Exploring dimensionality reduction and machine learning methods for the prediction of body composition abnormalities among an HIV+ population

Carolina Pelegrín Cuartero
Master in Bioinformatics and Biostatistics
Area 2 – Data Analysis

Nuria Pérez Álvarez
Carles Ventura Royo

05.06.2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Exploring dimensionality reduction and machine learning methods for the prediction of body composition abnormalities among an HIV+ population*
Nombre del autor:	*Carolina Pelegrín Cuartero*
Nombre del consultor/a:	*Nuria Pérez Álvarez*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	*Máster en Bioinformática y Bioestadística UOC-UB*
Área del Trabajo Final:	*Área 2 – Subárea 2 – Análisis de Datos*
Idioma del trabajo:	***Inglés/ English***
Palabras clave	*Dimensionality reduction, machine learning, body composition abnormalities*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

El objetivo de este trabajo fin de máster ha sido el de predecir tres tipos de anomalías corporales relacionadas con la calidad de hueso (osteoporosis/osteopenia), la redistribución de grasa (lipodistrofia) y una baja masa muscular, para un conjunto de pacientes con VIH. Dichas anomalías son efecto de la terapia antiretroviral y la inflamación crónica del sistema inmune causada por el propio virus.

Para la realización de este estudio, se dispuso de un conjunto de medidas corporales procedentes de un análisis DEXA; tres de ellas se usaron para establecer la presencia de cada enfermedad en base a valores de corte extraídos de la bibliografía.

Varios tipos de modelos de predicción se construyeron usando distintos sets de variables, incluyendo variables originales y variables sintéticas creadas por análisis de componentes principales, clustering de variables y análisis factorial múltiple. Para la predicción de cada enfermedad, solo se usaron aquellas variables no-directamente relacionadas con ella. Se ajustaron modelos de regresión logística y de machine learning, incluyendo “ensembles” o conjuntos de modelos; los mejores modelos se seleccionaron en base a su ajuste y el valor de AUC obtenido. El uso de “ensembles” mejoró sensiblemente la predicción de lipodistrofia y baja masa muscular, con un funcionamiento *excelente* según la escala de valores de AUC. La predicción de osteoporosis/osteopenia mostró resultados aceptables pero mucho peores que

para las otras dos anomalías, probablemente debido a que variables importantes en la definición de la calidad de hueso no estaban disponibles para la realización de este estudio.

Abstract (in English, 250 words or less):

The main aim of this study was to classify a set of patients with HIV as having different type of body abnormalities (i.e. osteoporosis/osteopenia, lipodystrophy, low muscle mass), caused by the antiretroviral therapy and the chronic inflammation of the immune system caused by the virus itself. Building classifiers may lead to earlier diagnose, decreasing health effects and improving life quality and expectancy of HIV+ patients.

For this study, a set of measurements from a DEXA analysis was available; three of them were used to establish presence of disease, based on cut-offs found at the bibliography.

Models were built with original ("raw") variables and synthetic variables created by principal component analysis, multiple factor analysis and clustering of variables. For the prediction of each disease, just not-directly-related features were taken into account. Different type of classification methods were used, including logistic regression, machine learning and ensemble learning methods. Models were fitted using training datasets and validated using test datasets; "best" models were selected based upon their accuracy and AUC value. Ensemble models greatly improved prediction of lipodystrophy and low muscle mass, with models showing an excellent performance, demonstrating its capacity to extract subtle patterns from the data. Performance of models for the prediction of bone-related disease was just acceptable, probably due to the class-imbalance present and the lack of important variables related to the bone quality.

Contents

List of Figures

1 Introduction

1.1 Introduction to the topic

The human immunodeficiency virus (HIV) is nowadays considered as a chronic disease, due to the advances made in antiretroviral therapy (ART). However, it has been shown that both ART and the chronic inflammation and activation of the immune system caused by the virus itself, have short and long-term impacts in the body morphology (Nasi et al. 2017) causing chronic age-associated diseases such as loss of bone mass (osteoporosis, osteopenia) (Powderly 2012; Lima et al. 2011), muscle mass (Neto et al. 2016) and fat redistribution, known as lipodystrophy or "HIV-associated lipodystrophy syndrome" (Freitas et al. 2010). These body composition abnormalities are known to be present in a large percentage of the population living with HIV; they are closely related with other health and economic problems such as fractures, metabolic diseases (i.e. diabetes), cardiovascular diseases, higher healthcare costs and eventually, higher mortality rates. They also cause significant cosmetic changes that can result in a visible manifestation of the HIV virus, causing stigmatization of the infected individuals that might lead to an early abandon of the treatment (Montessori et al. 2004), with the high risk that this implies for the patients' health.

The "gold standard" technique to measure body composition is Dual-energy X-ray absorptiometry (McClung 2003), known as DEXA or DXA. This method is capable of measuring lean, fat and mineral bone composition in several body compartments, such as the lumbar spine or the hip (McClung 2003; Kendler et al. 2013). Three of the measurements provided by DXA are currently used to identify the presence/absence of morphologic diseases:

- $T-scores$ are used to idenfity low-bone associated diseases (McClung 2003),
- fat mass ratio (FMR) is used to define presence/absence of lipodystrophy (Freitas et al. 2010), and
- appendicular skeletal muscle mass values ($ASM/height^2$) are used to identify low muscle mass (Cruz-Jentoft et al. 2018).

Based on DXA measurements, classifiers for the prediction of body composition abnormalities can be built, leading to earlier diagnoses. That may avoid more serious health effects and therefore, increase life quality and expectancy of HIV+ patients, as well as prevent them from abandoning the treatment. Traditionally, regression models have been used to diagnose medical diseases. However, over the last decades, newer machine learning models have been developed, leading to an improvement in the diagnose accuracy. The "best" method to use depends usually on the task (Kiang 2003); therefore, the usual approach consists of using different classification algorithms and compare their performance. In order to improve this performance, ensemble learning methods (combination of classifiers) can be used (Kilic and Hosgormez 2016); these better classification entails nonetheless a worse comprehensibility of the model.

1.2 Objectives and motivation

Knowing that population with HIV are likely to suffer from chronic morphologic diseases such as osteoporosis and osteopenia, lipodystrophy and loss of muscle mass, and taking into account that they remain asymptomatic at early stages (McClung 2003), the main aim of this Master's Thesis will be to **build classifiers in order to predict the presence/absence of those three body composition abnormalities within an HIV+ population, of male and female patients from all ages**. Different kind of classifiers will be explored and built, including traditional logistic regression models and machine learning (ML) ones. Combinations of various classifiers, known as ensemble learning methods, will be studied as well, these methods being known to be especially suitable for high dimensionality datasets, like medical ones (Tay et al. 2013). Since there is no "best" model that applies to every case, our approach will be to apply several regression and ML models to different feature sets and compare their performance, finally selecting, at least, one model with high accuracy (regardless of its interpretability), and one "easier-to-interpret" model.

Extracting different feature sets, representative of each disease, is one of the most important steps in building accurate classifiers, especially in datasets with a big amount of variables, which may also be highly correlated to each other.

Therefore, our second objective will be to **create appropriate feature sets for the prediction of each disease, using original ("raw") variables and synthetic variables created by dimensionality reduction techniques**.

Following those two main objectives, the main research questions to answer will be:

1. Is there enough information available to accurately predict the presence/absence of each of the three diseases under study?
2. Does use of dimensionality reduction methods improve the performance of the models?
3. Does use of more complicated models (i.e. machine learning, ensemble learners) improve predictions?

1.3 State of the art and research gap

Previous studies have shown a relationship between anthropometric and DXA measurements and the presence of body composition abnormalities. Widely studied seems to be the relationship between bone quality and body mass index (BMI) (Schtscherbyna et al. 2012; Pinnetti et al. 2014; Bolland et al. 2007), total body fat (Schtscherbyna et al. 2012), (older) age, lower weight and increasing height (Pinnetti et al. 2014; Carr et al. 2015; Yoo et al. 2013), for both "normal" and HIV+ populations. Therefore, it seems like *very basic anthropometric measurements may have high predictive capacity*. Other kind of data, such as nutritional status (Schtscherbyna et al. 2012), ethnicity (Carr et al. 2015), diabetes, as well as female-specific features (i.e. duration of menopause, duration of breast feeding or estrogen therapy) (Yoo et al. 2013), have also been found to have predictive capacity, but were not available for this study.

Logistic regression is still a widely used method in the prediction of morphologic diseases. In particular, presence of osteoporosis seems to be widely studied in different kind of populations, including postmenopausal women (Yoo et al. 2013) and HIV+ populations. Results of logistic regression models are often compared with ML models such as support vector machines, random forests or artificial neuronal networks (see (Yoo et al. 2013; Ioannidis et al. 2003) for some examples), with ML models often leading to a better performance. Ensemble learning techniques have also been lately applied to the classification of patients as having osteoporosis, osteopenia and normal bone quality (Kilic and Hosgormez 2016).

Most of the studies found on the topic usually focus on the prediction of one type of body composition abnormality, generally related to low bone mass or fat-redistribution; no papers have been found for the prediction of low muscle mass. Logistic and artificial neuronal network models seem to be the most widely used ones, with few papers exploring other kind of methods. In this thesis, a wider view on the prediction of body composition abnormalities associated with the human immunodeficiency virus is proposed. Use of logistic regression models for classification and some machine learning techniques will be applied to different datasets, including original variables - extracted from anthropometric and DXA measurements - and "synthetic" ones, obtained by dimensionality reduction techniques such as principal component analysis, clustering of variables or multiple factor analysis.

1.4 Planning, timing and tasks

An overview of the planning initially designed can be observed at Figure 1, created with the GattProject free software. Arrows indicate *dependence* between tasks (i.e. tasks that need to be accomplished before starting other tasks), red dots indicate the *milestones* related with the Master's Thesis, while bars indicate tasks, with colour indicating the *type of task* to be done:

- *bibliography search* (dark green);
- *statistical analysis/model building* (yellow);
- *first trials and methods' search* (blue);
- *'spare' time* or time left to finish incomplete tasks, select best models, get final conclusions, etc. (olive green);
- *writing assignments*, such as the final report and presentation (grey);
- *analysis* and *final writing* (black), as the two main tasks in which this study was initially split.

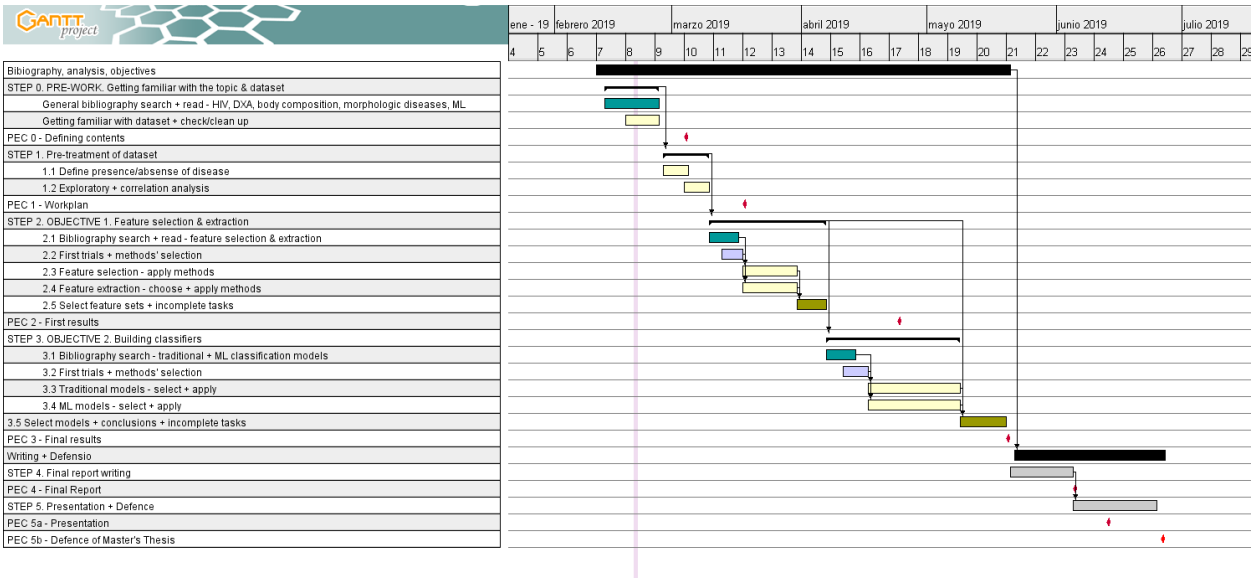


Figure 1: Original planning and timing.

After the official start of the Master's Thesis (see vertical purple bar on Figure 1), five main steps were followed:

1. **Getting familiar with the dataset**. This step included checking the available dataset and a bibliography search. Doing a bibliography search was a really important step, since it gave an overview of the topic: what has been done, in what kind of patients, what has not been done yet, etc.

2. **Pretreatment of the dataset**, including an exploratory analysis and the setting of presence/absence of disease based on features' cut-offs found at the bibliography.

3. **Dimensionality reduction and features' selection techniques**, in order to decrease the dimensionality and the multicollinearity of the dataset.

4. **Building classifiers** such as logistic regression, machine learning and ensemble learners, selecting the "best" models based on their performance, and drawing some final conclusions.

Construction of logistic models (step 4) and use of automated features' selection techniques (step 3) overalapped, and so did the timing reserved to those tasks.

Also, wrapping the code into functions was not included in the original plan. However, after finishing the code, we decided to invest some time in wrapping it into functions, which improved its quality and usability. Programming those functions did therefore delay the planned tasks; delay was compensated with a progress on the final writing.

1.5 Description of contents

This Master's Thesis has been structerd into:

1. An **introduction** on the topic, including *state of the art*, main objectives and original planning and timing, with comments on the changes done on that planning.

2. A chapter containing an **overview of the available dataset** and the **methodology** followed in order to get the results. This chapter is divided into:

    - Software used.
    - Pre-treatment and exploratory analysis' methods used to check the dataset, split patients into more homogeneous groups, correct class-imbalance and create training and test datasets used to train/fit and validate model's performance, respectively.
    - Dimensionality reduction methods used to create "synthetic" variables, that will be used as "normal" variables in the fitting of models. Methods explored include principal component analysis, clustering of variables and multiple factor analysis.
    - Machine learning and ensemble methods used.
    - Performance and model's selection criteria.

3. A chapter containing an **overview of the main results** obtained by applying the suggested methodology, and a discussion on those results.

4. A chapter containing the **main conclusions** and **future work**.

5. A chapter containing the acronyms used and a glosary clarifying some of the terms often used.

Also, two appendices can be found at the end of the report, including figures (appendix I) and the R code used (appendix II).

2 Methods

The methods and analysis presented in this chapter may vary between diseases, as lipodystrophy and low muscle mass have binary outputs (presence/absence of disease), while bone quality is of multi-class type (normal/osteopenia/osteoporosis); differences will be stated when needed.

2.1 Software

In order to carry out the analysis that will be presented at this chapter, *R* will be used. R is a programming language and environment widely used for statistical computing, since it integrates a collection of tools for data analysis, wrapped in the so-called "packages". Version R-3.6.0 of and 1.1.456 of *RStudio* will be used. Code, theory and analysis' results will be integrated into the final report using *R Markdown*.

In R, there are usually several functions to carry out one specific analysis. Chosen functions, and the packages they belong to, will be mentioned at the end of each subchapter. All the code - including customized functions programmed to extract modelling results - can be reviewed at the Appendix II.

2.2 Pre-treatment of dataset and exploratory analysis

2.2.1 Original dataset. Available data

The original dataset contained data of 1480 patients with HIV (rows) for 82 features (columns) columns, including age, DXA measurements (of bone, fat, muscle) and anthropometric measurements, such as weight, height or BMI.

After checking for missing data and entry errors, 23 patients were eliminated from the dataset: 22 containing at least one missing value, and one patient with entry errors.

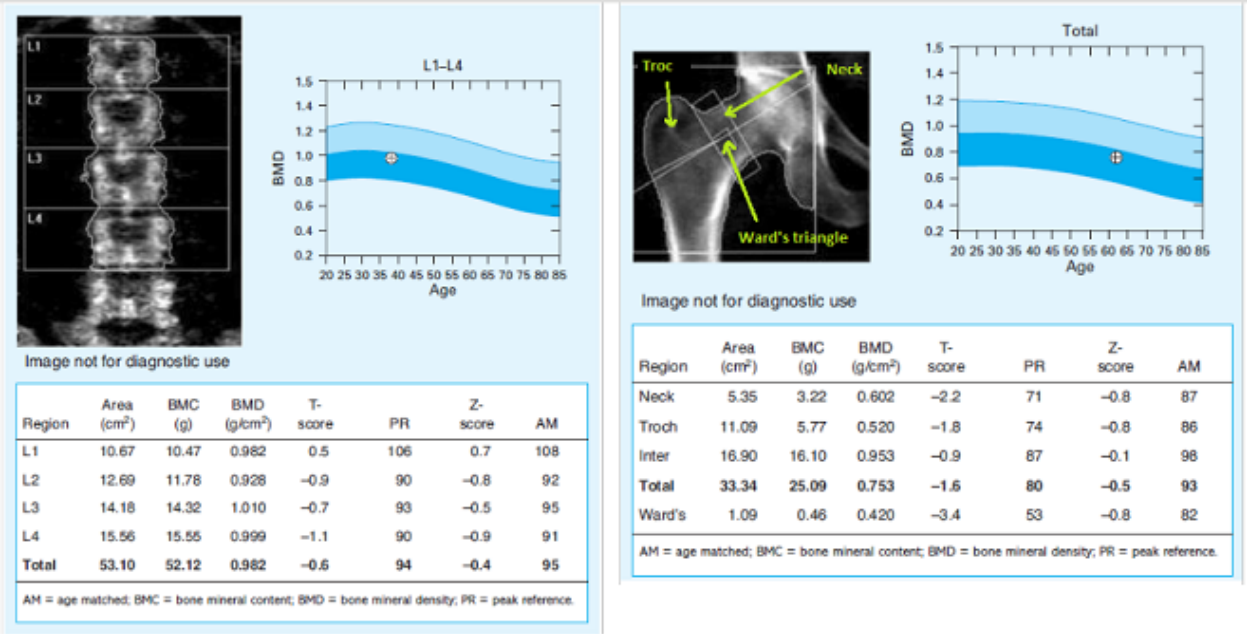
Columns that were empty or redundant were excluded; 63 features were finally included in the dataset. Since they did not seem to follow any specific order, features were relocated so that related ones were found to be next to each other, simplifying further analysis. A review of the variables finally taken into account, with their coded names and meanings can be found at Figure 2.

