Estimating the age of people from vocal recordings

Bello Renato
Politecnico di Torino
Student id: s341965
s341965@studenti.polito.it

Chiodo Martina
Politecnico di Torino
Student id: s343310
s343310@studenti.polito.it

***Abstract*—In this report we tried to solve a regression problem, the goal is to estimate the age of a person from a vocal recording. We used both the csv dataset and the vocal recording to extract all the features needed to make the regression.**

I. Problem overview

The proposed competition is a regression problem about age estimating; in fact, based on vocal recordings and some information regarding the person, such as his ethnicity and his gender, we aim to correctly determine the age of the person who is talking. The data set is divided into two parts:

- a *development* set, containing 2933 elements, each of them labelled
- an *evaluation* set, containing 691 elements

The development set will be used to build a regressor and the estimate will be done on the evaluation set.

We can make some considerations based on the development set. First, the dataset is complete, indeed none of the rows contains missing values. Second, the dataset also contains the path to the vocal recordings from which we have decided to extract more features from the spectrogram. Third, the sample rate is the same for all the audio recordings.

To better understand the distribution of the features we can plot some histograms. From the histograms shown in figure (1) we can notice that most of the features seems to be distributed as Gaussian distribution with not many outliers. Some exceptions are *max_pitch*, *min_pitch* and *num_characters*: the first two features mentioned have a very wide range of values but almost all their distribution mass is concentrated in a single point, thus there are many outliers; on the contrary, *num_characters* concentrates his mass distribution in two far values.

Another useful tool is the correlation plot shown in figure (2): from this plot we can spot that many features are highly correlated and this may yield redundancy into the dataset.

Another thing worth mentioning is that the values of our response variable, *age*, are not uniformly distributed. As shown in the histogram, most of the recordings are of people in the age range $[15, 35]$. This would probably affect the performance of our regressor because the model might become biased towards predicting ages within this range, potentially leading to less accurate predictions for ages outside this range.

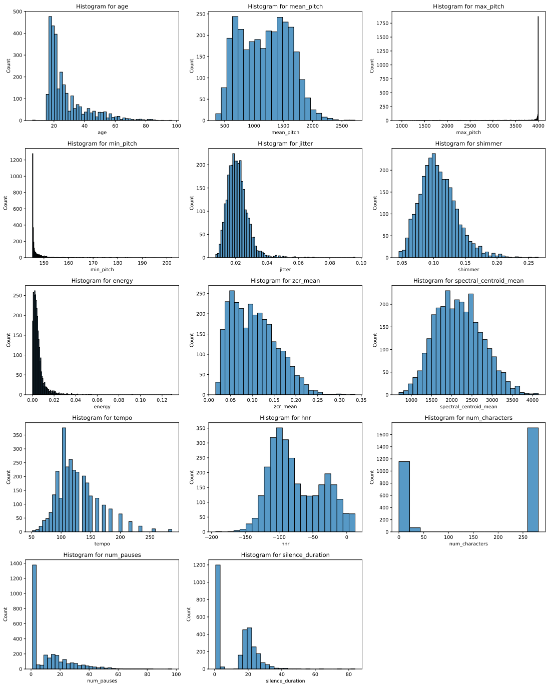

Fig. 1: Histograms of some features.

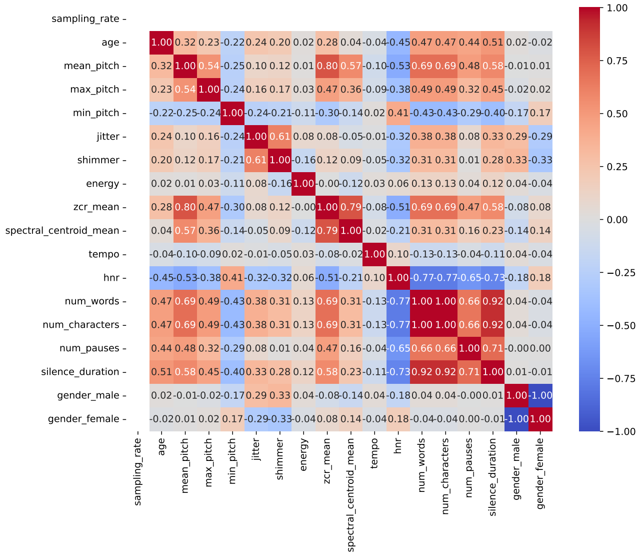

Fig. 2: Correlation among features.

