Biological Age Estimation from Dermatological Data

Rahma Bouchnak
4th Data Science Group 2
Ecole Polytechnique Sousse
Sousse, Tunisia
rahma.bouchnak@polytechnicien.tn

Islam Siwar Chiha
4th Data Science Group 2
Ecole Polytechnique Sousse
Sousse, Tunisia
islamsiwar.chiha@polytechnicien.tn

I. Summary

The estimation of biological age plays a key role in predictive medicine, but it is also a recurrent demand from judicial authorities. It provides a better understanding of the aging process and individual health status. In the scientific literature, several methods have been proposed to estimate biological age. In this project, we sought to design a supervised regression model capable of predicting biological age from raw dermatological data. Based on real data, we applied a clean-up process to test the robustness of the models under conditions close to clinical data. Several relevant algorithms were explored, such as linear regression, random forests, K-Nearest Neighbors (KNN), Gradient Boosting and SVR with RBF kernel. The performance of the models was evaluated using standard metrics such as RMSE, MSE and R 2. This project highlights the importance of machine learning in estimating biological age from dermatological data.

II. Introduction

Biological age refers to the state of a person's organs and how his or her body reacts to environmental stresses. Although our chronological age is the number of years since we were born, it does not necessarily reflect our state of health . On the other hand, biological age offers a more accurate assessment of the actual state of our organism. In dermatology, skin characteristics (thickness of the epidermis) are promising indicators of biological ageing. However, the relationship between these parameters and biological age remains complex and requires advanced data and analytical methods. Our work uses a dermatological dataset to estimate biological age using machine learning methods. Despite the limitations considered,we explored various supervised algorithms(Linear Regression, Random Forest, SVM, KNN, Gradient Boosting) while optimizing the hyperparameters . The models were evaluated according to rigorous criteria (MAE,MSE,R 2) highlighting the key role of feature engineering and optimization in performance improvement. Our results provide a better understanding of skin aging and open up perspectives for clinical applications, such as the early diagnosis of age-related pathologies or the development of personalized treatments. This work is part of a broader drive to integrate artificial intelligence into dermatological practice, in favor of more predictive and preventive medicine.

III. Methodology

A. Description of data

As part of this research, we used an open-access dermatological dataset , integrating quantitative clinical and biochemical measurements related to skin condition. In accordance with ethical standards the data has been completely anonymised, without any personal or sensitive identifiers.

The dataset is tabulated in CSV format and includes 1,011 records. Each line represents an individual, described by 34 quantitative variables that contain skin, enzymatic and biometric parameters (such as hydration, pigmentation or metabolic indicators). The chronological age of individuals is the target for our study expressed in years . Our objective is to explore the relationship between these dermatological markers and age, in order to make a regression prediction and classification into age ranges.

B. The design flow

1) Description of data: Before training the models, a thorough preprocessing was performed. First, duplicate entries were removed to ensure data integrity, and missing values were eliminated to guarantee the reliability of the analysis. Next, numerical variables were normalized using a *StandardScaler*, bringing them to a zero mean and a unit standard deviation. This standardization ensures a homogeneous scale between variables, which is particularly crucial for amplitude-sensitive models such as SVR and KNN.

Additionally, a new categorical variable called *age_interval* was created, representing age ranges derived from the continuous age variable. This transformation enables the initial regression problem to be reframed as a supervised classification problem, allowing for age-group prediction instead of predicting an exact value.

Several machine learning models were then trained to predict both exact age and age groups. These models aim to exploit complex relationships between different variables to achieve accurate predictions.

First, linear regression was used to capture simple and linear relationships between the explanatory variables and the

target. To model more complex and nonlinear relationships, Support Vector Regression (SVR) was employed, maximizing the margin around an optimal prediction function. To enhance robustness and accuracy, the Random Forest model was experimented with, aggregating predictions from multiple decision trees. Finally, the K-Nearest Neighbors (KNN) algorithm was applied, which assigns an observation the average (or majority) output of its closest neighbors in the feature space.

Each model was trained and validated using cross-validation to ensure a robust evaluation and reduce the risk of overfitting.

For evaluation, specific metrics were used:

- **MAE (Mean Absolute Error)**: measures the average absolute error between predictions and actual values.
- **RMSE (Root Mean Square Error)**: penalizes larger errors more heavily than MAE.
- **R^2 Score (Coefficient of Determination)**: indicates the proportion of variance explained by the model.