	FEATURES	MEANING
General + anthropometric measurements	*Gender*	
	Age	
	Height	
	Weight	
	BMI	Body mass index (kg/m2)
Bone-related features	*L1BMD, L2BMD, L3BMD, L4BMD*	Bone mineral density measurements at spine sites
	L1L4BMD, L2L4BMD	BMD spine sections
	L1T, L2T, L3T, L4T	T-values at spine sites
	L1L4T, L2L4T	T-values at spine sections
	L1Z, L2Z, L3Z, L4Z	Z-values at spine sites
	L1L4Z, L2L4Z	Z-values at spine sites
	NeckFBMD, WardsBMD, TrochBMD, TotalFBMD	BMD values at hip sites
	NeckFT, WardsT, TrochT, TotalFT	T-values at hip sites
	NeckFZ, WardsZ, TrochZ, TotalFZ	Z-values at hip sites
	minT_gral, minT_hip	Minimum T-values extracted from all bone sites/hip
Fat-related features	*RAFg, LAFg, BothAFg*	Arms (right, left, both)
	LLFg, RLFg, BothLFg	Legs (right, left, both)
	Tfg	Trunk fat mass
	TotalFg	Total fat mass
	FMI	Fat mass Index (kg/m2)
	FMR	Fat mass ratio (% at trunk / % at legs)
	Indexdistributionfat	Fat % at trunk / % at limbs
	FtrunkgFLegsg	Fat at trunk (g) / fat at legs (g)
	FtrunkgFtotalg	Fat at trunk (g) / total fat (g)
	FlegsgFtotalg	Fat at legs (g) / total fat (g)
	FlimbsgFtotalg	Fat at limbs (g) / total fat (g)
	LlegFgBMI	Fat at left leg (g) / BMI
Muscle-related features	*RALg, LALg, BothALg*	Arms (right, left, both)
	RLLg, LLLg, BothLLg	Legs (right, left, both)
	Tlg	Trunk muscle mass
	TotalLg	Total muscle mass
	FFMI	Free fat mass index (kg/m2)
	Apendicularleanmas	Appendicular lean mass index (muscle at limbs / hieght2)

Figure 2: Review of all available features included at the final dataset, used to fit prediction models.

- *Bone measurements* included bone mineral density (BMD), *T-values* and *Z-values*,[1] for different spine and hip sites; a representation of those sites can be found at Figure 3. Minimum T-value (for all sites and for hip sites) was calculated and included as well (`minT_gral, minT_hip`).

[1] *T-values* and *Z-values* are scores that show the quality of the bone mass, compared with the bone mass of an average "healthy" population; they are used to diagnose bone-quality related diseases, such as osteoporosis and osteopenia.



Region	Area (cm^2)	BMC (g)	BMD (g/cm^2)	T-score	PR	Z-score	AM
L1	10.67	10.47	0.982	0.5	106	0.7	108
L2	12.69	11.78	0.928	−0.9	90	−0.8	92
L3	14.18	14.32	1.010	−0.7	93	−0.5	95
L4	15.56	15.55	0.999	−1.1	90	−0.9	91
Total	**53.10**	**52.12**	**0.982**	**−0.6**	**94**	**−0.4**	**95**

AM = age matched; BMC = bone mineral content; BMD = bone mineral density; PR = peak reference.

Region	Area (cm^2)	BMC (g)	BMD (g/cm^2)	T-score	PR	Z-score	AM
Neck	5.35	3.22	0.602	−2.2	71	−0.8	87
Troch	11.09	5.77	0.520	−1.8	74	−0.8	86
Inter	16.90	16.10	0.953	−0.9	87	−0.1	98
Total	**33.34**	**25.09**	**0.753**	**−1.6**	**80**	**−0.5**	**93**
Ward's	1.09	0.46	0.420	−3.4	53	−0.8	82

AM = age matched; BMC = bone mineral content; BMD = bone mineral density; PR = peak reference.

Figure 3: Examples of DEXA images and data for spine (left) and hip (right) measurements, extracted from Hain (2006).

- *Fat and muscle features* included measurements at arms, legs and trunk, as well as measurements of *total fat* and *total muscle*; some ratios were included as well.

It has to be taken into account that **not all available features can be used to predict every disease, but just those that are not directly related to them**. For instance, prediction of lipodystrophy will just use anthropometric, bone-related and muscle-related features, but none of the fat-related ones. Therefore, **a different set of features will be used for the prediction of each disease**, including a different number of features: 31 for the prediction of osteoporosis/osteopenia, 48 for the prediction of lipodystrophy and 54 for the prediction of low muscle mass.

2.2.2 Presence of disease

Presence of disease will be established by different cut-offs extracted from the bibliography (McClung 2003; Freitas et al. 2010; Cruz-Jentoft et al. 2018) (see Figure 4).

Bone disease		**Fat disease**		**Muscle disease**	
T-score	*Diagnosis*	*FMR*	*Diagnosis*	*ASM/height²*	*Diagnosis*
< +1.0 & > -1.0	Normal	≥ 1.961	Lipodystrophy (men)	< 7.0 Kg	Low muscle mass (men)
< -1.0 & > -2.5	Osteopenia	≥ 1.329	Lipodystrophy (women)	< 6.0 Kg	Low muscle mass (women
≤ - 2.5	Osteoporosis				

Figure 4: Reference variables and cut-offs used to define presence/absence of disease.

- Presence of **low-bone** associated disease, such as osteoporosis and osteopenia, will be defined using *T-scores*, following the World Health Organization recommendations (McClung 2003). T-score values are calculated at every measurement site (i.e. hip, spine) from BMD values, as the number of standard deviations from the "normal population" (i.e. young population with a healthy bone quality) (Powderly 2012). Minimum T-score of each patient will be extracted and considered to define the overall bone quality of the patient.

- Presence of **fat-redistribution** associated disease or lipodystrophy will be diagnosed attending to values of *fat mass ratio* (FMR) extracted from the DXA analysis; cut-offs from Freitas et al. (2010) will be used. Diagnose of HIV-related lipodystrophy by FMR was first proposed by Bonnet et al. (2005) and has been proved to be an objective diagnostic tool (Beraldo et al. 2015).

- Presence of **low muscle mass** will be defined by cut-offs for *Apendicular Skeletal Muscle Mass* (ASM) values, defined at the second European Group on Sarcopenia in Older People (EWGSOP2) (Cruz-Jentoft et al. 2018).

2.2.3 Exploratory analysis

Previous to any feature selection or models' fitting, a preliminary and exploratory analysis will be carried out on the original dataset. This analysis will include a correlation and normality study of main features, cluster analysis and split of the dataset into more homogeneous groups.

- **Multicollinearity**

As previously mentioned, most of the available *features* are different kind of body measurements from a DXA analysis; therefore, they *are expected to be redundant or multicollinear*. If that is the case, relative importance of predictors is difficult to assess and models are unstable, since small changes in the dataset can strongly affect them (Dormann et al. 2013).

Multicollinearity is not a problem *per se*. For instance, if the aim of a study is to predict an output (i.e. presence/absence of a certain disease), and models are applied to a similar

dataset as the one used to build the classifier, results will be able to be extrapolated, since new data will be expected to have the same collinearity as the data used to build the model.

Multicollinearity can be avoided by eliminating redundant variables or building models with "synthetic" variables crated by dimensionality reduction techniques, such as principal component anlysis or multiple factor analysis.

- **Correlation study**

In order to state the correlation between variables, Pearson's coefficient will be calculated and used to build a *correlogram*[2]. The *Pearson's coefficient* is a dimensionless measure of the relationship between two variables (therefore not affected by changes in the variables' units). It is calculated as:

$$r = \frac{S_{XY}}{S_X S_Y}$$

where S_x, S_y represent the sample standard deviations and S_{XY} represent the sample covariance.

Pearson's coefficient assumes that variables are normally distributed; therefore, normality will also be checked for some of the most representative variables, using boxplots.

Correlation between variables used to define presence of disease and those variables not directly related to them will be calculated as well. The ten most correlated variables for each case will be used to build logistic regression models, as a case of "manually selected variables", as it will be later explained.

2.2.4 Cluster analysis and split of dataset

Two different splits of the dataset into more homogeneous groups will be studied: *split by cluster analysis* and *split into men and women* patients. Building models for different sets of patients will let us take into account their particularities while improving performance of classifiers.

Cluster analysis is an unsupervised method of grouping observations into clusters, taking into account that each observation can just belong to one cluster. In order to determine the "best" number of clusters, the *Silhouette method* will be used. First introduced in 1987 by Peter Rousseeuw, silhouettes are values that represent the quality of the clustering. One silhouette is calculated for each possible partition; number of clusters that maximize the average silhouette value will be the optimal one (Rousseeuw (1987)).

Silhouette method for the partition of observations into clusters will be applied by the function `fviz_nbclust()` from the package `factoextra` (Kassambara and Mundt 2017), including the arguments `x` (scaled dataset of patients), `kmeans` (partition method to use), `method` (method to be used for estimating the optimal number of clusters) and `k.max` (maximum number of clusters to consider).

[2]graphic representation of the strength of the relationship between variables.

```
fviz_nbclust(x = dexa.clust, kmeans, method = "silhouette", k.max = 10)
```

Once the optimal number of clusters has been calculated and data has been scaled - to avoid features with bigger range of values having greater influence in the clusters- *K-means clustering method* will be used to actually build the clusters. The goal of this method is to create a specific amount of clusters, each one having the smallest within-cluster variation, which is defined by the Euclidean distance between observations. In other words, the K-means algorithm will try to minimize the value of $W(C_k)$:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

where C_k denotes the number of observations in the kth cluster, i represents the ith observation and $(x_{ij} - x_{i'j})^2$ is the Euclidean distance (between two observations).

Split within clusters will then be compared with a split within male and female patients.

2.2.5 Study of class-imbalance

Final datasets used for prediction will contain one output representing the presence/absence of disease (previously calculated) **and all the features not directly related to that disease**. Those three datasets will then be split into (two) more homogeneous groups, as stated at the previous step, and presence of disease within each group will then be investigated.

A dataset is considered to be "imbalanced" if the classes are not equally represented, as it is often the case in real-world data. Presence of class-imbalance may lead models to overfit, just predicting the majority class, not being able to correctly diagnose. Thus, it is important to check that no substantial class-imbalance exists, and correct it - if possible - in case it does.

Different approaches exist to deal with class-imbalance; in this case, method *SMOTE* (synthetic minority over-sampling technique) will be used (Chawla et al. 2002), which creates new observations of the minority class, in a process that follows the next main steps:

1. All data points are plotted and samples of interest (belonging to the minority class) are identified (see green points at Figure 5a). Those samples are called *feature vectors*.
2. K-nearest-neighbours to those feature vectors are then identified (see black and yellow dots at Figure 5b).
3. Distance between feature vectors is calculated and multiplied by a random number between 0 and 1.
4. A new datapoint is plotted on the line by adding the calculated number to the feature vector under consideration (see red dot at Figure 5c).
5. Process is repeated until enough new data points have been created.

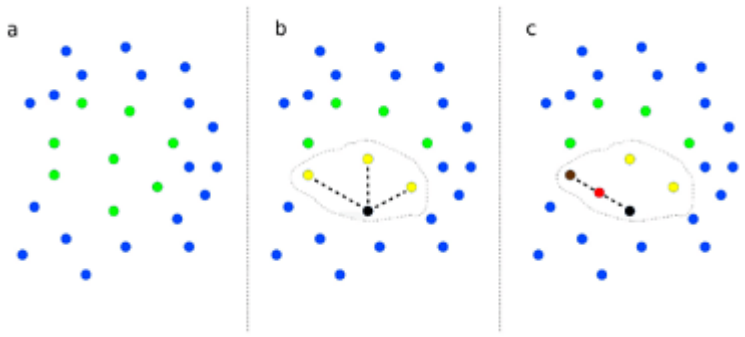


Figure 5: Example of SMOTE process. Figure has been extracted from Schubach et al. (2017).

Method SMOTE will be applied to each dataset using the function `SMOTE()` from package `DMwR` (L. Torgo and Torgo 2013).

2.3 Dimensionality reduction methods. Creating "synthetic" variables

As previously mentioned, a big amount of features (63) are available, and multicollinearity is expected to exist between them. Dimensionality reduction methods will be **used to both, reduce dimensionality of the dataset by replacing the p original variables by k "new" synthetic variables (i.e. principal components, clusters), while avoiding for multicollinearity**. Those created variables will then be used as predictors at the construction of classification models; those models will be "simple" in terms of number of variables included, but will be however very hard to interpret.

Use of three different methods will be explored, namely:

- Principal component analysis or PCA
- Clustering of variables
- Multiple factor analysis or MFA

Each method will be applied to each of the available datasets (i.e. for each sex and disease under study).

Being the main aim of the study the *prediction* body composition abnormalities, focus will be set on extracting synthetic variables to use at modelling. That means, some basics of the three mentioned methods will be explained, but just one example will be included at this report (i.e. using the set of features for the prediction of lipodystrophy at female patients). A deeper analysis of all 18 possible cases (one per each sex, disease and dimensionality reduction method used) is outside of the scope of this study.

2.3.1 Principal Component Analysis

PCA is a technique that reduces the dimensionality of a dataset containing $X_1, ..., X_p$ more or less correlated variables, by creating p linear combinations of those variables, known as *principal components* (PCs). PCA does not reduce the number of features of the dataset - they are all included at each of the PCs.

Among all possible linear combinations of the original set of features, the first PC will be the one explaining most of the variability; it is defined as:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p$$

where $\phi_{11}, ..., \phi_{p1}$ represent the *loadings* of the first PC. Each PC has a *loading vector* ϕ_i, which geometrically defines the direction in the features' space along which the data varies the most (i.e. the direction of the principal component); $X_1, ..., X_p$ are the original variables and Z_1 is the factor for the first PC (the linear combination itself).

The second PC will be the linear combination, orthogonal to the first one, that accounts for as much of the remaining variation as possible, and so on. In other words, most of the variability in a dataset containing correlated features will be explained by the first few PCs, being those not correlated to each other - and therefore, avoiding for multicollinearity.

Scores of each PC are calculated as the projections of the n observations onto the direction or loading of that PC, and will be used as "synthetic" variables while building prediction models. For instance, score for the first PC is calculated as:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip}$$

The "best" number of PCs to be used can be selected by a *scree test plot* (see example at Figure 16). As a rule of thumb, components that appear before the "elbow" at the scree plot will be selected as being representative of the whole features' set, since those are the ones able to explain most of the variability contained in the original dataset.

PCA will be applied by the function `prcomp()` from the package `stats` (R Core Team 2013), indicating that data needs to be normalized (`scale = T, center = T`), and the number of principal components to be calculated (`rank. = 15`):

```
pca.res <- prcomp(pca.set, scale = T, center = T, rank. = 15)
```

2.3.2 Clustering of variables

This method consists of *creating clusters containing strongly-correlated variables*. Each cluster will be a linear combination of the variables contained within it, and will score as a single numerical variable, which represents how the cluster is correlated to each of the variables within it. These cluster scores can be then used to build prediction models. For more information on the method see Chavent, Genuer, and Saracco (2016).

Since "best" number of clusters is a priori unknown, *hierarchical clustering* will be used. First, p clusters are formed (each one containing one of the p original features) and a dissimilarity measure, such as the Euclidean distance, is calculated between them. Then, most near features are joined, forming a new cluster (Von Luxburg and others (2010)). The robustness of the partitions against random fluctuations in the data will be studied for all possible partitions and for different sets of n datapoints, created by bootstrap.[3] *Instability* is then calculated as the mean distance between clusterings:

$$Instab(K, n) = E(d(C_k(S_n), C_k(S'_n)))$$

where K is the number of clusters, n the size of the dataset, d the distance between two clusters and S_n, S'_n represent two different sets of n datapoints to compare.

Parameter k that minimizes that distance will be finally chosen.

Clustering of variables will be applied by functions of the package `ClustOfVar` (Chavent et al. 2011), including `hclustvar()` for carrying out hierarchical clustering, `stability()` to decide on the adequate number of clusters to use and `cutreevar()` to create the selected number of clusters.

```
hclust.res <- hclustvar(X.quanti = pca.set) # "X.quanti" gets a quantitative
                                            # dataset as input
clust.res <- stability(hclust.res, B = 20)  # "B" = bootstrap samples
clust.cut <- cutreevar(hclust.res, k = 15)  # "k" = number of clusters
```

2.3.3 Multiple Factor Analysis

Multiple Factor Analysis or MFA is an extension of PCA, for datasets where features are structured into groups; in this case of study, variables will be considered to belong to five different groups:

- *Age*
- *Anthropometric measurements*, including BMI, weight and height
- *Bone-related measurements*, including all BMD measurements, T-scores and Z-scores from spine and hip.
- *Fat-related* measurements and ratios
- *Muscle-related* measurements and ratios

The aim of MFA is the same as that of PCA: to simplify and reduce the dimensionality of the dataset by extracting a new set of linear combinations of the original variables, that will be used as new, synthetic variables in the construction of prediction models. As previously

[3] resampling technique that consists on creating k random samples (with replacement), often used to train models on them.

mentioned, the difference between MFA and PCA is that MFA takes into account the structure of the variables (in groups). Therefore, it needs to *balance the influence of each group in the construction of the dimensions* (the synthetic variables).