II. Proposed approach

A. Preprocessing

We easily noticed that some features among the one given requires some preprocessing: *ethnicity* and *gender* are a categorical features and, thus, they need to be encoded in order to be used in the regression; *tempo* is not automatically saved as a float type because the values are enclosed by parenthesis.

Then, we decided to take care of the redundancy spotted in the correlation plot by determining the pairs of features which have correlation 1. For example, from the correlation plot (Figure (2)) we can notice that *num_words* and *num_characters* shares correlation 1, thus we decided to just keep one of them to avoid redundancy.

We also decided to take advantage of the audio recordings we were given and thus we extracted additional features from them. First, we extracted the Mel Frequency Cepstral Coefficients [1], which are largely used in biometric application, such as the recognition of a speaker or some of his characteristics, because of their capacity of encoding relevant linear and non-linear features of the speech. Then we also extracted the spectrogram, partitioned it in chunks and extracted mean and variance of each of them.

After these steps of preprocessing, we noticed that the total number of features has increased considerably and it may be possible that some of them are highly correlated. Therefore, a feature selection is necessary in order to extract a subset of relevant features. This aspect will be better presented later on.

We concluded the preprocessing by applying some transformation to the values of the features. First, we applied a log transformation to some feature due to the extreme number of outliers spotted by observing at the box plot of some numerical features, as shown in Figure 3. The feature transformed are: *num_pauses*, *energy*, *jitter*, *min_pitch*, *max_pitch* (this transformation gained an improvement of the RMSE during the test evaluation).

Then, we proceeded by standardizing the dataset. It is known that machine learning estimators may poorly perform if the data are not standardized because features with higher variance in space may dominate the objective function.

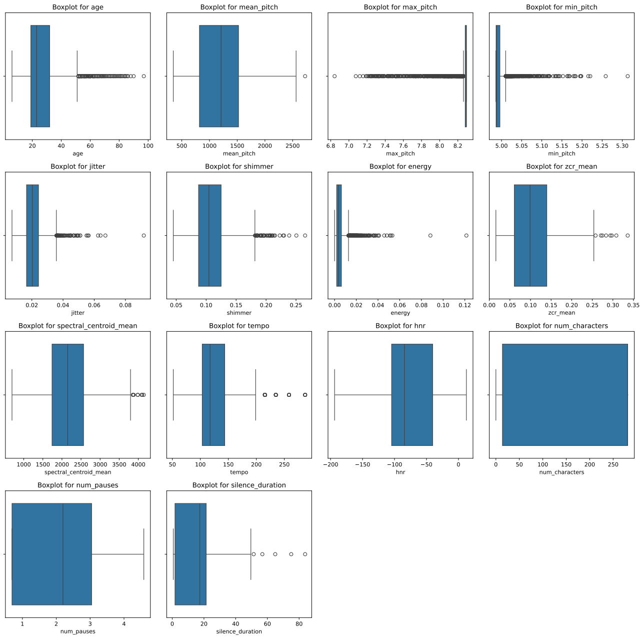

Fig. 3: boxplot of numerical features

B. Model selection

We choose 2 models for testing our regression.

- *Ridge regression* [2]: is a particular linear regression where the objective function to minimize is defined as:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right) + \lambda\|\mathbf{w}\|_2^2 \tag{1}$$

  Where y_i is the target variable, $\mathbf{x}_i$ are the characteristics and $\mathbf{w}$ are the coefficients of the regression. Instead of the classical linear regression, in this function there is a term ($\lambda\|\mathbf{w}\|_2^2$) that imposes a penalty proportional to the squared L_2 norm of the coefficients to encourage the reduction of their overall magnitude.
- *MLP regression* [3]: this regression id based on a *neural network* that uses the *MSE* (2) (*mean squared error*) as a *loss function*, which is minimized using a stochastic gradient method and the *backpropagation*.

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

  In this formula $\hat{y}_i$ represents the prediction made by the neural network for the target variable y_i.

For both classifiers, the best working configuration of hyperparameters has been identified through a grid search, as explained in the following sections.