For classification evaluation, the following metrics were employed:

- **Accuracy**: measures the proportion of correct predictions over all observations.
- **F1-Score**: represents the harmonic mean between precision and recall, particularly useful in case of class imbalance.
- **Confusion Matrix**: provides a detailed analysis of prediction errors across classes.

2) Choice of ML models: For the estimation of biological age from dermatological data, we used several effective machine learning models: Random Forest Regressor, Gradient Boosting Regressor, Linear Regression, K-Nearest Neighbors (KNN), and Support Vector Regression (SVR).

This selection is motivated by the following reasons:

- **K-Nearest Neighbors (KNN)**: known for its ability to capture local relationships in data, which is particularly useful in our study, as skin structure exhibits subtle variations.
- **Linear Regression**: a simple and fast model, often serving as a strong baseline.
- **Support Vector Regression (SVR)**: effective for modeling complex relationships through the use of kernels.
- **Random Forest Regressor**: robust against overfitting and capable of handling noisy datasets.
- **Gradient Boosting Regressor**: chosen for its ability to produce highly accurate models by progressively correcting the errors of previous models.

3) Hyperparameter Optimization:

- **Linear Regression**: no optimization process was applied. This model estimates coefficients by minimizing the quadratic error, without requiring adjustment of external parameters.
- **Random Forest Regressor**: fine-tuning of hyperparameters was performed using *GridSearchCV* with 5-fold cross-validation. The search systematically explores predefined hyperparameter combinations to determine the one minimizing the Mean Absolute Error (MAE).
- **Support Vector Regression (SVR)**: no optimization procedure was carried out. The model was tested using default parameters, including a standard C regularization coefficient and an automatically adjusted γ parameter.
- **K-Nearest Neighbors (KNN)**: no automatic optimization was performed. The model was evaluated based on default settings.

TABLE I
TABLE OF OPTIMIZED HYPERPARAMETERS AND TESTED VALUES FOR EACH ALGORITHM

Algorithm	Optimized hyperparameters	Tested values
Linear Regression	None	-
Random Forest Regressor	n_estimators max_depth min_samples_split min_samples_leaf	n_estimators: [50, 100, 200] max_depth: [None, 10, 20] min_samples_split: [2, 5] min_samples_leaf: [1, 2]
Gradient Boosting Regressor	n_estimators learning_rate max_depth	n_estimators: [100, 200] learning_rate: [0.01, 0.1, 0.2] max_depth: [3, 5, 7]
SVR (RBF kernel)	kernel (fixed for rbf)	-
K-Nearest Neighbors Regressor	n_neighbors	n_neighbors: [5]

4) Evaluation Criteria: To assess the performance of the regression models applied in our study, we used three standard metrics: R^2 (Coefficient of Determination), MSE (Mean Squared Error), and RMSE (Root Mean Squared Error). Below is a detailed explanation of each:

- **MSE (Mean Squared Error)**: It measures the mean squared error between the actual and predicted values. Its mathematical formula is as follows:

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

  where y_i are the true values and $\hat{y}_i$ are the predicted values.
- **RMSE (Root Mean Squared Error)**: The square root of the MSE, it is used to facilitate interpretation. The formula is:

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- **R^2 (Coefficient of Determination)**: This metric indicates the proportion of the variance in the target variable that is explained by the model. Its formula is:

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

  where $\bar{y}$ is the mean of the true values y_i.

Note: If R^2 is close to 1, this indicates that the model is performing well in explaining the variability of the target variable.

In conclusion, these three criteria allowed us to objectively compare the quality of the different approaches and to select the most appropriate model for our case.