Each variable of the group j will be weighted by:

$$\frac{1}{\lambda_1^j}$$

where λ_1^j is the first eigenvalue of the factor analysis applied to set J. These weights are identical for the variables of the same group, and vary from one group to another. In this way, a set with high dimensionality will contribute to various axes, but will not necessarily contribute more to the first one (Pages 2004).

Multiple Factor Analysis will be carried out using the function `MFA()` from the package `factoMineR` (Lê et al. 2008), where:

- `group`: indicate the grouping of the variables. In the example, variables are grouped in `Age` (first variable), `anthropometric measurements` (next three variables at the dataset), `bone measurements` (next 33 variables) and `muscle measurements` (last 10 variables at the dataset). Each group has to include just one type of variable (i.e. continuous/categorical)
- `type`: indicates the type of variables included in each group ("s" indicates "scaled", "c" indicates "continuos").
- `ncp`: indicates the number of dimensions to be extracted.
- `name.group`: let us name the groups of variables used.

```
res.mfa <- MFA(pca.set, group=c(1,3,33,10), type=c(rep("s",4)), ncp=15,
               name.group=c("age","antrop","bone","muscle"))
```

As for PCA, "best" number of MFA components will be selected by studying its scree plot, selecting the number of components over the "elbow".

2.4 Logistic regression models and features' selection techniques

2.4.1 Logistic regression models

Logistic regression models the probability that an output variable belongs to a particular category or class; predicting a qualitative response is therefore referred as "classification". Logistic regression also allos for the prediction of more than two classes, such as for "normal/osteopenia/osteoporosis".

Linear regression is the equivalent to logistic regression, for quantitative responses. However, while linear regression models the outcome directly, logistic regression models the probability p of the outcome belonging to a particular category, using the *logistic function*:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}$$

where $X_1, ..., X_p$ respresent the predictors and β_i are the regression coefficients.

In linear regression, coefficients that minimize the sum of squared residuals are chosen, while in logistic regression they will be estimated by the *maximum likelihood method*, which looks for values of β such that the predicted probability for each individual are the closest to the actually observed outcome. In mathematical terms, coefficients β are chosen to maximize the next function:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

Models will be built using a *training set*, containing 2/3 of the patients; classifier will be then validated using a new set of observations or *test set*, that will contain the rest of the patients.

Logistic models will be fit using different sets of features, including:

1) **original or "raw" features**, including *whole available sets* and *smaller sets of manually and automated selected features*;
2) **synthetic features**, including principal components extracted by principal component analysis and multiple factor analysis, as well as scores from clustering of variables

Logistic regression models will be fit in R using the function `multinom()` from the package `nnet` (B. Ripley, Venables, and Ripley 2016). In general, just two arguments are necessary: `formula` (formula to build the model with) and `data = trainSet` (dataset to fit the model at).

```
log.mod <- multinom(formula, data = trainSet)
```

2.4.2 Manual selection of features' sets

Sets of manually selected features will include:

1) A set of *basic features* (same for male and female patients), including those features that summarize the most important information in the dataset. Set of "basic" features will include:

   - `Age`
   - anthropometric features, such as `Weight, Height` and `BMI`
   - main features explaining bone quality of the patient, including total bone mineral density (`TotalBMD`) and minimum T-value (`minT_gral`)

- main features related with fat quality, including total amount of fat and fat mass ratio (`TotalFg, FMR, FMI`)
- main features related with muscle quality, such as total lean (muscle) mass and Apendicular lean mass (`TotalLg, Apendicularleanmas, FFMI`)

As already explained, features directly related to a certain disease will not be used in the prediction of that disease.

2) A set of *correlated features*, extracted from the results of the correlation analysis. For each disease, correlation between the variable used to establish presence of that disease and the rest of the variables will be studied, and the ten most-correlated variables will be extracted. Sets of correlated features may vary within sexes and within diseases.

2.4.3 Automated selection of features' sets

Stepwise and LASSO (least absolute shrinkage and selection operator) methods will be applied to automatically select variables, reducing dimensionality and/or multicollinearity within the dataset.

- **Stepwise**

Stepwise is a widely used method for selecting subsets of predictors. Stepwise cannot really avoid multicollinearity in a model (Dormann et al. 2013), but at least we expect it to help eliminating some of the redundant variables from it. Its name comes from "step", because the model selection is based on adding/removing one predictor at a time:

1) **Forward** stepwise selection begins with the "null" model (the one without any predictors) and *adds one predictor at a time*

2) **Backward** stepwise selection starts with a model containing all the available predictors and *removes one at a time*

3) **Mixed** stepwise selection begins with the "null" model and adds, at each step, the predictor that leads to a better model fit. However, and especially if multicollinearity exists, statistical significance of each parameter depends on the rest of the predictors; therefore, whenever we add or remove one predictor, significance of the rest may vary; mixed stepwise will then get rid of those predictors which are not significant any more.

Stepwise selection will be applied with the function `stepAIC()` from the MASS package (W. N. Venables and Ripley 2002), specifying the type of stepwise at the argument `direction`, using the logistic model previously fitted.

```
step.mod <- stepAIC(log.mod, direction = "backward", trace = 0)
```

Another way of dealing with multicollinearity is using robust logistic regression methods, such as *LASSO*. LASSO shrinks the coefficient estimates, forcing some of them to be zero and consequently, performing variable selection (James et al. 2013). LASSO will be applied using the argument `alpha = 1` while fitting a regression model with the function `glmnet()` from the package `glmnet` (Hastie and Qian 2014).

```
# Create predictor matrix + extract response variable (y)
X <- model.matrix(formula, data = trainSet)[,-1]
Xtest <- model.matrix(formula, data = testSet)[,-1]
y <- trainSet[,labelName]

set.seed(params$models.seed)
  if (disease == "bone") {
  family.type <- "multinomial"
} else {
  family.type <- "binomial"
}

# Extract lambda for best fit by CV
cv.lasso <- cv.glmnet(X, y, alpha = 1, type.measure = "mse",
                      nfold = 10, family = family.type)
bestLam <- cv.lasso$lambda.min # lambda for best result is stored at
                               # "lambda.min"
# Fit model
grid <- 10^seq(10, -2, length = 100) # create a grid
lasso.mod <- glmnet(X, y, alpha = 1, lambda = grid, # fit models using grid
                    type.multinomial = "grouped",
                    family = family.type)

# Predict using model built with best lambda value (bestLam)
predClass <- predict(lasso.mod, newx= Xtest,
                           s = bestLam, type = "class")
```

2.5 Building machine learning classifiers

Machine learning (ML) refers to the creation and evaluation of algorithms that facilitate tasks such as classification or prediction, taking advantage of computational power (Tarca et al. 2007). This kind of models are more flexible and expected to perform better, but they are also more complicated and hard to interpret, since it is often difficult to understand the relationship between the output and any of the predictors (Lantz 2015).

Two of the most widely used machine learning algorithms will be applied, namely Support Vector Machines (SVM) and Decision Trees (DT); those two algorithms were selected based on their usability, and the fact that they lead to reasonably good prediction results with low training times. Whole sets of features (or "raw" and "synthetic" type) will be used to fit machine learning models.

In order to achieve better predictions, some *ensemble learning approaches* will be considered as well; this technique consists on integrating the output from several learners to reduce their variance and better generalize to future prediction problems (Lantz 2015).

All **machine learning** models will be fitted using the function `train()` from the `caret` package (Kuhn and others 2008); algorithm to use will be specified at the argument `method`, with the following options:

- `method = "svmLinear"` for a SVM model with a linear kernel
- `method = "C5.0"` for a DT model, using the algorithm C5.0
- `method = "treebag"` for a bagged-tree algorithm
- `method = "rf"` for a random forest algorithm
- `method = "gbm"` for fitting a model with stochastic gradient boosting

Use of the function `train()` has the advantage of simplifying the process of parameter tuning. In order to select the "best" parameter values a resampling method will be carried out, specifying it at the argument `trControl`. Other arguments to be used are `metric` (metric to use at the fitting of models) and `preProc`, in case we want a pre-processing of the data to be done.

```
# trainControl() function to specify resampling method, used at the
#    parameter tuning
myControl <- trainControl(method="boot",
                          number=25,
                          classProbs = T)
# Metric used at the tuning will be "ROC" curve, if possible; if not,
#    "Accuracy" will be used (default value)
myMetric <- "ROC"
# Preprocessing will be used at SVM method
preProc <- c("center", "scale")

# Example for SMV (with linear kernel)
ml_method <- "svmLinear"
ml.model <- train(x = trainSet[,predictors], #"predictors"=s et of features
                                             # for the prediction of a
                                             # specific disease
                  y = trainSet[,labelName],  #"labelName"= output (diagnose)
                  method = ml_method,
                  metric = myMetric,
```

```
                    trControl = myControl,
                    preProc = preProc)
```

Ensembles with different type of learners (stacking-ensembles) will be built using functions from the package `caretEnsemble`[4] (Deane-Mayer and Knowles 2016) for the prediction of *lipodystrophy* and *low muscle mass*. Function `caretList()` will be used to fit a list of models, specified at the argument `methodList`. Function `caretStack()` takes those fitted models as input (at the argument `all.models`), and specifies the learner supervisor that should combine the predictions of the fitted models (argument `method`).

```
# Fit a list of different models
model.list <- caretList(x = trainSet[,predictors],
                        y = trainSet[,labelName],
                        trControl = myControl,
                        methodList = c("C5.0","nnet", "glmnet", "gbm",
                                       "svmLinear"),
                        metric = myMetric)

# Combine them (stack them) using a supervisor learner
stack.mod <- caretStack(all.models = model.list,
                        method="glmnet",
                        metric=myMetric,
                        trControl=myControl)
```

`caretEnsemble` is not available nowadays for the prediction of multi-class models. Therefore, an ensemble of models will be manually simulated for the prediction of bone abnormalities. The process will consist of fitting some models, calculate their predictions and manually extract the majority voting of all of them - in order to calculate their accuracy. AUC values will be calculated from the averate of each class' probability predictions, using the `multiclass.roc()` function.

Code used can be found at the end of Appendix II.

2.5.1 Support Vector Machines

Support Vector Machines (SVM) is a ML algorithm, widely used due to their high accuracy. "Support Vectors" are the points of each class closest to the maximum margin hyperplane, a line that leads to the greatest separation between classes (see Figure 6, extracted from Lantz (2015)). The main goals of SVM are then:

- to create a flat boundary (or "hyperplane") that divides the space, separating groups of similar classes;

[4]Package *caretEnsemble should* be installed from the repository at github (*devtools::install_github('zachmayer/caretEnsemble')* in order to assure the correct predictions of the model.

- to find the maximum margin hyperplane, as the line that will generalize the best to future data.

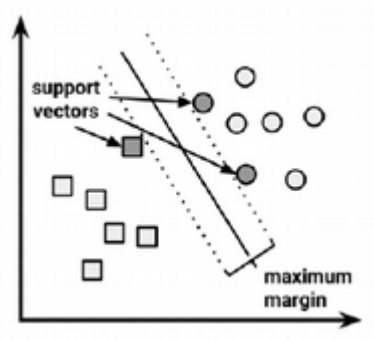


Figure 6: Representation of classes, support vectors and maximum margins. Figure extracted from Lantz (2015).

SVM can be used for linearly and non-linearly separable data, using for the last case kernels that make non-linear relationships to seem linear by adding new features or dimensions into the dataset. The adequate type of kernel to be used will depend on the learning task and the relationships between features, and there's no rule to choose a specific one. In this case, linear kernel will be used.

2.5.2 Decision Trees

Decision Trees (DT) are also widely used ML algorithms, because they are easy to fit, interpretable (they select variables and show the importance of them in the final output) and can be applied to all kind of data. DT use tree structures to model the relationship between features and outcomes; the process starts with a *root node* (for the input data), which is successively separated into *decision nodes* or "choices to be made". Data is then split across *branches* in new decision nodes until a "final decision" is made, reaching the output in the *terminal node*. In order to carry out that process, a splitting criterion is used, such as the information gain F:

$$F = S_1 - S_2$$

where S_1 represents the entropy (randomness) present within a set of class values, in the tree segment before the split, and S_2 represents the entropy value in the partition resulting from the split.

Entropy values range from zero, for a low class-diversity, to a maximum, for a very diverse partition; entropy depends then on the diversity of classes within the partitions.

2.5.3 Model Ensembling: boosting, bagging and stacking

Some ML models, such as DT, artificial neural networks (ANN) or SVM, are called to be "unstable" because they suffer from high variance; that means, applied to different splits of the same dataset, obtained results may be quite different. In contrast, other models, such as k-Nearest-Neighbors (kNN) or logistic regression ones, are known to be stable and have low variance, yielding similar results while applied to different datasets.

Ensemble machine learning consists on integrating the output from a diverse set of weak, "base" learners in order to achieve better predictions (Sarkar and Natarajan 2019) and reduce their variance, avoiding overfitting and in general, better generalizing to future prediction problems (Lantz 2015).

Learners of same or different type can be combined using different ensembling approaches, such as *bagging, boosting and stacking*; they all consist in fitting a number of "base learners" and combine their results into a final single prediction.

- **Bagging**

The bagging-ensemble technique, also known as "bootstrap aggregation" consists on fitting multiple independent models, usually of DT type, in k different training sets created by *bootstrap*. The predictions from the k sets will then be combined, creating a "single predictive model".

In this case, two different algorithms will be applied:

- *Treebag* algorithm
- *Random forests* (RF), which are a special case of bagged trees. Random forests fit DT in different boostrap samples, but also uses different (smaller) features' sets to build those trees.

- **Boosting**

Boosting consists in sequentially building multiple models, usually of DT type, where weighted votes are given to the fitted models depending on their performance, so that the final prediction will be more influenced by those that achieved a better performance (Lantz 2015). In this case, *Stochastic gradient boosting* will be used, being one of the most known boosting methods.[5]

- **Stacking**

The stacking-ensemble method consists on building multiple models, usually of different type, using a *supervisor model* to combine their predictions. In general, predictions obtained must have a low correlation in order to get a good result.

[5] *AdaBoost.M1* is the most well-known boosting method. However, training time was way too long, so its use was finally rejected.

2.6 Performance and models' selection criteria

All selected models will be fitted on training sets; prediction will be carried out on test sets. Once prediction has been done, performance of the models will be studied using confusion matrices and AUC values.

2.6.1 Confusion matrices

Confusion matrices include measurements like *overall accuracy, sensitivity and specificity*, being the balance between the last two an indicator of a good performance of the model. Definitions of those measurements can be found at the glossary. An example of a confusion matrix can be found at Figure 7.

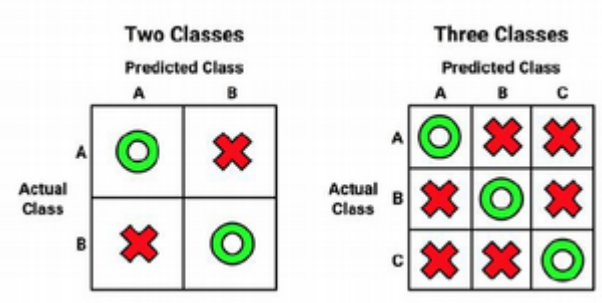


Figure 7: Example of confusion matrices of two and three classes, extracted from Lantz (2015).

Confusion matrices will be built with the function `confusionMatrix()` from the `caret` package (Kuhn and others 2008), using arguments `predLog` (predictions obtained by the function `predict()`) and `label` (vector that contains the class or output of interest):

```
# calculate predictions from model "log.mod"
predLog <- predict(log.mod, newdata = testSet, type = "class")
# obtain confusionmatrix
c <- confusionMatrix(predLog, testSet[,labelName])
```

2.6.2 Area under the ROC curve

The *area under the ROC curve (AUC)* will also be used to study the overall performance of the classifiers. AUC values range from 0.5 for a classifier with no predictive capacity, to 1.0 for a perfect classifier. In other words: the closer the value of AUC is to 1.0, the better will be the model's performance, being the model able to correctly identify presence of disease.

Overall performance	AUC value
Excepcional	0.9 to 1.0

Overall performance	AUC value
Great	0.8 to 0.9
Acceptable	0.7 to 0.8
Poor	0.6 to 0.7
No classification capacity	0.5 to 0.6

Although AUC criterion was at first developed to be used at binary classification, some approaches for multi-class classification exist, such as the Hand and Till one. More information on the method can be found at Hand and Till (2001).

AUC value for binary classification will be calculated by the function `roc()`, while function `multiclass.roc()` will be used to calculate AUC value for multi-class cases. Both functions belong to package `pROC()` (Robin et al. 2011). Main arguments to use are:

- `testSet[,labelName]` or the output to study
- `predProb`: probabilities for each class and observation
- `print.auc = T`: it will let us print the AUC result
- `$auc`: object where AUC value is stored

```
# AUC for binary class models
auc <- roc(testSet[,labelName], predProb, smoothed = TRUE, plot=T,
           auc.polygon=T, max.auc.polygon=TRUE, grid = T, print.auc=T)
auc <- round(auc2$auc,3)

# AUC for multi-class models
multiRoc <- multiclass.roc(testSet[,labelName], predProb, plot = F,
                           percent = T)
auc <- multiRoc$auc
auc <- round(auc*0.01, 3)
```

Final model selection will involve a trade-off over model performance, in terms of accuracy and AUC value. Simpler logistic models (i.e. models with "raw" and with less number of variables) will be preferred. Simplicity will not be an important selection criteria for machine learning models, since those are already very hard to interpret.

3 Results & Discussion

3.1 Presence of disease

Presence of disease was established based on cut-offs found at the bibliography (McClung 2003; Freitas et al. 2010; Cruz-Jentoft et al. 2018) (see Figure 8).