C. Hyperparameter tuning

There are two main sets of hyper-parameters to be tuned:

- The number of features to use to fit each model is indicated by n_{RI} and n_{MLP}.
- Ridge regressor and MLP regressor parameters

For tuning the parameters we considered a split of the development set into *train* and *validation* set, containing the 80% and the 20% of the points.

For convenience we assume that two set are independent, therefore we can first optimize the number of features used to fit the regressor and then perform some tuning of the hyperparameters of the two regressor.

- *Ridge parameters*: Before finding the best parameters using a grid search, we found the best features for fitting the regression. We used a *scikit-learn* class called *RFE* [4] (*recursive features elimination*), to execute this task. We trained and tested the ridge model increasing the feasible number of features each time, obtaining that the best number of features is 50, as shown in Figure 4. The feature selection has been performed by using the default parameters.

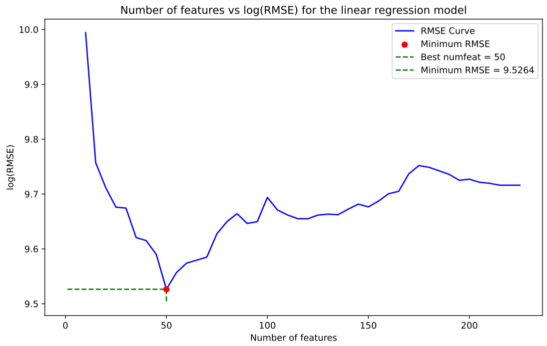

Fig. 4: number of features vs log(RMSE) for the Ridge regression

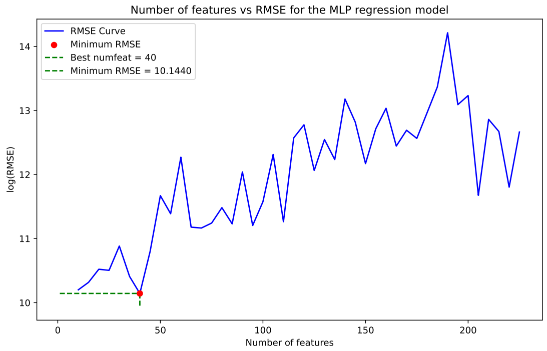

Fig. 5: number of features vs log(RMSE) for the MLP regression

- *MLP parameters*: Before finding the best parameters using a grid search, we found the best features for fitting the regression. For the *MLP regressor* we used a *scikit-learn* class called *SelectKbest* [5], to execute this task. We trained and tested the *MLP regressor* increasing the feasible number of features each time, obtaining that the the best number of feature for fitting the regression is 40, as shown in Figure 5. The feature selection has been performed using, as initial guess for the MLP Regressor, the following parameters: *hidden_layer_sizes* = (50,), *activation* = *relu*, *alpha* = 0.01 and *max_iter* = 2000.

After finding the best features for the two models, we applied a grid search for finding the best hyper-parameters, those are shown in the table I.

We incorporated the grid search with a 5-fold cross-validation, to ensure greater model robustness. The parameters have been selected by choosing the ones which minimized the mean squared error.

III. Results

From applying the pipeline we have just discussed, we obtained the following hyper-parameters for Ridge Regressor

- $n_{RI} = 50$
- *alpha*: 2
- *fit_intercept*: *True*
- *max_iter*: *None*

TABLE I: Hyperparameters considered for the grid-search (n_{tot} is the total number of features)

Model	Parameter	Values
Feature Selection	n_{RI}, n_{MLP}	10 → n_{tot}, step 5

Ridge Regressor	alpha
	fit_intercept
	max_iter
		{0.5, 1, 2}
		{True, False}
		{None, 500, 1000}

MLP Regressor	hidden_layer_sizes
	activation
	solver
	alpha
	max_iter
		{(100,), (100,2), (50,)}
		{tanh, logistic, relu}
		{adam, lbfgs, sgd}
		{0.001, 0.01, 1}
		{1000, 2000}

