Age Classification: an IML approach

Statistical Learning 2021

G14

Domenico Mattia Cinque	1784965
Liù Arioli	1812888
Nicola Calabrese	1797714
Michelangelo Saveriano	1823326

1 Abstract

In this project we would like to propose a classification method that discriminates between different age classes based on face image analysis and then study it in an Intepretable Machine Learning perspective. The algorithm classifies subjects into 4 different age categories. We use the *UTKFace* dataset and we based our preprocessing and features extraction phases on [1]. A Random Forest model has been used to classify the images. Finally, we use the `SHAP` library for the interpretability of the model's results.

2 Data description

UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. The labels of each face image is in the format [`age`]_[`gender`]_[`race`]_[`date&time`].jpg where

- [`age`] is an integer from 0 to 116, indicating the age
- [`gender`] is either 0 (male) or 1 (female)
- [`race`] is an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern).
- [`date&time`] shows the date and time an image was collected.

For our analysis we keep only the fields `age`, `gender` and `race` and we decide to leave out the last field `date&time` because it is not meaningful to our analysis, but we think that it can be useful in some cases for discriminate the quality of the images and then tune future parameters such as `Canny` thresholds or kernel dimensions of the Gaussian filter for blurring. Moreover, for each image we have a text file with the landmarks of the faces in the format of a list of (x, y) values.

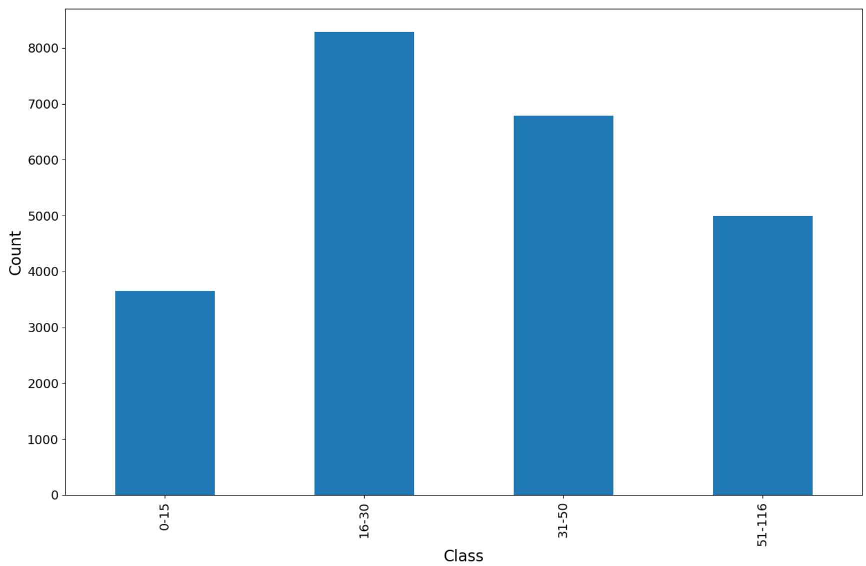


Figure 1: Classes distribution

3 Implementation

3.1 Pre-processing

For the Test set we tried to replicate the same steps made on *UTKFace* dataset, we have collected an hundred of personal photos, labeled them in the same way, provides the correspondingly aligned and cropped faces and extract the same landmarks defined above using a pre-trained neural network implemented in the `dlib` library [3] (likely the same used in *UTKFace* since the extracted points are the same). In this way we have standard face images for feature extraction.

(a) Original image (b) Face cropping (c) Landmarks

Figure 2: Landmarks extraction for test images

3.2 Feature Extraction

Face anthropometry The first features that we extracted are some key ratios in the face, as shown in [1]. These ratios provides an overview of the face anthropometry and are useful because many studies have shown that the face shape changes with ages. If $D(A, B)$ is the Euclidean distance between A and B, the ratios are defined as follows:

$$\text{Ratio 1} = \frac{D(\text{Left Eye, Right Eye})}{D(\text{Middle of Eyes, Nose})}$$

$$\text{Ratio 2} = \frac{D(\text{Left Eye, Right Eye})}{D(\text{Middle of Eyes, Lip})}$$

$$\text{Ratio 3} = \frac{D(\text{Left Eye, Right Eye})}{D(\text{Middle of Eyes, Chin})}$$

$$\text{Ratio 4} = \frac{D(\text{Middle of Eyes, Nose})}{D(\text{Middle of Eyes, Lip})}$$

$$\text{Ratio 5} = \frac{D(\text{Middle of Eyes, Lip})}{D(\text{Middle of Eyes, Chin})}$$

$$\text{Ratio 6} = \frac{D(\text{Left Eye, Right Eye})}{D(\text{Top of Head, Chin})}$$

$$\text{Ratio 7} = \frac{D(\text{Left SoF, Right SoF})}{D(\text{Top of Head, Chin})}$$