Male patients were observed to have higher rates of osteopenia and lipodystrophy while female patients showed much higher rates of low muscle mass. Most frequent combination of disease was that of low bone quality and low muscle mass for female patients, and low bone quality with lipodystrophy for male patients. Other combinations of disease were more rare to find. No-presence of disease was also rare.

	Osteoporosis	Osteopenia	LMM	Lipodystrophy	Osteo + LMM + Lipo	Osteo + LMM	Osteo + Lipo	LMM + Lipo	Normal
Male	31.6%	49.1%	16.7%	32.2%	3.4%	15.5%	27.1%	3.6%	13.2%
Female	33.3%	38.9%	54.8%	28.8%	9.3%	46.0%	20.3%	11.0%	12.2%

Figure 8: Presence of disease, alone and in combination, for male and female patients. Presence of disease was studied at the original set of patients.

A class-imbalance existed, with an over-representation of "normal" cases, and was corrected by the method SMOTE. This method did not work for multi-class cases, so original dataset was used for the prediction of bone quality, while balanced set was used for the prediction of lipodystrophy and low muscle mass. Class for low muscle mass was originally balanced for female patients; in that case, SMOTE was still used in order to increase the size of the dataset to work with. Final class-percentage, for each features' set can be found at Figure 9.

	Osteoporosis	Osteopenia	Lipodystrophy	Low muscle mass
Female	27.68%	38.98%	50%	50%
Male	19.31%	49.14%	50%	47.06%

Figure 9: Final presence of classes at dataset, after using SMOTE method for balance.

A comparison of body composition abnormalities in young versus old patients was also carried out. Bone quality of patients seemed to decrease with age, with higher rates of osteopenia at younger patients and osteoporosis at older ones, probably because osteopenia develops into osteoporosis. Lipodystrophy was more frequently observed at older patients. Low muscle

mass seemed to be a characteristic body abnormality among female patients, with similar rates at young and old patients (see Figure 10).

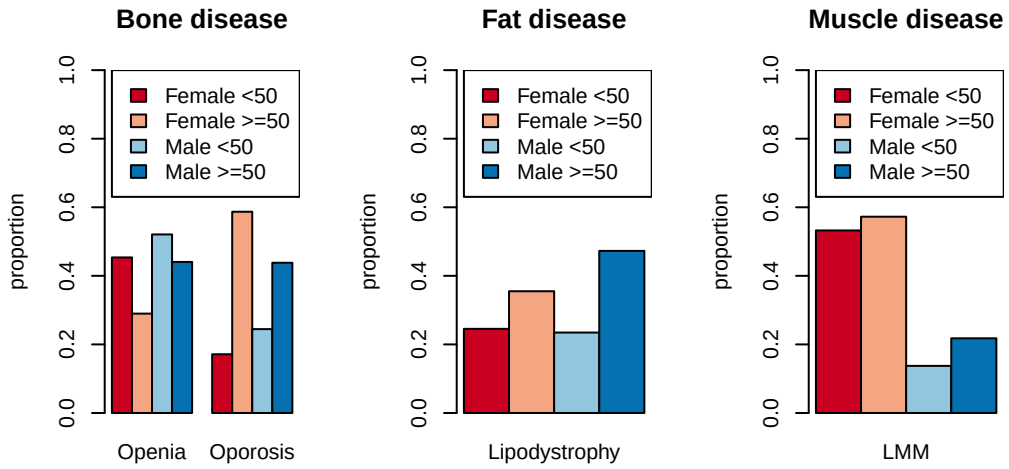


Figure 10: Presence of bone, fat and muscle disease. Study of the differences between young (<50 years) and old (>= 50) patients, for both female and male patients.

In general, the population under study showed much higher rates of morphologic disease than it would be expected for a "normal" (healthy) population, with rates at least twice as higher for bone-quality abnormalities (Tian et al. 2017; C. N. Y. Lee et al. 2015), as well as for low muscle mass (Shafiee et al. 2017). Lipodystrophy syndromes are typically observed in HIV+ patients but are very rare in normal population, with an estimated prevalence of 1.3 to 4.7 cases per million (Chiquette et al. 2017).

3.2 Exploratory analysis

3.2.1 Normality

Normality was visually stated, using boxplots (see Figure 11 for some of the results). In general, normality was observed at all features related to anthropometric, bone and muscle mass; some fat-related measurements had skewed distributions towards the right, probably caused by the abnormal redistribution of fat along the body typically observed in HIV+ patients. An histogram of fat mass ratio can be found at Figure 12.

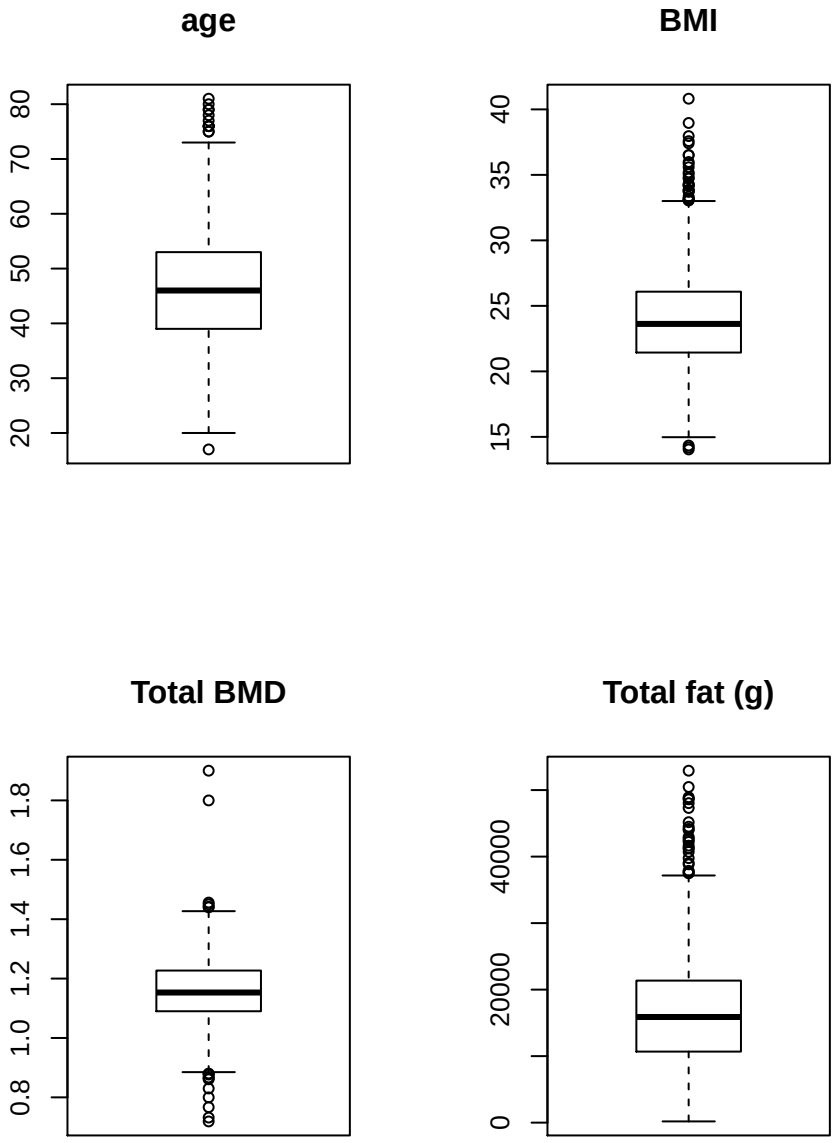


Figure 11: Boxplots used to study the distribution of some of the main features. Normal distribution was accepted for all of them.

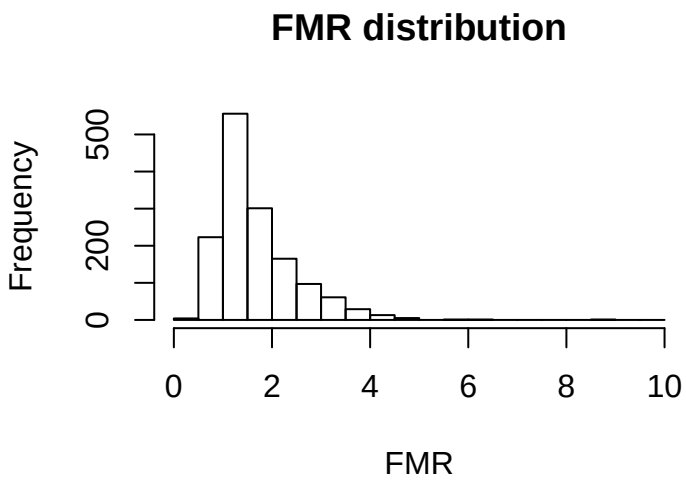


Figure 12: Histogram of FMR, skewed towards the right; probably caused by the abnormal fat redistribution typically observed on patients with HIV.

3.2.2 Correlation study

Pearson's coefficient was used to build a correlogram of a selected set of variables, including age, anthropometric measurements, bone mineral density measurements (from the hip and the spine), as well as the main fat and lean measurements and ratios. Correlogram can be found at Figure 13. In general, many variables were found to be highly correlated to each other, which implied the *presence of multicollinearity within them and the need of applying features' selection and/or dimensionality reduction techniques as a first step - prior to building classifiers.*

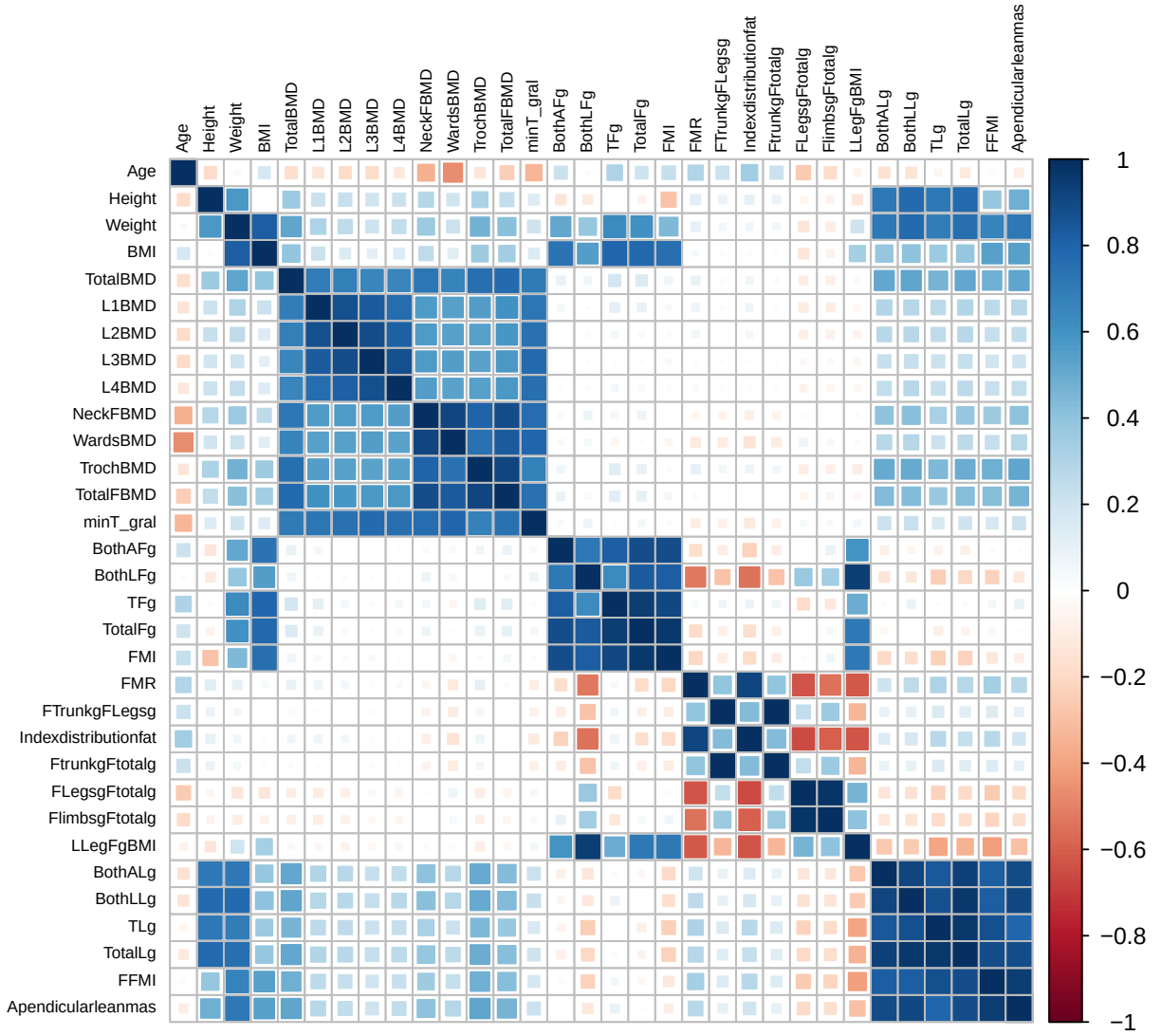


Figure 13: Correlogram for the most representative variables. Blue and red colours indicate positive and negative correlation, respectively; darker shades indicate stronger correlation.

Main conclusions extracted from the correlogram were:

- *The older the patient, the worse the bone quality* and therefore, the higher the risk of developing osteopenia/osteoporosis. As it can be observed, age is inversely correlated with the minimum *T-score* and some of the hip bone measurements.
- *A better body condition* (in terms of height, weight, higher amount of fat and muscle) *seems to be correlated with a better bone quality.*

As previously stated, three of the available features were used to define presence/absence of the morphologic abnormalities under study:

- Presence of osteoporosis/osteopenia was defined by the minimum T-value, extracted from all bone measurements (`minT_gral`).
- Presence of lipodystrophy was diagnosed by the FMR
- Presence of low muscle mass was diagnosed by the ASM (`Apendicularleanmas`)

Correlation coefficients were calculated separately for male and female datasets, between each of those three features and all variables not directly related with the correspondent disease. Variables that were not common to both sexes have been coloured in green (see Figure 14).

minT_gral				***FMR***				***Apendicularleanmas***			
Female		**Male**		**Female**		**Male**		**Female**		**Male**	
Age	-0.42	*BothALg*	0.37	*FFMI*	0.22	*Age*	0.36	*Weight*	0.67	*BMI*	0.64
RALg	0.36	*BothLLg*	0.36	*TLg*	0.20	*FFMI*	0.20	*BMI*	0.62	*Weight*	0.62
BothALg	0.34	*LALg*	0.36	*Age*	0.19	*WardsBMD*	-0.17	*TrochBMD*	0.50	*TotalBMD*	0.46
LLLg	0.33	*RALg*	0.36	*TotalLg*	0.18	*WardsT*	-0.17	*TotalFBMD*	0.49	*TrochBMD*	0.43
RLLg	0.33	*RLLg*	0.36	*Ap.mas*	0.15	*NeckFT*	-0.14	*TotalFT*	0.47	*TrochT*	0.42
BothLLg	0.33	*LLLg*	0.36	*LLLg*	0.14	*NeckFBMD*	-0.14	*TrochT*	0.46	*TotalFT*	0.42
TotalLg	0.31	*TotalLg*	0.35	*BothLLg*	0.14	*TLg*	0.14	*TotalBMD*	0.41	*TotalFBMD*	0.42
Ap.mas	0.31	*Ap.mas*	0.32	*L1Z*	0.14	*minT_hip*	0.11	*NeckFBMD*	0.41	*NeckFT*	0.37
LALg	0.30	*Age*	-0.32	*RLLg*	0.13	*minT_gral*	0.10	*NeckFT*	0.38	*NeckFBMD*	0.37
FFMI	0.25	*TLg*	0.26	*L1L4Z*	0.1	*Ap.mas*	0.10	*LAFg*	0.38	*minT_hip*	0.36

Figure 14: Calculated correlation coefficients between variables used to define presence of disease and the rest of not-directly-related variables. Variables that are not common to female and male patients have been coloured.

- Similar results at male and female datasets were found for the variables that correlate to `minT_gral`. Variables included `Age` (inversely correlated) and *lean mass features*, in accordance with the conclusions extracted from the correlogram.
- `FMR` showed very low correlation values with the rest of the variables. Results differ within male and female patients, the only common features being `Age` and three muscle measurements (`FFMI, TLg, Apendicularleanmas`). `FMR` was more correlated to hip bone measurements for male patients, and to spine bone measurements for female ones.
- Male and female patients showed similar results in terms of correlation between `Apendicularleanmas` and the rest of non-related variables, with most correlated variables including anthropometric measurements (`Weight`, `BMI`) and hip measurements of BMD, as well as T-values. Low muscle mass doesn't seem to be influenced by age.

Sets of correlated variables from Figure 14 were then used to build logistic regression models.

3.2.3 Cluster analysis and split of the dataset

The *Silhouette method* was used to study clustering of variables, with two clusters being the optimal number, as it can be seen at Figure 15. Then, *K-means clustering method* was used to build those two clusters.

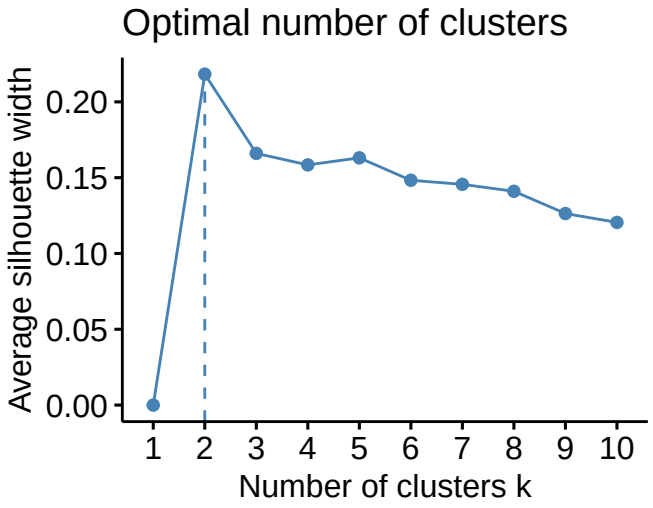


Figure 15: Optimal number of clusters extracted by the Silhouette method, for the original dataset.