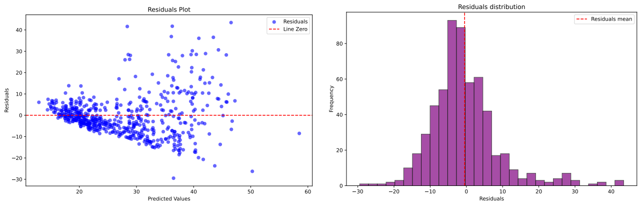

Fig. 6: Residuals plot and distribution for the Ridge Regressor

while for the MLP Regressor the hyper-parameters are

- $n_{MLP} = 40$
- *hidden_layer_sizes*: (100,)
- *activation*: *logistic*
- *solver*: *adam*
- *alpha*: 1
- *max_iter*: 1000

Once defined the quantities n_{RI} and n_{MLP} and tuned the hyper-parameters using a portion of the development set, we validated the resultant regressors on the other part of the dataset.

A first analysis of the performance of the model can be done by looking at the residuals, that is the quantity $\epsilon_i = y_i - \hat{y}_i$. Theoretically, they should represent the white noise in data and thus should be distributed as

$$\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

Looking at the behaviour of the residuals we obtained, shown in (6) and (7), we can appreciate that they are distributed around the mean value 0 for both models, but the hypothesis of homoscedasticity is not satisfied. The cause could be a lacking representativeness of people over the age of 50 in the dataset, as noticed during the data exploration and confirmed by the fact that most of the residuals are positive, meaning that the two regressors often predict a smaller age when the true one is over 35.

Moreover, we sadly notice that the variance of the residuals is very large especially when predicting larger values of the response variable. Comparing the two plots, it has to be said that the MLP regressor is the one whose residuals have greater variance.

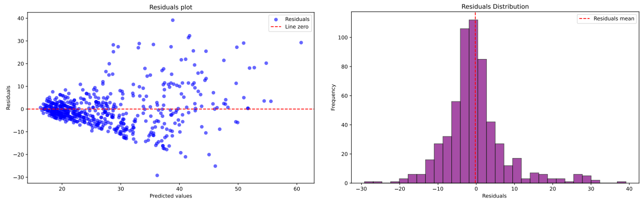

Fig. 7: Residuals plot and distribution for the MLP Regressor

During the phase of validation, we decided to evaluate the model not only in terms of RMSE but also by computing the R^2 score: the scores obtained are of 0.4520 for the Ridge regressor and of 0.3927 for the MLP regressor. The fact that the MLP Regressor is scored with a lower R^2 value restates what was shown in the residuals plots, that is that the MLP regressor residuals have higher variance. This allows us to say that the solution we have found works slightly better than just predicting the mean value of the response variable. Still, they are pretty far from their maximum value 1 which means that the regressors fail to capture great part of the variance of the data.

The public score obtained is 9.974 for the Ridge Regressor and 9.726 for the MLP Regressor. As these represent the first scores derived from the evaluation data, overfitting is unlikely, and it is reasonable to expect similar results on the private score.

IV. Discussion

The proposed approach obtains results that outperform, even if not by much, the naive baseline defined. We have empirically shown that the selected regressors perform similarly for this specific task, achieving satisfactory results in terms of RMSE.

However, we acknowledge that the results obtained from the regressors were not so accurate, this inefficiency arises from the fact that the target value is not well distributed, as shown in Figure 1. As a result, prediction for bigger values of age are not as precise as prediction for lower values of age, as shown by looking the residuals in Figures 6 and 7. It is highly likely that, using a dataset that is less imbalanced in the target variable, the results would be better.

An other idea for computing better results was splitting the dataset by the gender, in fact the voice is different for different genders. We tried to train two regressors, one for male and the other for female gender, however, the results were suboptimal. This type of strategy would have been successful if the dataset had been more complete.

References

[1] Z. K. Abdul and A. K. Al-Talabani, “Mel frequency cepstral coefficient and its applications: A review,” *IEEE Access*, vol. 10, pp. 122136–122158, 2022. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9955539.

[2] “Ridge regression,” https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.Ridge.html.

[3] “Mlp regression,” https://scikit-learn.org/1.6/modules/generated/sklearn.neural_network.MLPRegressor.html.

[4] “Rfe,” https://scikit-learn.org/1.6/modules/generated/sklearn.feature_selection.RFE.html.

[5] “Selectkbest,” https://scikit-learn.org/dev/modules/generated/sklearn.feature_selection.SelectKBest.html.