In addition to these ratios we computed the distances between every couple of a subset of landmarks as shown in figure:

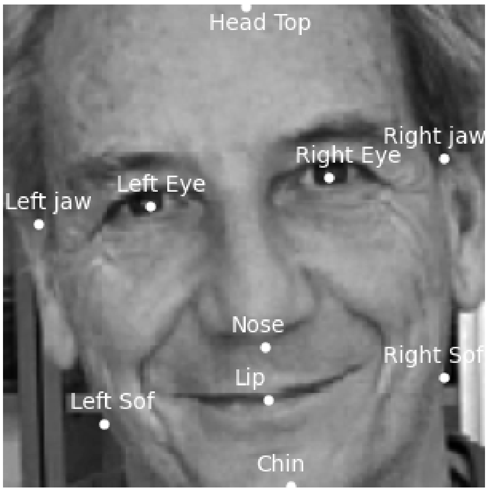
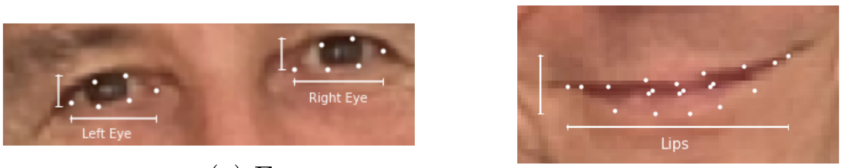


Figure 3: Main Landmarks

The number of these new features is $\binom{10}{2} = 45$ but the most of them are useless. To find the most informative, we trained a model using all the features and analyzed them using `SHAP` (a tool that will be explained more in detail later). These distances are obviously normalized dividing the x-coordinates of each point by the width of the face and the y-coordinates by the height of the face.
Another feature that we added is a measure of the dispersion of the eyes and the lips:

$$\text{Eye Dispersion}_x = \frac{s(\text{Left Eye Landmarks}_x) + s(\text{Right Eye Landmarks}_x)}{2}$$
$$\text{Eye Dispersion}_y = \frac{s(\text{Left Eye Landmarks}_y) + s(\text{Right Eye Landmarks}_y)}{2}$$
$$\text{Lip Dispersion}_x = s(\text{Lip Landmarks}_x) \qquad \text{Lip Dispersion}_y = s(\text{Lip Landmarks}_y)$$

where $s(\cdot)$ denotes the sample standard deviation. This can give us a sense of the shape of mouth an eyes (e.g: children tends to have *rounder* eyes so $\text{ED}_x \approx \text{ED}_y$).



(a) Eyes

(b) Lips

Figure 4: *Dispersion*

Wrinkles We decided to tackle the wrinkles identification problem as a classic edge detection. To keep things simple we used the `Canny` edge detector. It computes the magnitude of the gradient in each point of the image $\mathbf{I}$

$$\|\nabla \mathbf{I}\| = \sqrt{\mathbf{I}_x^2 + \mathbf{I}_y^2}$$

where $\mathbf{I}_x$ and $\mathbf{I}_y$ are the partial derivatives along the two axis. Then we set to 1 all the pixels with magnitude values bigger than a certain threshold t_2 and to 0 all the ones with values less than another threshold t_1. For the pixels that are in between these two levels, we set them to 1 if and only if they are next to other pixels marked as edge.

(a) Whole image

(b) Five areas for wrinkle analysis

Figure 5: Edge Detection

After applying the Canny detector we extracted the wrinkles density of the areas highlighted above: sides of the eyes, forehead and cheeks.

(a) Forehead

(b) Cheeks

(c) Sides of Eyes

Figure 6: Wrinkles of interest

We performed some tuning to decide which values to choose for (t_1, t_2).
In the set of our test images we applied beforehand a Gaussian filter in order to reduce the quality of the images to a level that is comparable to the ones in our train set.