Clusters were analyzed; they seemed to mainly split patients with low bone quality from the rest, with no other clear or useful separation - for instance, within men and women, patients of different age, etc. In other words, *division of patients between clusters did not seem to be of any advantage for the main aim of the study* and would have worsen the usability of the final classifiers.

It is widely accepted that men and women have different morphologic characteristics, so different prediction models are often built for men and women - at least if enough number of patients are available (see example at A. P. dos Santos et al. (2018)). Also, it was shown how male and female patients had different rates of disease; therefore, original dataset was finally split into one male dataset, containing 1103 patients, and one female dataset with 354 patients.

3.3 Dimensionality reduction methods

A big amount of features were originally available, with many of them being correlated to each other. Therefore, use of dimensionality reduction methods was explored, its aim being reducing the number of features of the datasets used at modelling, while avoiding for

multicollinearity. Those *methods were applied to each of the six available datasets (one per each sex and for the prediction of each disease under study)*, and one set of "new" (synthetic) variables was then obtained, for each case. Those new variables were then used as "normal" predictors at the construction of classification models.

At first, "best" number of synthetic variables was decided - using scree plots for PCA and MFA, or using the stability method for clustering of variables, as already explained at the methods' chapter. Models built with that "best" amount of variables did not have prediction capacity, so a fixed number (15) was finally extracted and used at modelling, yielding better results.

Being the main aim of the study the prediction of body composition abnormalities among an HIV+ population, *focus was set on extracting the synthetic variables to use at the prediction of disease*. Therefore, and as previously stated, dimensionality reduction methods will be applied to each possible case and synthetic variables will be extracted, but just one example will be included at this report: methods applied to dataset of female patients including features for the prediction of lipodystrophy. An analysis of all possible cases (18) would be outside the scope of this study.

3.3.1 Principal Component Analysis

PCA was applied using the function `prcomp()` and "best" number of principal components was visually stated by using a *scree plot* (see example at Figure 16); PCs over the "elbow" were at first extracted, being those able to explain most of the variance contained in the original dataset (over 90%, see Figure 17). However, and as already mentioned, "best" number of principal components were later on found to not have much predictive capacity building classification models, so 15 PCs were finally extracted and used at the models' fitting process.

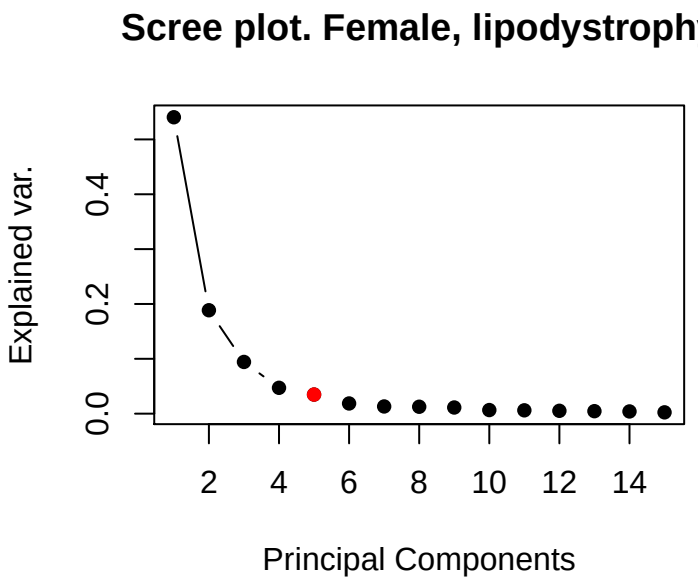


Figure 16: Scree plot of the first 15 principal components, for dataset of female patients for the prediction of lipodystrophy. PCs over the 'elbow' (red dot) were at first selected as the ones containing most of the variability in the dataset.

```
Importance of first k=15 (out of 47) components:
                            PC1    PC2     PC3     PC4
Standard deviation     5.0399 2.9770 2.10485 1.48835
Proportion of Variance 0.5404 0.1886 0.09426 0.04713
Cumulative Proportion  0.5404 0.7290 0.82326 0.87040
                            PC5     PC6     PC7     PC8
Standard deviation     1.2789 0.93512 0.78545 0.77015
Proportion of Variance 0.0348 0.01861 0.01313 0.01262
Cumulative Proportion  0.9052 0.92380 0.93693 0.94955
                            PC9    PC10    PC11   PC12
Standard deviation     0.72719 0.55510 0.53519 0.4945
Proportion of Variance 0.01125 0.00656 0.00609 0.0052
Cumulative Proportion  0.96080 0.96735 0.97345 0.9787
                           PC13    PC14    PC15
Standard deviation     0.46123 0.43573 0.33414
Proportion of Variance 0.00453 0.00404 0.00238
Cumulative Proportion  0.98318 0.98722 0.98959
```

Figure 17: Summary of principal components, in terms of explained variability and cummulative variability. For this example (female + lipodystrophy) the first 5 components were able to explain 90% of the total variability.

A representation of the two first PCs was also studied. At Figure 18, an example for female patients and presence/absense of lipodystrophy can be found. No clear separation exists for both dimensions, probably because dataset under study is too complicated. PCA can therefore not be used as an exploration analysis.

```
## Warning: Duplicated aesthetics after name standardisation:
```

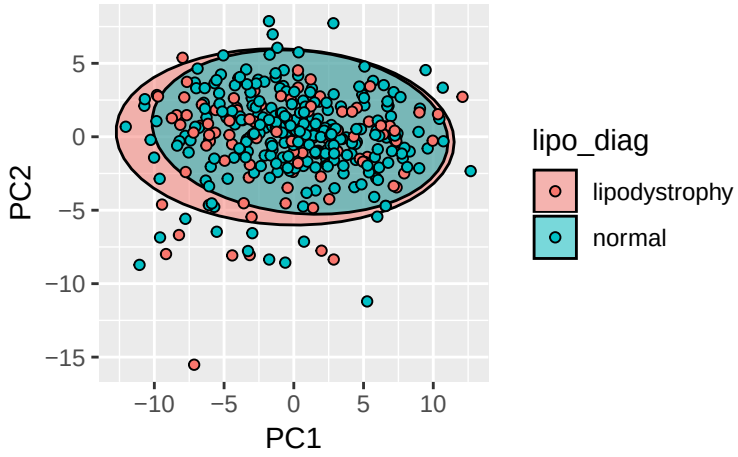


Figure 18: Graphic representation of two first PCs, for female and lipodystrophy. No clear separation exists for those two first dimensions.

Results regarding composition of each principal component (in terms of correlation between each PC and the original features) were inconclusive, with no specific set of variables being part of each dimension. Therefore, results have not been included. The reason may be the high number of features available, as well as the overall complexity of the dataset.

3.3.2 Clustering of variables

Variables were grouped using hierarchical clustering and best number of clusters was calculated by studying their stability, as stated at the methodology; stability of partitions can be observed at Figure 19.

The "best" (i.e. more stable) number of clusters was very low for all cases under study and models built with them did not show much prediction capacity. Since higher number of clusters also had a good stability (see example at Figure 19), 15 clusters were finally extracted for all cases. Scores from those 15 clusters were then used as predictors at the model building

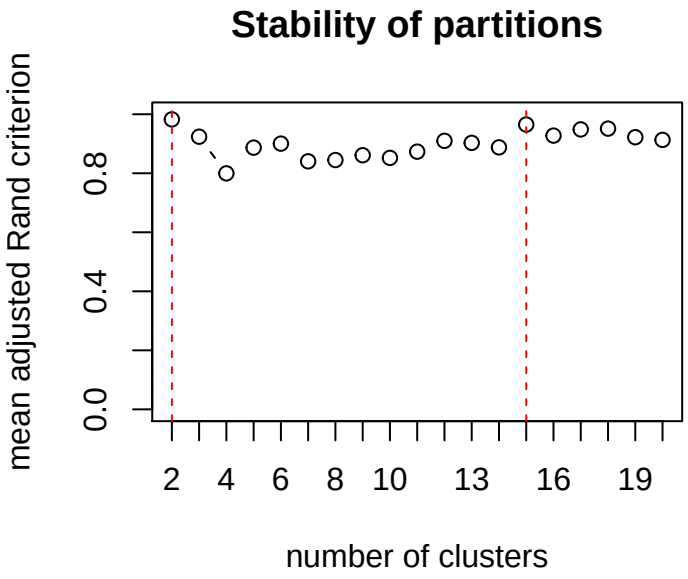


Figure 19: Graphic study of the stability of cluster partitions of features, example for female and lipodystrophy dataset. A fix number of 15 clusters was finally chosen, for all diseases.

3.3.3 Multiple Factor Analysis

As a result of applying MFA, one set of synthetic variables was obtained; as for the other dimensionality reduction methods, those were then used to fit classification models.

Figure 20 contains an example of the scree plot for the MFA analysis, using the female dataset for the prediction of lipodystrophy. Results in terms of "best" number of MFA components were similar to those obtained from PCA (see Figure 16). Synthetic variables to use were, in this case, the coordinates for each individual of the studied dataset, stored at `res.mfa$ind$coord`.

As for PCA and clustering of variables, 15 components were finally extracted, since models built with the "best" number of dimensions did not have much prediction capacity.

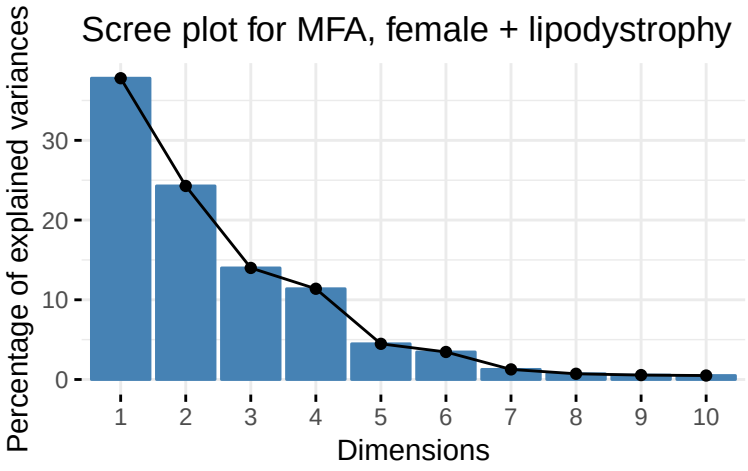


Figure 20: Scree plot for the results of MFA, for female and lipodystrophy. Five to six dimensions would be selected as representative for the dataset.

MFA was a really interesting method that let us extract results for groups, variables and individuals (observations). For instance, *correlations between groups and dimensions* could be visually studied, using the function `fviz_contrib()`. For the example under study, it could be easily observed how body measurements, especially muscle-related ones, contributed the most to the first dimension (see Figure 21), while `Age` was the one most contributing to the second one (see Figure 22).

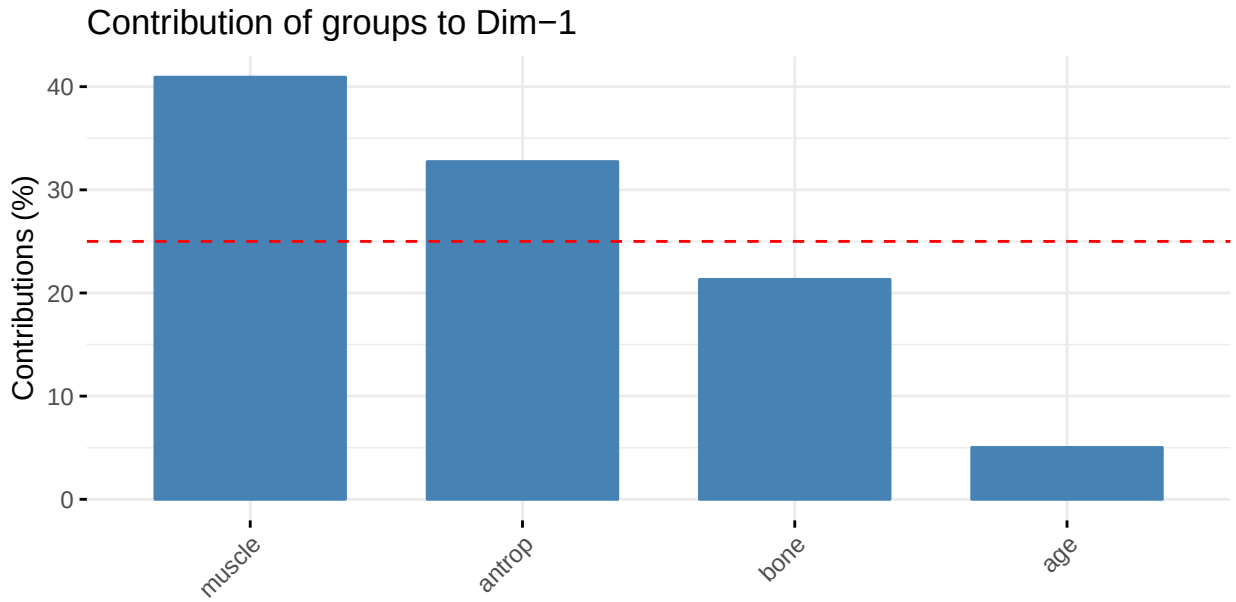


Figure 21: Contribution of groups to the first dimension. Example for female and lipodystrophy dataset.

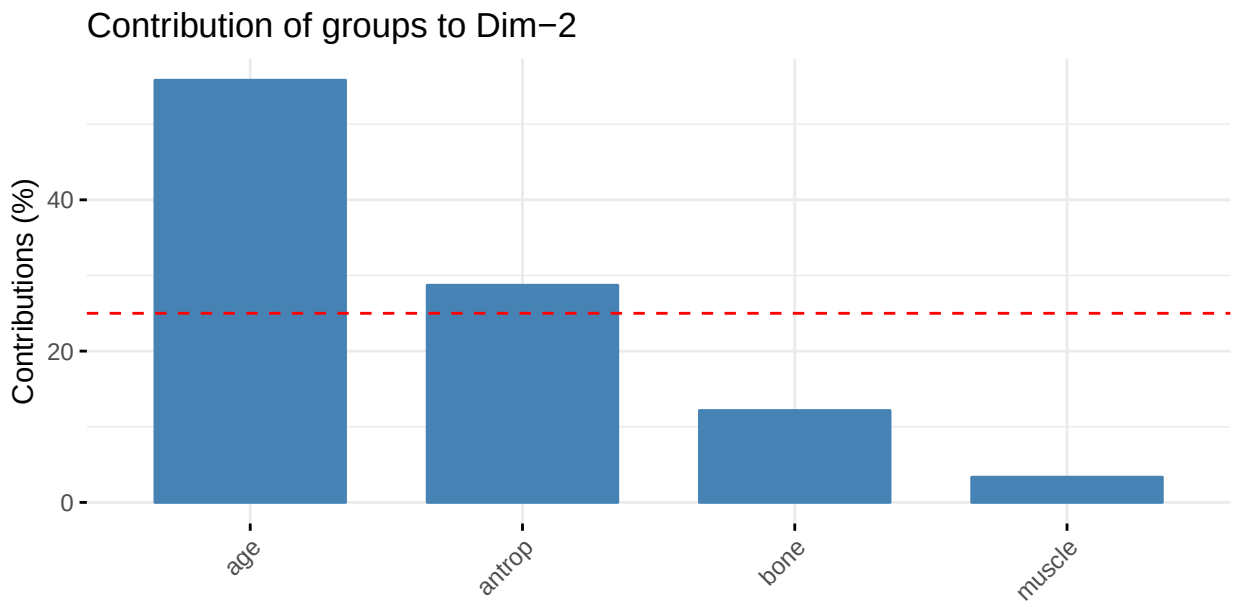


Figure 22: Contribution of groups to the second dimension of MFA analysis. Example for female and lipodystrophy dataset.

Contribution of variables to the dimensions was also graphically studied. For the example under study, the first component seemed to be mainly influenced by anthropometric measurements (`Weight, BMI, Height`, see Figure 23), while the second one was mainly influenced by the variable `Age` (see Figure 24).

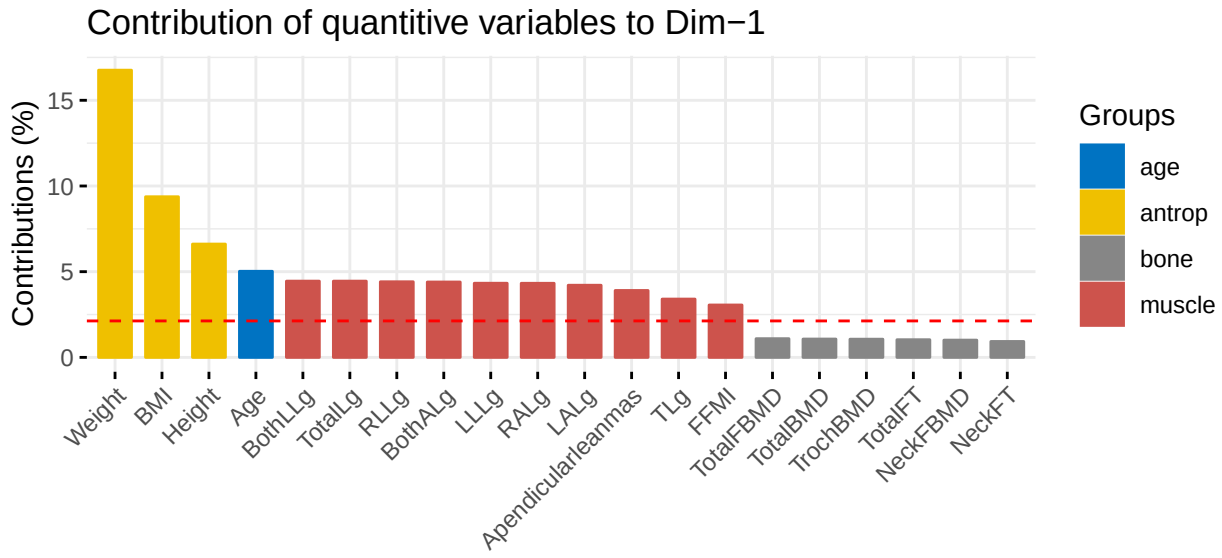


Figure 23: Contribution of variables to the first dimension of MFA. Example for female and lipodystrophy dataset.