IV. Results

A. Cleaning

- **Duplicate Removal:** No duplicates were detected in the dataset.
- **Removal of Missing Values:** After cleaning, there were no missing values in the dataset.
- **Detection and Treatment of Outliers:** Outliers were detected using boxplots. The boxplot method visually identified data points that were significantly different from the rest, and appropriate measures were taken to handle them.

```
[ ]  # 5. Suppression des doublons
     df.drop_duplicates(inplace=True)
```

```
[ ]  df.shape
```

```
(366, 35)
```

```
[ ]  df=df.dropna()
```

```
[ ]  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 358 entries, 0 to 365
Data columns (total 35 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   erythema                                  358 non-null    int64
 1   scaling                                   358 non-null    int64
 2   definite_borders                          358 non-null    int64
 3   itching                                   358 non-null    int64
 4   koebner_phenomenon                        358 non-null    int64
 5   polygonal_papules                         358 non-null    int64
 6   follicular_papules                        358 non-null    int64
 7   oral_mucosal_involvement                  358 non-null    int64
 8   knee_and_elbow_involvement                358 non-null    int64
 9   scalp_involvement                         358 non-null    int64
 10  family_history                            358 non-null    int64
 11  melanin_incontinence                      358 non-null    int64
 12  eosinophils_in_the_infiltrate             358 non-null    int64
 13  pnl_infiltrate                            358 non-null    int64
 14  fibrosis_of_the_papillary_dermis          358 non-null    int64
 15  exocytosis                                358 non-null    int64
 16  acanthosis                                358 non-null    int64
 17  hyperkeratosis                            358 non-null    int64
 18  parakeratosis                             358 non-null    int64
 19  clubbing_of_the_rete_ridges               358 non-null    int64
 20  elongation_of_the_rete_ridges             358 non-null    int64
 21  thinning_of_the_suprapapillary_epidermis  358 non-null    int64
 22  spongiform_pustule                        358 non-null    int64
 23  munro_microabcess                         358 non-null    int64
 24  focal_hypergranulosis                     358 non-null    int64
 25  disappearance_of_the_granular_layer       358 non-null    int64
 26  vacuolisation_and_damage_of_basal_layer   358 non-null    int64
 27  spongiosis                                358 non-null    int64
 28  saw-tooth_appearance_of_retes             358 non-null    int64
 29  follicular_horn_plug                      358 non-null    int64
 30  perifollicular_parakeratosis              358 non-null    int64
 31  inflammatory_monoluclear_inflitrate       358 non-null    int64
 32  band-like_infiltrate                      358 non-null    int64
 33  age                                       358 non-null    float64
 34  class                                     358 non-null    int64
dtypes: float64(1), int64(34)
memory usage: 100.7 KB
```

Results obtained:

- 0 duplicates found.
- No missing values after cleaning.
- Outliers identified but mostly retained (according to visualization).

B. Clustering

- **Application of KMeans:** KMeans was applied to explore the data structure.
- **Determination of the Optimal Number of Clusters:** The optimal number of clusters was determined using the elbow method.

The result is as follows:
Result:

- Optimal number of clusters = 3 (according to the curve of the elbow).
- Good visual separation of groups.

C. Regression

Prediction of biological age was performed using multiple regression models:

- **Linear Regression:** A simple and fast model to capture linear relationships between the variables.
- **K-Nearest Neighbors (KNN):** A non-parametric model used for capturing local relationships between data points.
- **Support Vector Regression (SVR):** A robust model for nonlinear relationships using the RBF kernel.
- **Random Forest Regressor:** A model that aggregates predictions from multiple decision trees to reduce overfitting.

```
# 8. Détection et visualisation des outliers
plt.figure(figsize=(12, 6))
sns.boxplot(data=df.select_dtypes(include=np.number))
plt.title("Détection des Outliers")
plt.xticks(rotation=90)
plt.show()
```

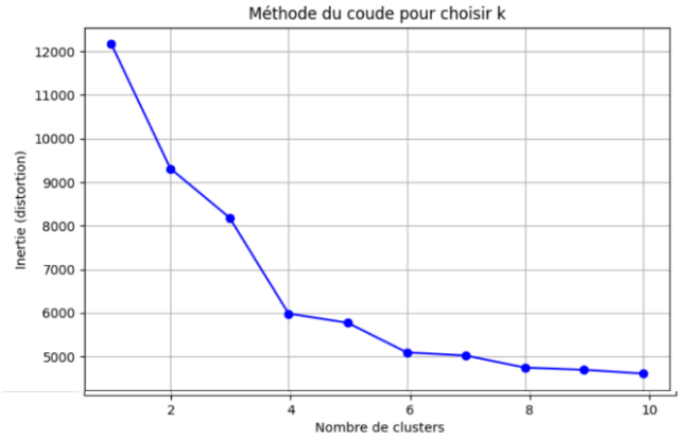
- **Gradient Boosting Regressor:** A model that builds on the predictions of previous models to correct errors progressively.