4 Results

We want to predict 4 classes: child with age $\in [0, 15)$; young adults $\in [15, 30)$; adults $\in [30, 50)$; elders $\in [50, \infty)$. After some trials, the algorithm that we chose is a `SVC` with a `rbf` kernel, $C = 100$ and $\gamma = 1$. We achieved an accuracy of 62% on the UTK test set. On our test set we got incredibly worse results, in fact the accuracy was of 35%! We think that this is in part due to the weakness of the algorithm but also to the low quality of the training set.

	Precision	**Recall**	**Sample size**
Children	0.82	0.70	1209
Young adults	0.61	0.73	2717
Adults	0.49	0.47	2232
Elders	0.70	0.58	1666

Table 1: Classification results on UTK

	Precision	**Recall**	**Sample size**
Children	0.25	1.00	16
Young adults	0.91	0.18	55
Adults	0.33	0.33	3
Elders	0.33	0.14	7

Table 2: Classification results on our Test set

The figure below shows the predicted classes for each different age, the blue distribution (children) proves to be the most accurate, in fact it is concentrated in a small area $[0, 20]$ with few outliers outside this range. The others classes instead are more widespread along all the possible age values (it seems that one centenarian is seen as a sparkly teenager!!).

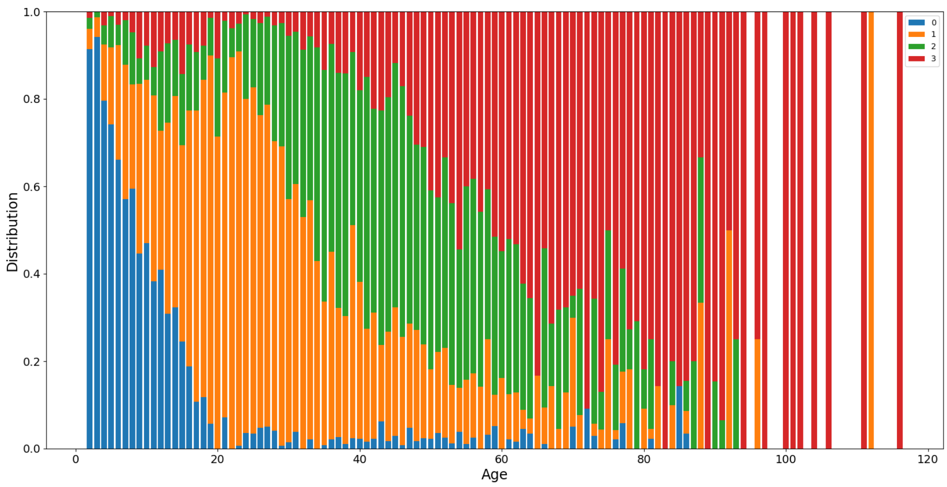


Figure 7: Predicted classes along all ages

If we want to see the same figure but with a probabilistic point of view, here are shown for each age the most likely class to be predicted with its value on the y-axis. Higher bars means more *confidence* in predicting that value. It is clear how the model struggles with adults and in the transition between class 0 and class 1.