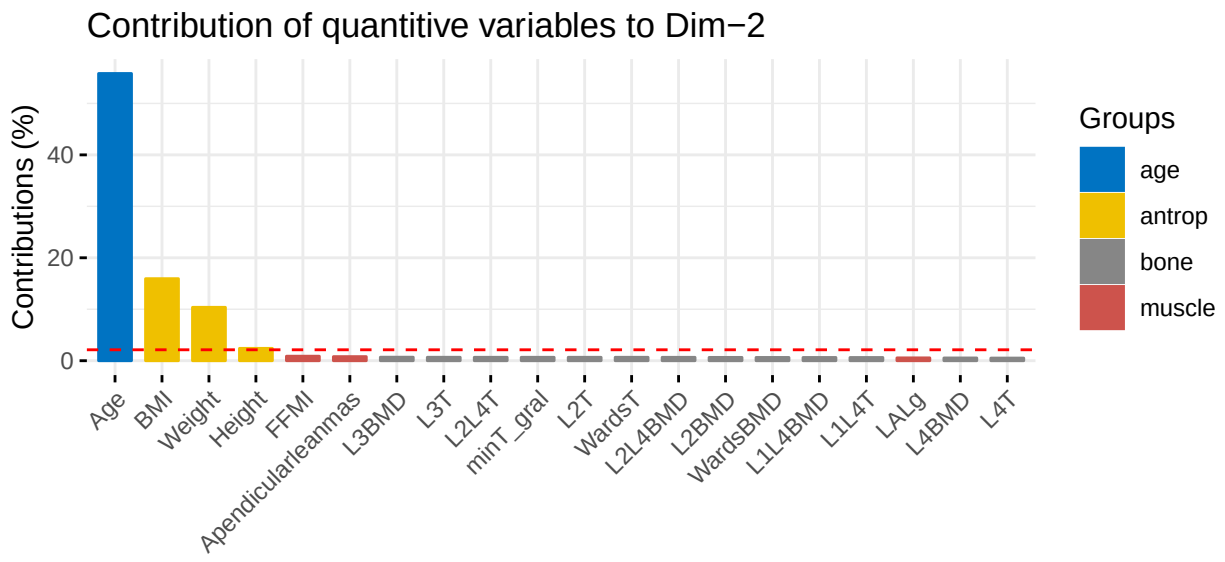


Figure 24: Contribution of variables to the second dimension of MFA. Example for female and lipodystrophy dataset.

3.4 Modelling

Logistic and machine learning models were built on different sets of features, depending on the disease under prediction. Models were fitted for male and female patients separately, using training sets, and validated using test sets.

Logistic regression models were built using: 1) *all the available original features* (one set per each disease); 2) two different *smaller sets of manually (original) selected features*; 3) three different sets of *synthetic variables*, extracted by PCA, clustering of variables and MFA.

Automated feature-selection methods were then applied to the fitted logistic models and backwards-stepwise method was finally selected, since it needed lower computing timings and led to models with similar accuracy as those obtained by the "mixed" method. Stepwise selection was able to create simpler models, in terms of number of variables, with similar or even better performance than the ones containing the whole features' sets. LASSO method did not improve performance, compared with the original models or the ones obtained by the stepwise method, and had additional disadvantages, like difficulties to extract sets of variables finally included in the model and estimation of AUC value; therefore, LASSO results were finally discarded and have not been included in this report.

Main results observed from logistic regression models were:

- Models built from *original (raw) variables* performed pretty well for all cases, both using the original complete set and the sets of features selected by stepwise.

- Models built with *manually selected sets of variables* performed reasonably well using the "basic ones", while performance of models using "correlated variables" was much worse. The reason may be that the "basic" set of variables were a good representation of the whole feature set and therefore, included most of the variability from the whole set of variables, while the "correlated" ones were often of the same type and therefore, not a good representation of the whole features' set.

- Use of *synthetic variables* did not seem to improve the performance of the logistic regression models, the only exception being clusters of variables for the prediction of lipodystrophy at male patients (see pink-coloured row at Figure 33 at Appendix I). Models built from original variables were then preferred.

Machine learning models were built on *whole sets of variables*, both of original and synthetic type. For those models containing parameters, training control was used to carry out an automatic tuning, as previously explained. In this case, no feature selection was carried out and models were not interpretable.

Trained models used *support vector machines* (with a linear kernel), *decision trees* (using the algorithm C5.0), *random forests, bagging of trees* and *stochastic gradient boosting* algorithms. All models were built using the function `train()` from the package `caret`, specifing the method used. Using the same function for the training of all models simplified the flow analysis and made models' results comparable. Furthermore, `caret` package had some advantages over other model-specific packages, like making tuning process easier, which was important taking into account the amount of cases under study.

Functions from package `ensembleCaret` were used to "stack" learners; that means, fit different type of models, not correlated between them, and combine their outputs using a supervisor learner. Since those could not be applied to the prediction of multi-class cases (osteoporosis/osteopenia/normal), a *majority vote approach* was used, with results being manually extracted. These method consisted of fitting different models and calculating the most predicted class, comparing it with the original output in order to extract the overall accuracy of the ensemble.

Performance of models was studied and **best ones were selected mainly based on their overall accuracy and AUC value**. For logistic models of similar accuracy and AUC values, those being simpler were preferred. Simplicity was not taken into account to choose the "best" machine learning models.

Customized functions and loops were programmed to fit models and get the final results, improving the usability of the code. Functions included:

- `select_sex`: to select the patients of interest (female/male).

- `select_disease`: to select the features' sets of interest, which depend on the disease that we would like to predict. Disease options include "fat" (for the prediction of lipodystrophy), "bone" (for the prediction of osteoporosis/osteopenia) and "muscle" (for the prediction of low muscle mass).

- `select_formula`: to select formula (variables) to use at the model fitting, with options being: 1) "all", to use all available variables; 2) "basic" set of raw features; 3) "corr" (correlated) sets of raw features.
- `select_type_data`: integrates the previous functions to select sex, disease, formula and data type, being the data type options "raw", "pca", "clust" (clusters of variables) or "mfa".
- `balance_and_split`: to balance classes and split dataset into training and test sets.
- `log_results`: to fit logistic regression models, for a determined sex, disease under study and type of data.
- `ml_results`: to train ML models, using the default tuning.

Loops were programmed in order to apply functions to several/all possible cases at once. Loops were also used to apply stepwise method to logistic regression models built by `log_results`. The whole set of customized functions and loops can be found at the Appendix II.

Results obtained by applying those functions and loops have been summarized at Figures 31 and 32 for the prediction of bone-related abnormalities by logistic and ML models (respectively); Figures 33 and 34 for the prediction of lipodystrophy; and Figures 35 and 36 for the prediction of low muscle mass, for both male and female patients. At the tables, "best" models have been coloured in grey, in case original variables were used to fit the model, and pink, for models using synthetic variables. All those tables can be found at *Appendix I*.

In general, it was seen that **best results were obtained by models built with raw variables and synthetic variables from MFA analysis**. Results of those two cases will be shown, for male and female patients and for the prediction of each disease.

3.4.1 Results for the prediction of bone disease

Bone-quality related abnormalities (i.e. osteoporosis/osteopenia) were hard to predict; fitted models did not get a high accuracy, although overall performance in terms of AUC value for some models was acceptable, for both male and female patients.

Two logistic regression models were finally selected as the "best" ones, both in terms of interpretability and accuracy. Machine learning models were not able to improve the performance, with the additional disadvantage of not being interpretable (see Figure 25).

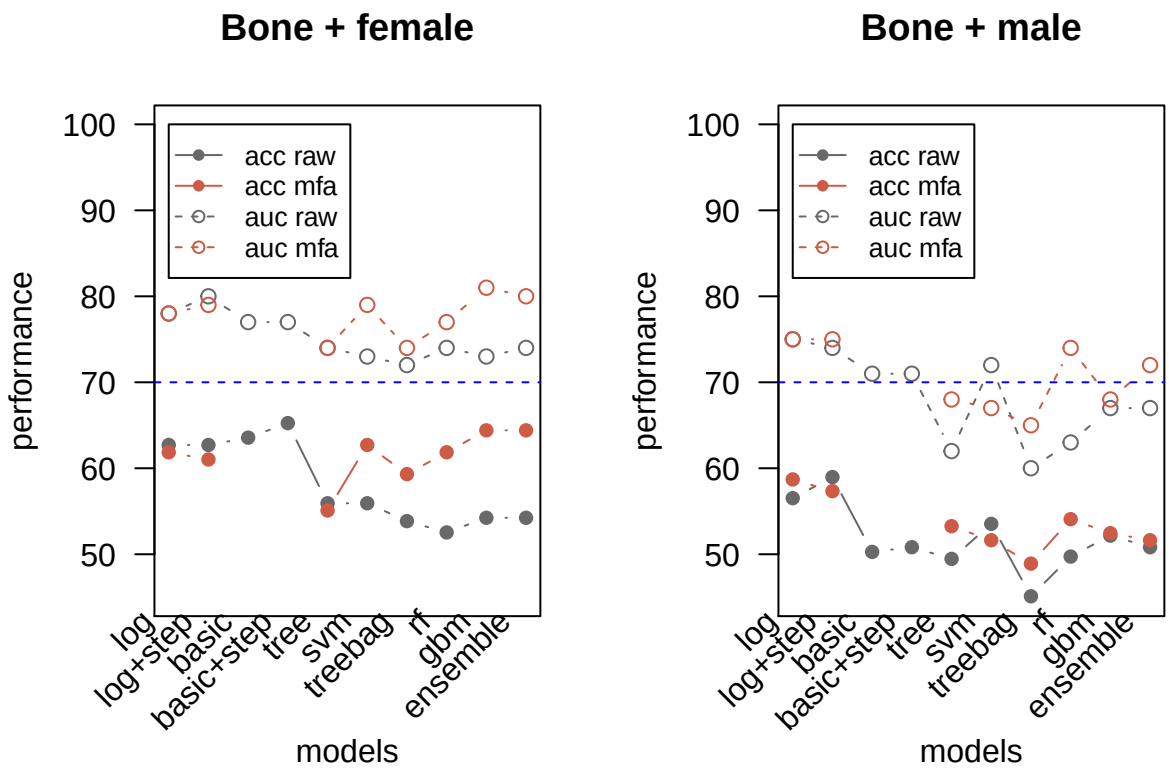


Figure 25: Performance of main models at the prediction of low bone quality, for female (left) and male (right). Filled dots show accuracy values, while empty ones show AUC values (in a scale from 50 to 100, to make them comparable).

A model with 65.25% accuracy and AUC value of 0.77 for *female* patients was finally selected; it consisted of nine variables from the set of "basic" features. "Best" model for the *male* dataset had an accuracy of 58.97% and AUC value of 0.74, and contained 17 variables; it was obtained by applying stepwise at the original features' set.

Variables included in both models, as well as their relative importance, can be observed at Figure 26. Most important variables included in the model, for both male and female patients, were rates related to weight, fat and muscle condition (`BMI`, `FMI`, `FFMI`, `FMR`, `Indexdistributionfat`). Surprisingly, age did not seem to have a great importance in the models, although it was indeed included.

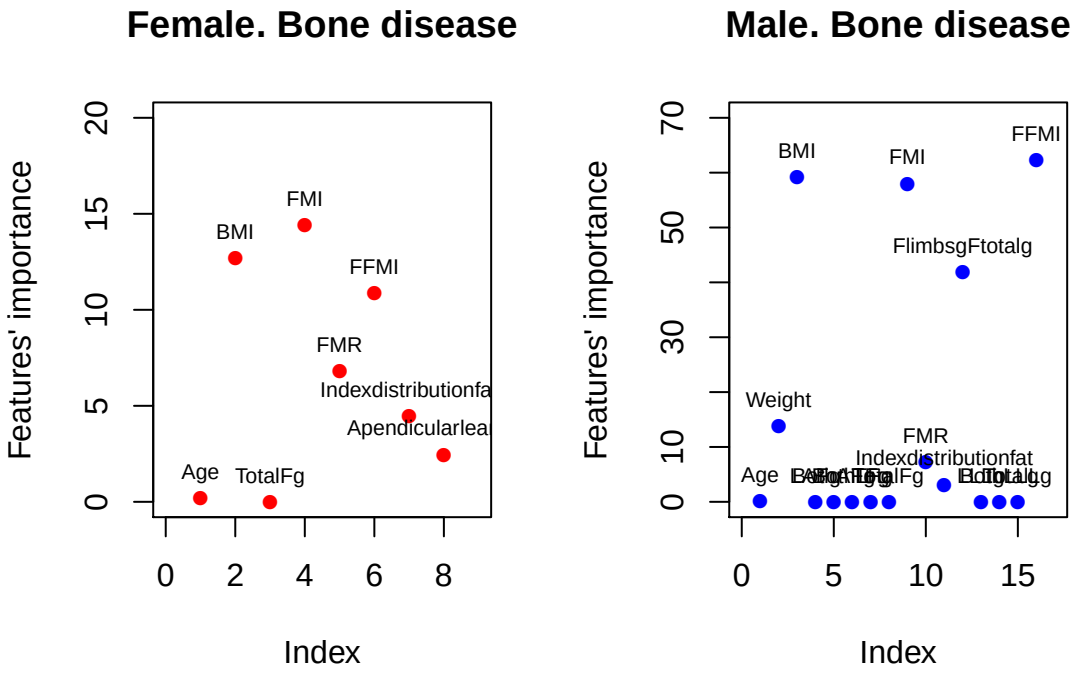


Figure 26: Importance of variables in the prediction of bone disease, for female (left) and male (right) patients.

3.4.2 Results for the prediction of lipodystrophy

Models for the prediction of Lipodystrophy were more accurate than bone-quality ones. As it can be seen at Figure 27, use of machine learning methods greatly improved performance for both female and male patients.

- Models for male patients showed very similar results for raw and MFA variables, with best results (in terms of accuracy and AUC values) corresponding to RF, ensemble and C5.0 algorithms.
- Results for female patients were more variable, with worse performance of logistic models and higher performance of ML models using MFA variables, especially RF and treebag algorithms.

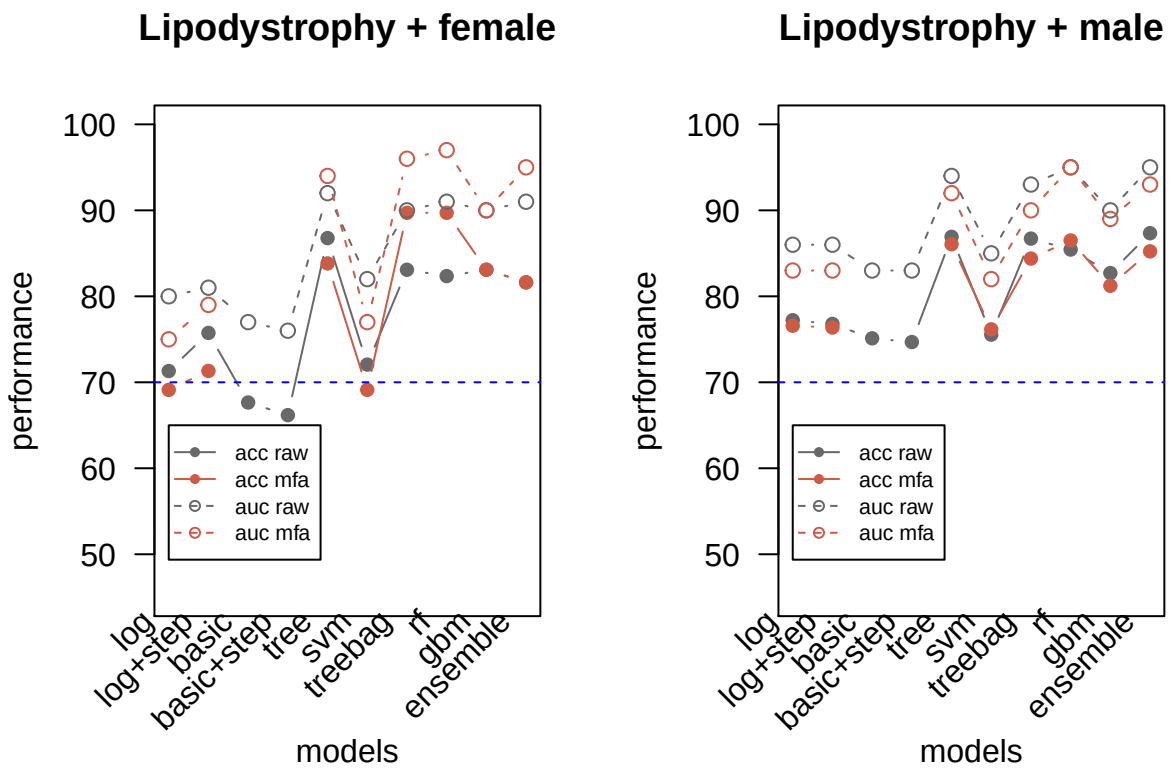


Figure 27: Performance of main models at the prediction of lipodystrophy, for female (left) and male (right).

A model with 75.74 % accuracy and AUC value of 0.81 for *female* patients was finally selected as the "best" one in terms of interpretability; it contained 23 of the original variables, selected by stepwise. "Best" model for the *male* dataset contained 25 variables, also selected by stepwise, and had an accuracy of 76.79 % and AUC value of 0.86, with a good balance between sensitivity (78.46%) and specificity (75%).

Variables included in the models, as well as their relative importance, can be observed at Figure 28. Again, results in terms of variable importance seem to be similar for both sexes; variables with the bigger weight on the models were `height` and some of the bone variables at the spine, as well as the bone mineral density at the tronch (hip).

