE III
MODEL PERFORMANCE COMPARISON ACROSS 5 RANDOM STATES. THE NUMBERS IN THE TABLE REPRESENT RMSE VALUES.

Finally, since SVR and RF outperformed KNN, we decided to discard the latter and focus on testing the two best-performing models. Both were evaluated on the reserved evaluation set to obtain the final public score. The two models, with respective RMSE values of 9.440 and 9.558, demonstrated consistent behavior on this new dataset.

IV. DISCUSSION

Observing the results, it is clear that SVR and RF outperform KNN, which heavily relies on local neighborhoods and often struggles with high-dimensional data. KNN struggles with imbalanced datasets, as it tends to favor majority classes, which can skew predictions.

Also retaining noisy data to minimize information loss may have introduced more noise, further degrading its predictive accuracy.

Interestingly, the model without PCA consistently performed the best, likely because PCA excels with linear correlations but struggles with capturing more complex, nonlinear relationships.

During our tests, RF demonstrated its versatility. By employing bootstrap aggregation and feature bagging, it effectively resists overfitting when hyperparameters are carefully tuned. One of the limitations of the model is the underestimation of older ages (see Fig.3). Although this is mitigated by the weight-based approach, it remains evident. This problem is due to the scarcity of samples for older ages.

SVR is, by a slight margin, the model that achieved the best results with the test set. When combined with the weighting strategy to address the unbalanced dataset, it outperformed the other models. Nevertheless, like RF, it still struggled with predicting older ages accurately, with this issue remaining evident (see Fig.4). Since the error in predicting older ages appears systematic, future efforts could focus on addressing this limitation. One potential improvement could be the application of SMOTE (Synthetic Minority Over-Sampling Technique) or similar techniques to oversample the underrepresented age groups.

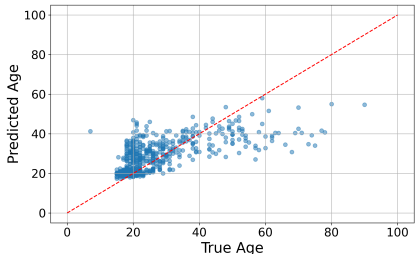

Fig. 3. Example of Predicted age vs. True Age obtained by using RF with the optimal version of hyperparameters and random state equal to 42.

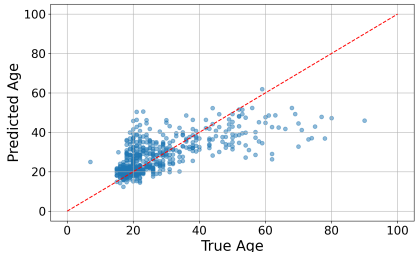

Fig. 4. Example of Predicted age vs. True Age obtained by using SVR with the optimal version of hyperparameters and random state equal to 42.

Both RF and SVR are strong contenders for regression tasks. However, They have limitations in the field of speech processing and age estimation, particularly when compared to more advanced approaches commonly used in these domains. Neural networks, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), are widely considered state-of-the-art for speech recognition, especially when large training datasets are available. From the perspective of simplicity and lower computational cost, RF and SVR could represent practical and competitive choices for certain applications.

REFERENCES

[1] A. A. Abdulsatar, V. V. Davydov, V. V. Yushkova, A. P. Glinushkin, and V. Y. Rud, "Age and gender recognition from speech signals," *Journal of Physics: Conference Series*, vol. 1410, p. 012073, dec 2019.
[2] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," 2015.

[3] W. Spiegl, G. Stemmer, E. Lasarcyk, V. Kolhatkar, A. Cassidy, B. Potard, S. Shum, Y. C. Song, P. Xu, P. Beyerlein, J. Harnsberger, and E. Nöth, "Analyzing features for automatic age estimation on cross-sectional data," in *Interspeech 2009*, pp. 2923–2926, 2009.

[4] A. Fidan, R. O. Bircan, and S. Karamzadeh, "A new approach for age estimation system based on speech signals," in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 388–393, 2021.

[5] M. Awad and R. Khanna, "Support vector regression," in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pp. 67–80, Apress, 2015.