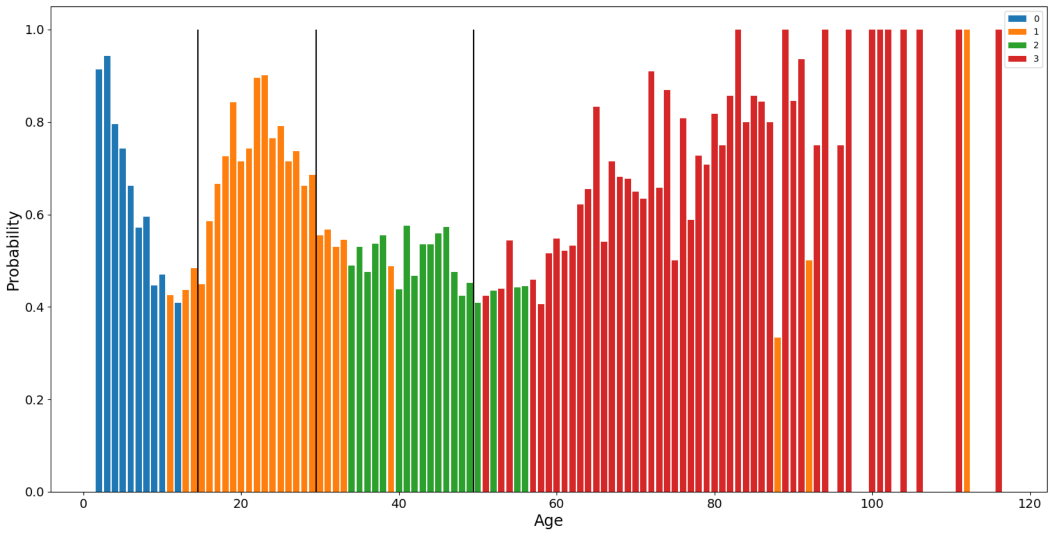


Figure 8: Most likely predicted class along all ages

5 IML

We decided to use the SHAP framework [2] to interpret our model. The main idea behind this framework is to evaluate the effect of each feature to the prediction of a particular output. This is achieved by computing the so called *Shapley* values: these are a concept borrowed by Game Theory and they represent in some sense the contribution of each feature (player) to the final prediction of the model (game). In order to isolate the contribution of each single feature on the model, sub-models are trained over all the possible combinations of features (for M features, 2^M models are trained). For a deeper explanation of how the *marginal contribution* of each feature is computed and how these are wrapping up to obtain the *SHAP values* we suggest to refer to [5] Having the predictions returned by all these models, we can see how each feature influences the final prediction.

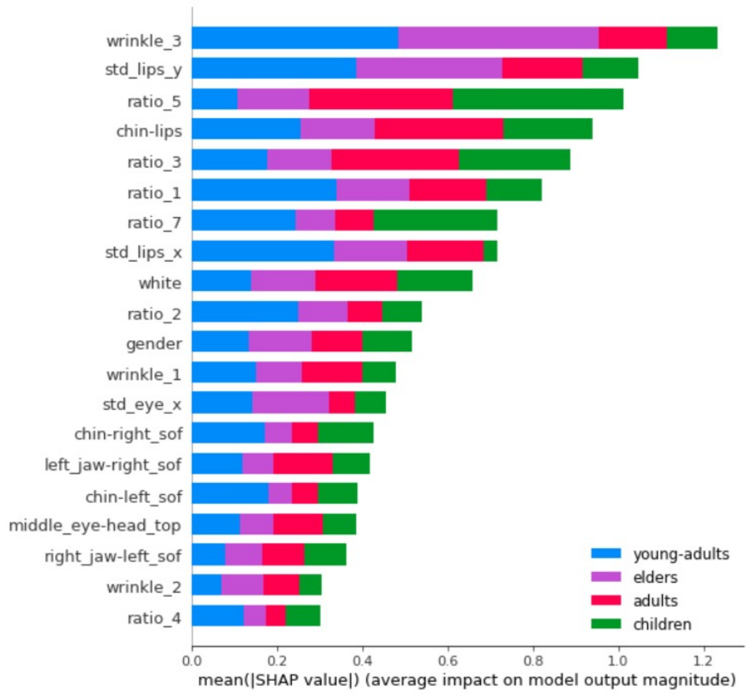


Figure 9: Most important features

As we can see the `wrinkle_3` is the features that higher global importance, in particular for *Elders* and for *Young Adults*, instead the `ratio_5` which is related to the roundness of the face is most important for *Children* and *Adults* in fact studies related to craniofacial growth have shown that the face shape changes from circular to oval as a person grows.

Below there is the `pairplot` of the 5 most important features according to `SHAP`. It is evident that these features provide some separation between the classes, as shown by the color gradient. For example, looking at plot in top-left position, we can see how, as the age increase, the density of wrinkles shifts towards higher values. Also the plot in the top-right position is interesting: it shows how children tends to have less wrinkles and larger faces (higher values of `ratio_3`), contrary to elders (as it should be).