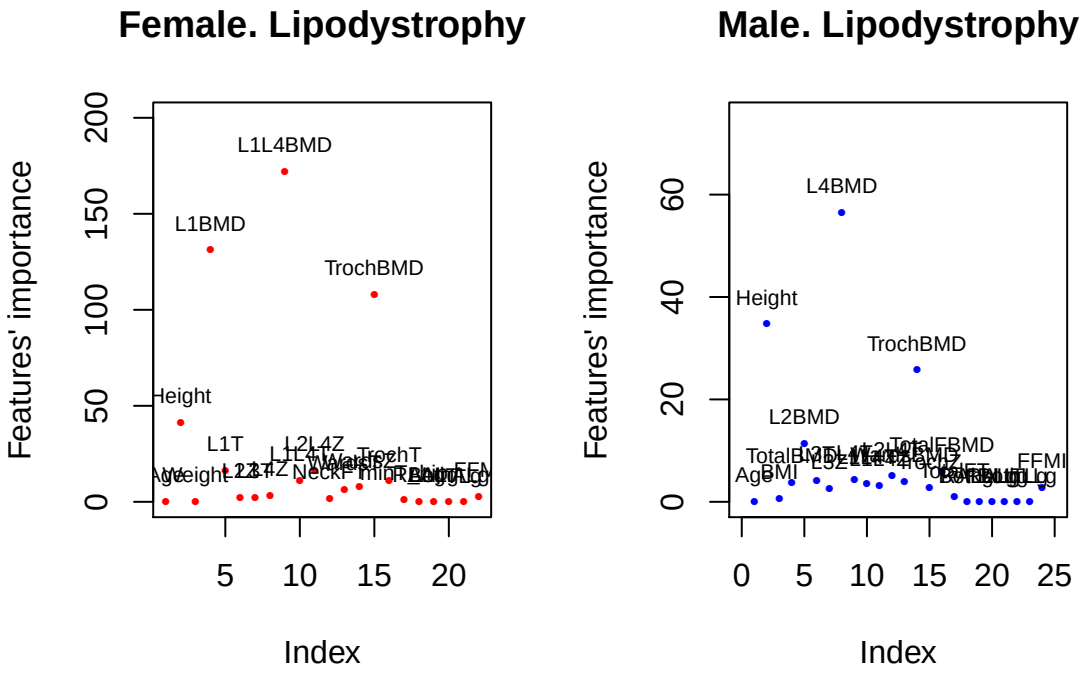


Figure 28: Importance of variables in the prediction of lipodystrophy, for female (left) and male (right) patients.

As already mentioned, machine learning models greatly improved the performance, and two of them were selected as the "best" ones in terms of accuracy. Use of the random forest algorithm and synthetic variables from MFA obtained the best results for the *female* dataset, improving accuracy and AUC values in around 15%. For the *male* dataset, best results were obtained by the stacking ensemble applied to the original set of variables, which increased the accuracy and AUC values in roughly a 10% with respect to the logistic regression models (see Figure 27).

3.4.3 Results for the prediction of low muscle mass

Best results were obtained for the prediction of low muscle mass. Main results (in terms of accuracy and AUC value) for raw and MFA variables can be found at Figure 29.

Both, logistic and machine learning models performed very well and results were very good - and stable - for all the models built from raw and MFA variables (see Figure 29). Again, logistic models were preferred in terms of interpretability while machine learning ones were selected as the best accurate ones Both kind of models showed an excellent performance, with AUC values over 0.95; ensembles were able to get AUC values of roughly 1.

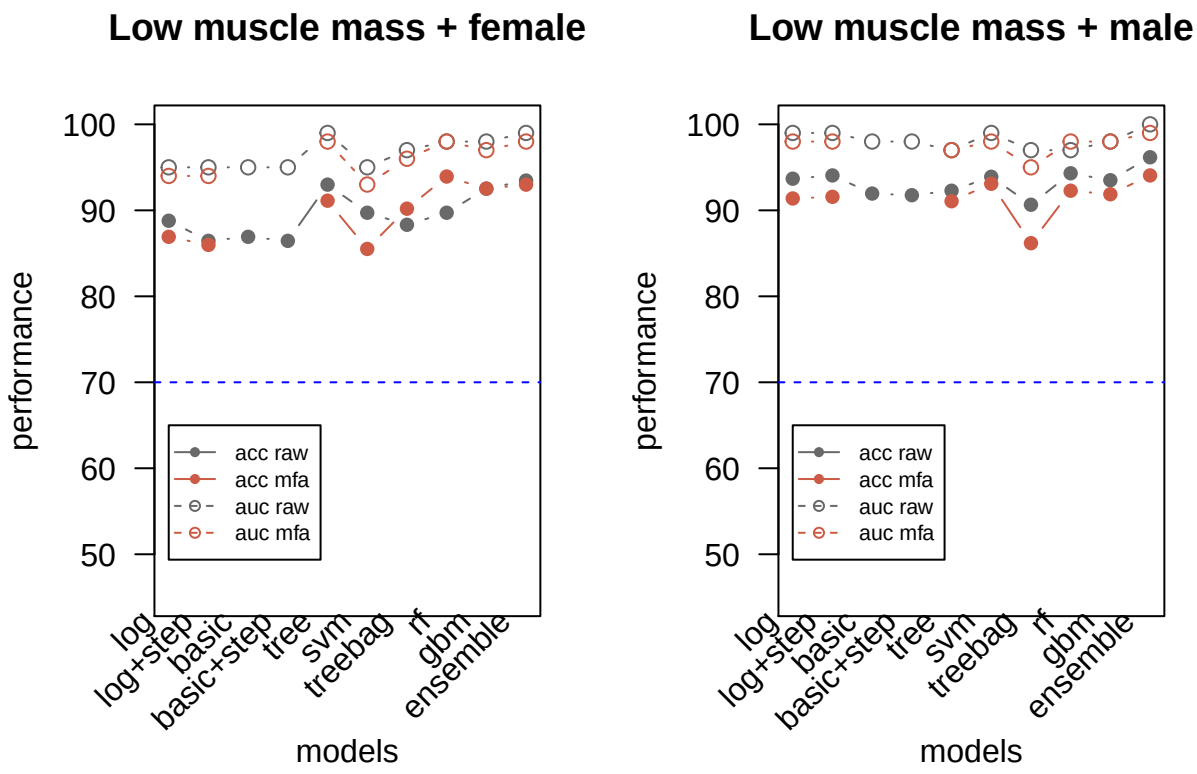


Figure 29: Performance of main models at the prediction of low muscle mass, for female (left) and male (right).

Logistic selected models contained ten basic features for the female dataset, and just six for the male dataset, with `BMI` and `FMI` being two of the most important ones, and with `height` also greatly influencing the model built at the female dataset (see Figure 30). Those models did not have the highest accuracy and AUC values, but they showed very good results and were very simple, so they were preferred over more complicated ones.

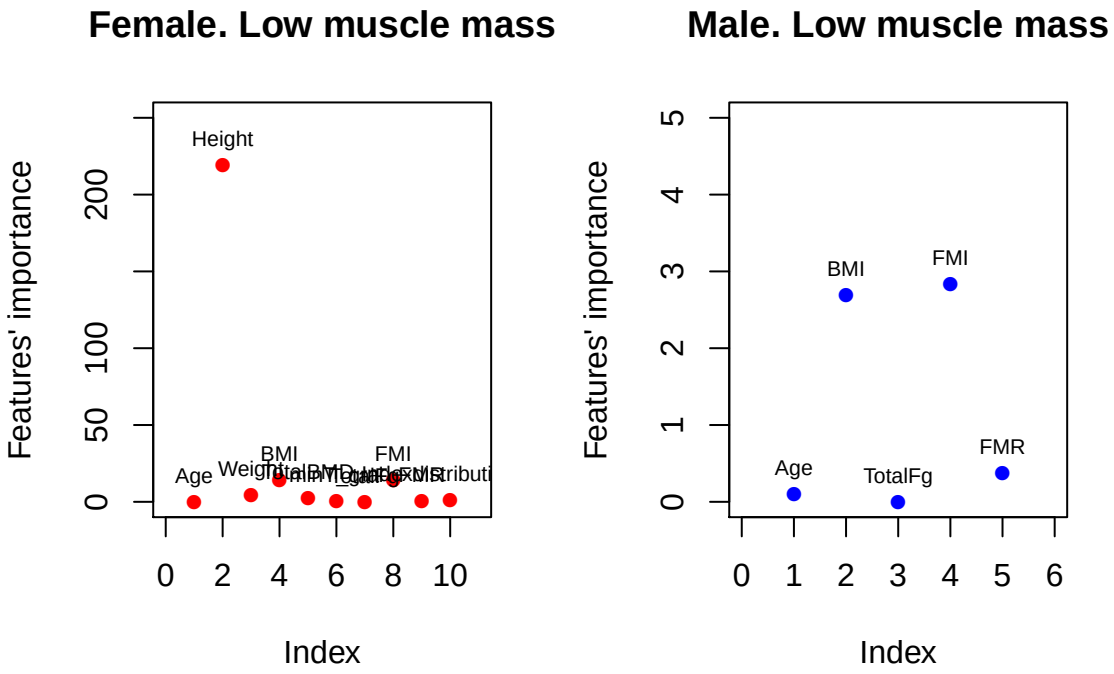


Figure 30: Importance of variables in the prediction of low muscle mass, for female (left) and male (right) patients. .

As already mentioned, machine learning models were able to improve the performance, with higher accuracy and AUC values in both female and male datasets (see Figure 29); selected models correspond, for both sexes, to stacking ensemble applied to the original set of variables. Model accuracy increased in 6% for female patients and 4% for male patients, with respect to the logistic models; AUC value also increased, as well as sensitivity value, showing the capacity of the models to predict almost all true positive cases ("presence of disease").

4 Conclusions & Future Works

The main aim of this study was to predict, for a set of male and female patients with HIV, the presence/absense of different type of body composition abnormalities related to bone quality (osteoporosis/osteopenia), fat-redistribution (lipodystrophy) and low muscle mass. Different sets of features were used for the prediction of each disease, namely those ones not directly related to the disease itself.

First, an **exploratory analysis** was done. *Rates of disease* were found to be much higher than the ones expected for a "healthy" population, agreeing with the literature. Patients among HIV+ populations are expected to develop that kind of morphologic diseases, both because of the effect of antiretroviral treatments, and for the inflammation that the virus itself causes. Presence/absence of disease was imbalanced, with an over-representation of "normal" cases; *class-imbalance* for lipodystrophy and low muscle mass was corrected by the SMOTE method, which lead to an improvement of the models' performance.

Male and female patients showed different rates of disease. For instance, male patients were observed to have higher rates of osteopenia and lipodystrophy, while female patients showed much higher rates of low muscle mass. Therefore, *dataset was finally split within male and female patients, and different models were built for each sex.* Sets were then split into two parts: a training set, used to fit the models, and a test set, with new observations used to validate the fitted model.

Since a big amount of features was originally available, with many of them being correlated to each other, *use of different dimensionality reduction methods was explored, with the aim of reducing dimensionality of the datasets and avoiding for multicollinearity.* Chosen techniques were PCA, MFA and clustering of variables, from which some synthetic variables were extracted, which were then used as "normal" predictors at the construction of classification models. Models built with the "best" syntehtic variables did not have much prediction capacity, so a fixed number of 15 variables was finally extracted and used at modelling, yielding better results.

Original and synthetic variables were used to build *logistic regression* and *machine learning models*. Models were fitted using training sets. Some smaller sets of original variables were also used (i.e. "basic" and "correlated" variables) in logistic regression models. The "backward" stepwise method of selection of variables was applied to the fitted models, creating simpler ones (in terms of number of variables) with a similar or even better performance than the original ones.

Machine learning algorithms explored included some of the most widely used learners found at the literature, such as C5.0 (decision tree) and support vector machines with a linear kernel. Some combinations of models' predictions, known as "ensemble models", was also explored. Ensembles explored were of *bagging, boosting and stacking* type, the last one consisting on combining the predictions of different types of "base" learners using a supervisor model. Since stacking functions were not available for multi-class cases (i.e. prediction of osteoporosis/osteopenia/normal), a *majority vote approach* was used, consisting of extracting the majority vote from the prediction of different trained models, comparing that "new"

vote (prediction) with the original output, in order to calculate the overall accuracy of the ensemble.

Best models were selected mainly based on their overall accuracy and AUC value, although sensitivity value and balance between sensitivity and specificity was also taken into account. For logistic regression models of similar accuracy and AUC values, those being simpler (in terms of number and kind of variables included) were preferred over more complicated ones. Since machine learning models are very hard to interpret, selection was based just on accuracy and AUC values, and not in the type of data used to build the model.

Results obtained from modelling varied within diseases.

- *Bone-quality related abnormalities* (i.e. osteoporosis/osteopenia) were hard to predict and use of Machine learning algorithms did not improve the results. The reason may be that bone abnormalities are affected by more variables than there were available for this study. For instance, data related to nutritional status, time since antiretroviral therapy started, diabetes, menopause, etc, have been found to be related to the presence of bone disease, but were not available. Also, class-imbalance may have negatively affected the results.
- Results obtained for *lipodystrophy and low muscle mass* were much better; machine learning models were able to improve results obtained by logistic regression, showing an "excellent" performance, with AUC values over 0.9. Algorithms leading to the best results were random forests combined with MFA variables (for female dataset) and stacking ensembles applied to original variables (for male dataset). In other words, it seems like DXA measurements carry enough information to the prediction of those diseases.

It was interesting to see how synthetic features from MFA analysis lead to very good results. Considering features as being structured into groups may somehow increase the pattern discovery of models, leading to better results.

Future works could include:

- Explore a method to balance classes of bone-disease, in order to improve model's performance.
- Repeat analysis in a new dataset and compare results. Ideally, more features should be considered (i.e. variables related to nutritional status, diabetes, presence of menopause, time since patient takes ART, time since patients has been infected by HIV), especially for the prediction of bone quality.
- Deeper investigate MFA and the best number of components, building models that include a different number of them. It has been shown how synthetic variables obtained by MFA got very good results used at machine learning ensembles. Adjusting the number of MFA variables may lead to even better performance results.
- Investigate package MLR (https://mlr.mlr-org.com/). Package MLR was also developed for the creation of ensemble learners and accepts multi-class methods.

5 Acronyms

ANN: artificial neural networks
ART: antiretroviral therapy
ASM: appendicular skeletal muscle mass
AUC: area under the (ROC) curve
BMD: bone mineral density
BMI: body mass index
DT: decision trees
DXA or DEXA: Dual-energy X-ray absorptiometry
FMR: fat mass ratio
HIV: human immunodeficiency virus
KNN: k-Nearest-Neighbors
LASSO: least absolute shrinkage and selection operator
MFA: multiple factor analysis
MMH: maximum margin hyperplane
ML: machine learning
PCA: principal component analysis
RF: random forests SVM: support vector machines

6 Glossary

Accuracy: proportion of correctly classified cases, calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP are *true positives* or cases of the considered as "positive" class (i.e. "presence of disease") correctly predicted; TN are *true negatives*, or cases of the "negative class" ("absence of disease) correctly predicted; FP represent *false positives* or cases incorrectly classified as being positive; and FN: *false negatives* are cases incorrectly classified as belonging to the"negative" class.

Artificial Neural Networks: algorithm that models the relationship between a set of input signals and a set of output signals, simulating the biological functioning of the brain.

Bootstrap: technique that consists on creating "new" sets of observations by resampling with replacement,

Ensemble learners: machine learning method that consists of combining several ML learners, of the same or different type, to improve model's performance. There are different ways of combining predictions of different learners, but one of the most typical one is using a "supervisor" model.

K-nearest-neighbours: machine learning algorithm that classifies data into categories by their similarities.

Lipodystrophy: body abnormality that consists of an abnormal fat-redistribution, typically found among HIV+ patients, with a typical fat-loss in face, buttocks, arms and legs.

Original or raw variables: anthropometric variables, as well as variables extracted from a DXA analysis, included in the original dataset under study.

Osteoporosis and osteopenia: morphologic disease consisting of a decrease in the bone quality, that increases risk of fractures, typically found among post-menopausal women and HIV+ patients.

Sensitivity or true positive rate: proportion of positive examples correctly classified, calculated as:

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity or true negative rate: proportion of negative examples correctly classified, calculated as:

$$Specificity = \frac{TN}{TN + FP}$$

Synthetic variables: variables created by applying dimensionality reduction methods to sets of original features. Different synthetic variables were be created to the prediction of each disease, since each disease under study used different sets of features.

7 Appendix I. Figures

			FEMALE PATIENTS			MALE PATIENTS		
BONE DISEASE		**type**	**Num**	**Acc.**	**AUC**	**Num**	**Acc.**	**AUC**
all feat	**log reg**	raw	30	62,71	0,78	30	56,52	0,75
		pca	15	52,54	0,70	15	53,53	0,70
		clust	15	57,63	0,75	15	57,88	0,75
		mfa	15	61,86	0,78	15	58,70	0,75
all feat	**log reg**	raw	20	62,71	0,80	17	58,97	0,74
	+ stepwise	pca	8	49,15	0,72	9	54,62	0,69
		clust	12	61,68	0,77	14	58,15	0,75
		mfa	10	61,02	0,79	14	57,34	0,75
basic feat	**log reg**	raw	10	63,56	0,77	10	50,27	0,71
	+ stepwise	raw	9	65,25	0,77	9	50,82	0,71
corr. Feat	**log reg**	raw	10	52,54	0,74	10	52,99	0,69
	+ stepwise	raw	8	52,54	0,74	8	52,72	0,69

Figure 31: Logistic regression models fitted for the prediction of bone abnormalities, for female and male patients. Finally slected models were chosen based on performance (accuracy, AUC) and simplicity; those results have been coloured in grey.