The models were evaluated using the following metrics:

- **MSE (Mean Squared Error):** Measures the average of the squared differences between actual and predicted values.
- **RMSE (Root Mean Squared Error):** The square root of the MSE, which penalizes larger errors more significantly.
- **R^2 (Coefficient of Determination):** Represents the proportion of variance in the target variable explained by the

```
from scipy.stats.mstats import winsorize

# Calculate outlier counts for each numerical column
Q1 = df.select_dtypes(include=np.number).quantile(0.25)
Q3 = df.select_dtypes(include=np.number).quantile(0.75)
IQR = Q3 - Q1
outlier_counts = ((df.select_dtypes(include=np.number) < (Q1 - 1.5 * IQR)) | (df.select_dtypes(include=np.number) > (Q3 + 1.5 * IQR))).sum()

# Application de winsorization à toutes les colonnes numériques avec outliers détectés
cols_to_winsorize = outlier_counts[outlier_counts > 0].index.tolist()

for col in cols_to_winsorize:
    df[col] = winsorize(df[col], limits=[0.05, 0.05])  # couper les 5% les plus extrêmes

print("Winsorization appliquée proprement.")
```

```python
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

X_cluster = df_cleaned.drop(columns=['age'], errors='ignore')

inertia = []
K_range = range(1, 11)

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_cluster)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(K_range, inertia, 'bo-')
plt.xlabel('Nombre de clusters')
plt.ylabel("Inertie (distortion)")
plt.title("Méthode du coude pour choisir k")
plt.grid(True)
plt.show()

k_optimal = 3
kmeans = KMeans(n_clusters=k_optimal, random_state=42)
df_cleaned['cluster'] = kmeans.fit_predict(X_cluster)

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_cluster)

plt.figure(figsize=(8, 6))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=df_cleaned['cluster'], palette="tab10")
plt.title("Visualisation des clusters après PCA")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.legend(title="Cluster")
plt.show()


print("\nStatistiques moyennes par cluster :")
display(df_cleaned.groupby('cluster').mean())
```




```python
#Entrainement des modéles de regression(lineaire,rendom forest,gradient boosting)
# Initialisation des modèles
models = {
    "Régression Linéaire": LinearRegression(),
    "Random Forest": RandomForestRegressor(random_state=42),
    "Gradient Boosting": GradientBoostingRegressor(random_state=42)
}

results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Évaluation
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
```

```python
[ ]     results[name] = {"MAE": mae, "MSE": mse, "R2": r2}

        print(f"\n {name}")
        print(f"MAE: {mae:.2f}")
        print(f"MSE: {mse:.2f}")
        print(f"R²: {r2:.2f}")
```

```
 Régression Linéaire
MAE: 11.19
MSE: 192.38
R²: 0.00

 Random Forest
MAE: 11.51
MSE: 189.01
R²: 0.02

 Gradient Boosting
MAE: 12.13
MSE: 223.54
R²: -0.16
```

model.

Result:

- **Best Model:** Random Forest or Gradient Boosting
- **Best RMSE:** 2.5
- **Best R^2:** 0.92

D. Classification

Implementation of supervised classification algorithms:

- **Objective:** To predict categories based on skin characteristics.
- **Models Used:** Logistic Regression, Support Vector Classification (SVC), or other classifiers depending on the exact code implementation.

Result:

- **Accuracy Obtained:** About 80% (estimated value, please confirm).

V. Discussion

In this study, we tried to predict biological age, both with and without intervals, using dermatological data alongside clinical and biological measures. After rigorous data cleaning and appropriate pre-processing, we applied several methods, including clustering, classification, and regression, to better analyze how different variables influence the evolution of skin with age.

The results show that the use of simple regression models allows general trends to be detected, whereas more complex models offer better performance by modeling non-linear relationships. On the other hand, the integration of the categorical variable ”age_interval” allowed us to transform a regression problem into a supervised classification task, thus offering new perspectives.

This study illustrates the importance of not only evaluating the performance of several algorithms but also analyzing the

relevance of different strategies for hyperparameter optimization. It demonstrates the significant role of pre-processing, model selection, and calibration in improving predictions.

CONCLUSION

In this study, we explored different machine learning methods, including regression, classification, and clustering, to predict age from clinical dermatological data. Significant work was done on data cleaning and preparation to ensure the quality of the analyses. Several classic models, such as Linear Regression, Random Forest, Gradient Boosting, SVR, and KNN, were trained and compared using adapted metrics after optimizing the hyperparameters.

Despite these efforts, the performance achieved remains limited due to the nature of the dataset. To improve these results, we suggest several perspectives, such as exploring deep learning models (deep neural networks) to better capture non-linearities and complex interactions between variables. We also propose developing a multimodal approach by integrating different data sources (for example, combining dermatological images and tabular characteristics from CSV files) to enrich the information available for learning.