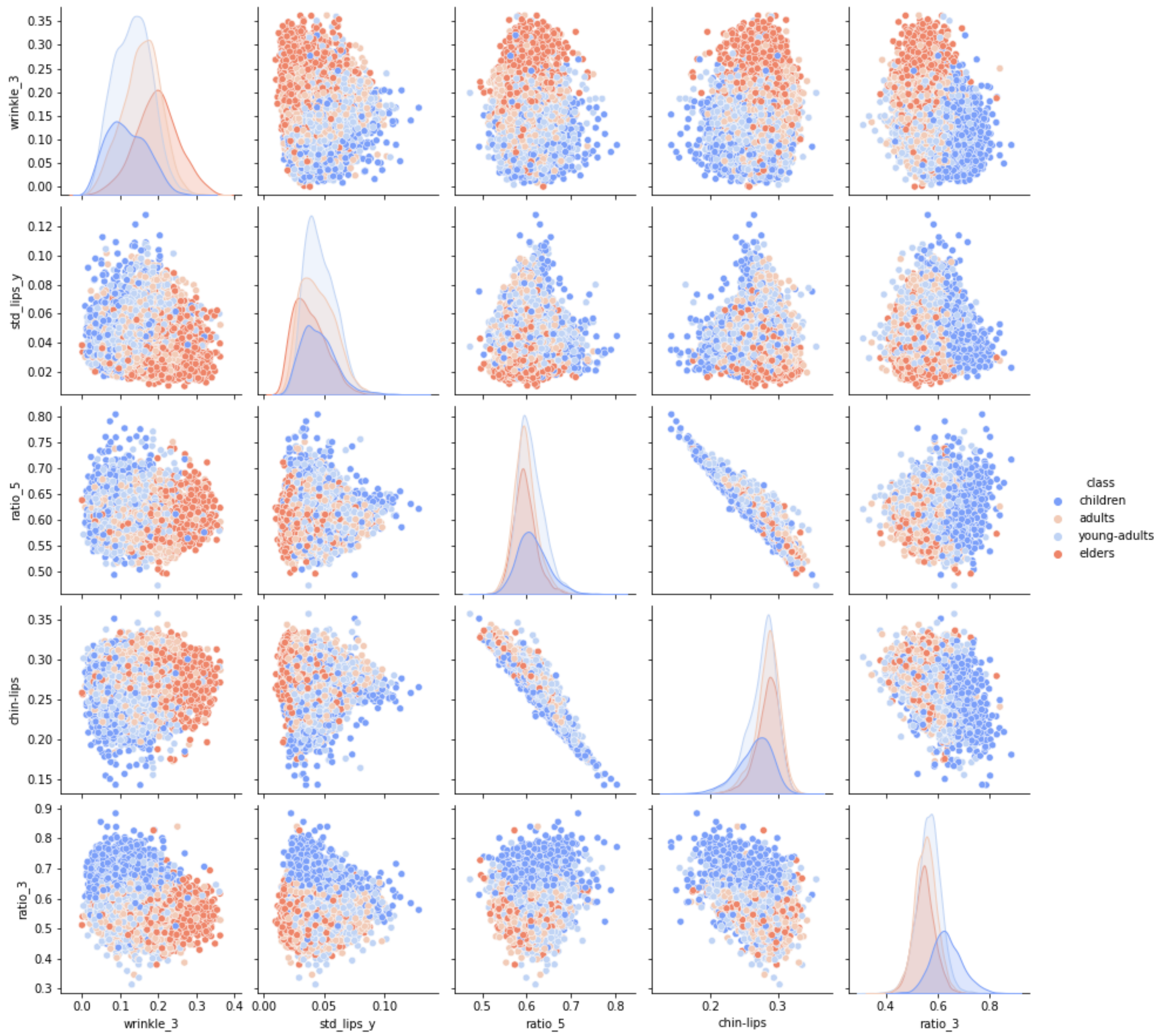


Figure 10: Pairplots

One last plot is the SHAP `force-plot`[4] for the image of the subject in example in 2. He was classified correctly in the last class and we can see that the features that push most toward this class are `wrinkle_3` and `std_lips_y`.

Figure 11: Force plot

References

[1] Nazre Batool and Rama Chellappa. "Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints". In: *Pattern Recognition* 48.3 (2015), pp. 642–658. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2014.08.003. URL: https://www.sciencedirect.com/science/article/pii/S0031320314002945.

[2] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[3] Adrian Rosebrock. *Facial landmarks with dlib, OpenCV, and Python*. 2017. URL: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/.

[4] Dr.Dataman. *Explain Your Model with the SHAP Values*. 2019. URL: https://towardsdatascience.com/explain-your-model-with-the-shap-values-bc36aac4de3d.

[5] Samuele Mazzanti. *Shap Values Explained Exactly How You Wished Someone Explained to You*. 2020. URL: https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30.