BONE DISEASE		FEMALE PATIENTS		MALE PATIENTS	
	type feat.	Acc.	AUC	Acc.	AUC
C5.0	raw	55,93	0,74	49,46	0,62
	pca	44,92	0,57	47,83	0,65
	clust	51,70	0,66	48,91	0,64
	mfa	55,09	0,74	53,26	0,68
Random forests	raw	52,54	0,74	49,73	0,63
	pca	56,78	0,71	51,63	0,66
	clust	51,70	0,69	48,10	0,63
	mfa	62,71	0,79	51,63	0,67
treebag	raw	53,85	0,72	45,11	0,60
	pca	45,76	0,63	50,27	0,64
	clust	51,70	0,69	48,10	0,63
	mfa	59,32	0,74	48,91	0,65
svmLinear	raw	55,93	0,73	53,53	0,72
	pca	55,93	0,69	53,53	0,68
	clust	55,93	0,73	52,45	0,70
	mfa	61,86	0,77	54,08	0,74
gbm	raw	54,24	0,73	52,17	0,67
	pca	45,76	0,66	50,27	0,64
	clust	54,24	0,73	51,90	0,67
	mfa	64,41	0,81	52,45	0,68
Ensemble *majority vote*	raw	54,24	0,74	50,82	0,67
	pca	50,85	0,67	51,90	0,66
	clust	54,24	0,74	50,00	0,67
	mfa	64,41	0,80	51,63	0,72

Figure 32: Machine learning models fitted for the prediction of osteoporosis/osteopenia, for male and female patients. Finally selected models have been coloured.

		FEMALE PATIENTS						MALE PATIENTS				
FAT DISEASE		type	Num	Acc.	Sens.	Spec.	AUC	Num	Acc.	Sens.	Spec.	AUC
all feat	**log reg**	raw	47	71,32	77,05	66,67	0,80	47	77,22	77,64	76,75	0,86
		pca	15	70,59	75,41	66,67	0,78	15	75,53	78,05	72,81	0,83
		clust	15	72,79	72,13	73,33	0,81	15	77,43	79,27	75,44	0,85
		mfa	15	69,12	80,33	0,60	0,75	15	76,58	79,68	73,25	0,83
all feat	**log reg**	raw	23	75,74	81,97	70,67	0,81	25	76,79	78,46	75,00	0,86
	+ stepwise	pca	12	69,12	72,13	66,67	0,79	13	75,74	79,27	71,93	0,83
		clust	9	71,32	70,49	72,00	0,81	12	78,06	80,49	75,44	0,85
		mfa	9	71,32	78,69	66,67	0,79	11	76,37	76,83	75,88	0,83
basic feat	**log reg**	raw	10	67,65	72,13	64,00	0,77	10	75,11	76,02	74,12	0,83
	+ stepwise	raw	7	66,18	72,13	61,33	0,76	9	74,68	75,61	73,68	0,83
corr. Feat	**log reg**	raw	10	62,50	70,49	56,00	0,74	10	74,05	76,02	71,93	0,83
	+ stepwise	raw	5	65,44	73,77	58,67	0,75	6	74,05	76,02	71,93	0,83

Figure 33: Logistic regression models fitted for the prediction of lipodystrophy, for female and male patients. Finally slected models were chosen based on performance (accuracy, AUC) and simplicity; those results have been coloured in grey.

FAT DISEASE		FEMALE PATIENTS				MALE PATIENTS			
	type feat.	Acc.	Sens.	Spec.	AUC	Acc.	Sens.	Spec.	AUC
C5.0	raw	86,77	88,53	85,33	0,92	86,92	89,02	84,65	0,94
	pca	83,09	73,77	90,67	0,94	85,02	87,40	82,46	0,94
	clust	71,32	68,85	73,33	0,75	85,44	86,59	84,21	0,93
	mfa	83,82	96,72	73,33	0,94	86,08	86,59	85,53	0,92
Random forests	raw	82,35	85,25	80,00	0,91	85,44	89,02	81,58	0,95
	pca	90,44	88,53	92,00	0,96	86,71	88,62	84,65	0,95
	clust	83,82	85,25	82,67	0,92	87,34	92,28	82,02	0,95
	mfa	89,71	96,72	84,00	0,97	86,50	86,99	85,97	0,95
treebag	raw	83,09	81,97	84,00	0,90	86,71	89,02	84,21	0,93
	pca	80,88	75,41	85,33	0,90	86,08	87,81	84,21	0,92
	clust	83,82	83,61	84,00	0,90	85,65	87,81	83,33	0,93
	mfa	89,71	95,08	85,33	0,96	84,39	84,96	83,77	0,90
svmLinear	raw	72,06	81,97	64,00	0,82	75,54	75,20	76,32	0,85
	pca	71,32	77,05	66,67	0,79	74,90	76,83	72,81	0,82
	clust	72,79	72,13	73,33	0,81	75,74	78,05	73,25	0,83
	mfa	69,12	81,97	58,67	0,77	76,16	79,27	72,81	0,82
gbm	raw	83,09	81,97	84,00	0,90	82,70	85,37	79,83	0,90
	pca	83,09	75,41	89,33	0,91	81,86	84,55	78,95	0,89
	clust	83,09	81,97	84,00	0,87	81,01	84,15	77,63	0,89
	mfa	83,09	88,53	78,67	0,90	81,22	80,89	81,58	0,89
Ensemble *supervisor*	raw	81,62	81,97	81,33	0,91	87,34	89,43	85,09	0,95
	pca	85,29	81,97	88,00	0,93	85,23	88,21	82,02	0,91
	clust	83,09	85,25	81,33	0,91	86,08	88,62	83,33	0,94
	mfa	81,62	91,80	73,33	0,95	85,23	86,99	83,33	0,93

Figure 34: Machine learning models fitted for the prediction of lipodystrophy, for male and female patients. Finally selected models have been coloured.

		FEMALE PATIENTS						MALE PATIENTS					
MUSCLE DISEASE		type	Num	Acc.	Sens.	Spec.	AUC	Num	Acc.	Sens.	Spec.	AUC
all feat	**log reg**	raw	53	88,79	91,84	86,21	0,95	53	93,68	95,80	91,90	0,99
		pca	15	81,31	83,67	79,31	0,91	15	89,46	89,92	89,09	0,96
		clust	15	77,57	80,61	75,00	0,85	15	90,81	92,44	89,44	0,97
		mfa	15	86,92	90,82	83,62	0,94	15	91,38	92,44	90,49	0,98
all feat	**log reg**	raw	29	86,45	87,76	85,35	0,95	44	94,06	94,96	93,31	0,99
	+ stepwise	pca	14	81,78	84,69	79,31	0,91	13	88,89	89,50	88,38	0,96
		clust	11	78,51	83,67	74,14	0,85	15	90,81	92,44	89,44	0,97
		mfa	11	85,98	89,80	82,76	0,94	15	91,57	92,02	91,20	0,98
basic feat	**log reg**	raw	10	86,92	90,82	83,62	0,95	10	91,95	95,38	89,09	0,98
	+ stepwise	raw	9	86,45	89,80	83,62	0,95	6	91,76	93,28	90,49	0,98
corr. Feat	**log reg**	raw	10	78,97	80,61	77,59	0,86	10	84,10	84,45	83,80	0,93
	+ stepwise	raw	9	78,04	79,59	76,72	0,86	7	84,29	85,71	83,10	0,93

Figure 35: Logistic regression models fitted for the prediction of low muscle mass, for female and male patients. Finally selected models were chosen based on performance (accuracy, AUC) and simplicity; those results have been coloured in grey.

MUSCLE DISEASE		FEMALE PATIENTS				MALE PATIENTS			
	type feat.	Acc.	Sens.	Spec.	AUC	Acc.	Sens.	Spec.	AUC
C5.0	raw	92,99	97,96	88,79	0,99	92,28	94,36	90,16	0,97
	pca	89,25	91,82	84,48	0,94	93,50	94,36	92,62	0,98
	clust	87,38	90,82	84,48	0,94	93,09	96,77	89,34	0,98
	mfa	91,12	94,90	87,93	0,98	91,06	93,55	88,53	0,97
Random forests	raw	89,72	92,86	87,07	0,98	94,31	95,97	92,62	0,97
	pca	91,12	91,84	90,52	0,98	91,87	92,74	90,98	0,98
	clust	87,38	88,78	86,21	0,96	90,65	91,94	89,34	0,96
	mfa	93,93	95,92	92,24	0,98	92,28	93,55	90,98	0,98
treebag	raw	88,32	94,90	82,76	0,97	90,65	91,94	89,34	0,97
	pca	87,85	91,84	84,48	0,94	88,21	90,32	86,07	0,95
	clust	85,51	87,76	83,62	0,95	88,62	89,52	87,71	0,94
	mfa	90,19	90,82	89,66	0,96	86,18	92,74	79,51	0,95
svmLinear	raw	89,72	89,80	89,66	0,95	93,90	97,58	90,16	0,99
	pca	82,71	82,65	82,76	0,90	91,46	95,16	87,71	0,96
	clust	80,84	84,69	77,59	0,86	91,06	94,36	87,71	0,98
	mfa	85,51	86,74	84,48	0,93	93,09	95,16	90,98	0,98
gbm	raw	92,52	93,88	91,38	0,98	93,50	95,97	90,98	0,98
	pca	88,79	94,90	83,62	0,96	92,68	96,77	88,53	0,97
	clust	87,85	91,84	84,48	0,92	90,65	95,16	86,07	0,97
	mfa	92,52	95,92	89,66	0,97	91,87	97,58	86,07	0,98
Ensemble *supervisor*	raw	93,46	97,96	89,66	0,99	96,17	98,74	94,01	1,00
	pca	90,65	92,86	88,79	0,98	92,53	92,86	92,25	0,98
	clust	88,79	92,86	85,35	0,95	93,10	94,54	91,90	0,99
	mfa	92,99	95,92	90,52	0,98	94,06	95,80	92,61	0,99

Figure 36: Machine learning models fitted for the prediction of lipodystrophy, for male and female patients. Finally selected models have been coloured.

8 Appendix II. Used R code

```
knitr::opts_chunk$set(echo = TRUE)

# Load/install necessary packages
load.libraries <- c("knitr", "foreign", "corrplot", "cluster",
                    "factoextra", "NbClust", "RColorBrewer", "ClustOfVar",
                    "FactoMineR", "caret", "nnet", "pROC", "MASS",
                    "DMwR","stats", "ROCR", "glmnet", "C50",
                    "randomForest","kernlab", "caretEnsemble")
install.lib <- load.libraries[!load.libraries %in% installed.packages()]
for(libs in install.lib) install.packages(libs, dependences = TRUE)
sapply(load.libraries, require, character = TRUE)
```

```r
# Preliminary check & clean of dataset
#-------------------------------------
#library(foreign)
# Import file
dexa.last <- read.spss(file.path(params$folder.data, params$data.last),
                       to.data.frame = TRUE)

# Eliminate unnecesary columns
dexa.last <- dexa.last[-c(3,6,9,12,15,18,21,24,27,68,73,76:78,80:82)]

# Eliminate patients with any NAs
dexa.nafree <- dexa.last
dexa.nafree <- na.omit(dexa.nafree)

# Eliminate patient with entry error
dexa.keep <- which(dexa.nafree$Weight > 0)
dexa.nafree <- dexa.nafree[dexa.keep,]
```

```r
# Establish presence of disease
#------------------------------
## 1. Bone disease

# Extract minimum T-value (from all sites) + add to dataset
v <- c("L1T", "L2T", "L3T", "L4T", "L1L4T", "L2L4T", "NeckFT", "WardsT",
       "TrochT", "TotalFT")
minT <- c()
for (i in 1:nrow(dexa.nafree)) {
  eachMin <- min(dexa.nafree[i,v])
  minT <- c(minT, eachMin)
}

dexa.nafree$minT_gral <- minT

## Extract minimum T-value - from hip + add to dataset
v.hip <- c("NeckFT", "WardsT", "TrochT", "TotalFT")
minT.hip <- c()
for (i in 1:nrow(dexa.nafree)) {
  eachMin <- min(dexa.nafree[i,v.hip])
  minT.hip <- c(minT.hip, eachMin)
}
dexa.nafree$minT_hip <- minT.hip

## Divide patients by T-scores in osteoporosis/osteopenia/normal
bone.diag <- c()
```

```r
for (i in 1:nrow(dexa.nafree)) {
  if (dexa.nafree$minT_gral[i] >= -1) {
    bone.diag <- c(bone.diag, "normal")
  } else {
    if ((dexa.nafree$minT_gral[i] < -1 ) & (dexa.nafree$minT_gral[i]
                                            > -2.5)) {
      bone.diag <- c(bone.diag, "osteopenia")
    } else {
      if (dexa.nafree$minT_gral[i] <= -2.5) {
        bone.diag <- c(bone.diag, "osteoporosis")
      }
    }
  }
}
# add to dataset - as factor
dexa.nafree$bone_diag <- as.factor(bone.diag)


# 2. Low muscle mass

dexa.nafree$lmm_diag <- NA
# Check for low muscle mass - female
dexa.nafree$lmm_diag[dexa.nafree$gender == "F" &
                       dexa.nafree$Apendicularleanmas < 6.0] <- "lmm"
dexa.nafree$lmm_diag[dexa.nafree$gender == "F" &
                       dexa.nafree$Apendicularleanmas >= 6.0] <- "normal"

# Check for low muscle mass - male
dexa.nafree$lmm_diag[dexa.nafree$gender == "M" &
                       dexa.nafree$Apendicularleanmas < 7.0] <- "lmm"
dexa.nafree$lmm_diag[dexa.nafree$gender == "M" &
                       dexa.nafree$Apendicularleanmas >= 7.0] <- "normal"

# transform column into factor
dexa.nafree$lmm_diag <- as.factor(dexa.nafree$lmm_diag)


# 3. Lipodystrophy

dexa.nafree$lipo_diag <- NA
# Check for low muscle mass - women
dexa.nafree$lipo_diag[dexa.nafree$gender == "F" &
                        dexa.nafree$FMR >= 1.329] <- "lipodystrophy"
dexa.nafree$lipo_diag[dexa.nafree$gender == "F" &
```

```
                        dexa.nafree$FMR < 1.329] <- "normal"

# Check for low muscle mass - men
dexa.nafree$lipo_diag[dexa.nafree$gender == "M" &
                        dexa.nafree$FMR >= 1.961] <- "lipodystrophy"
dexa.nafree$lipo_diag[dexa.nafree$gender == "M" &
                        dexa.nafree$FMR < 1.961] <- "normal"

# Transform into a factor
dexa.nafree$lipo_diag <- as.factor(dexa.nafree$lipo_diag)
```

```
# Rearrange dataset (order columns) + split into male/female
# ------------------------------------------------------------
#names(dexa.nafree)
dexa.ordered <- subset(dexa.nafree, select=c(1,68,69,70,2,64,65,63,4,51,52,3,21:50,66,67
        15,17,19,53,56:62,6,8,10,12,14,16,18,20,54,55))
dexa.clean <- dexa.ordered # we will work with "dexa.clean"

## Create male dataset
dexa.m <- dexa.clean[dexa.clean$gender == 'M',]
dexa.m$gender <- NULL

## Create female dataset
dexa.f <- dexa.clean[dexa.clean$gender == 'F',]
dexa.f$gender <- NULL
```

```
# Figure to study presence of disease - young vs old patients
#-------------------------------------------------------------
#library(RColorBrewer)
# women: 216 young, 138 old
# men: 699 young, 404 old

# 1. Osteoporosis/osteopenia
par(mfrow = c(1,3))
mydata <- data.frame(
Openia = c(
table(dexa.f$Age_cat == "<50" & dexa.f$bone_diag == "osteopenia")[2]/216,
table(dexa.f$Age_cat == ">=50" & dexa.f$bone_diag == "osteopenia")[2]/138,
table(dexa.m$Age_cat == "<50" & dexa.m$bone_diag == "osteopenia")[2]/699,
table(dexa.m$Age_cat == ">=50" & dexa.m$bone_diag == "osteopenia")[2]/404),
Oporosis = c(table(
dexa.f$Age_cat == "<50" & dexa.f$bone_diag == "osteoporosis")[2]/216,
table(dexa.f$Age_cat == ">=50" & dexa.f$bone_diag == "osteoporosis")[2]/138,
table(dexa.m$Age_cat == "<50" & dexa.m$bone_diag == "osteoporosis")[2]/699,
```

```
table(dexa.m$Age_cat == ">=50" & dexa.m$bone_diag == "osteoporosis")[2]/404))
# plot
barplot(as.matrix(mydata), col = brewer.pal(n = 4, name = "RdBu"),
        main = "Bone disease", beside = T, ylim = c(0,1))
legend(1,1, c("Women <50", "Women >=50", "Men <50", "Men >=50"),
       cex = 1, fill= brewer.pal(n = 4, name = "RdBu"))

# 2. Lipodystrophy
mydata2 <- data.frame(
Lipodystrophy = c(
table(dexa.f$Age_cat == "<50" & dexa.f$lipo_diag == "lipodystrophy")[2]/216,
table(dexa.f$Age_cat == ">=50" & dexa.f$lipo_diag == "lipodystrophy")[2]/138,
table(dexa.m$Age_cat == "<50" & dexa.m$lipo_diag == "lipodystrophy")[2]/699,
table(dexa.m$Age_cat == ">=50" & dexa.m$lipo_diag == "lipodystrophy")[2]/404))
# plot
barplot(as.matrix(mydata2), col =  brewer.pal(n = 4, name = "RdBu"),
        main = "Fat disease", beside = T, ylim = c(0,1))
legend(1,1, c("Women <50", "Women >=50", "Men <50", "Men >=50"), cex = 1,
       fill= brewer.pal(n = 4, name = "RdBu"))

# 3. Low muscle mass
mydata3 <- data.frame(
LMM = c(
table(dexa.f$Age_cat == "<50" & dexa.f$lmm_diag == "lmm")[2]/216,
table(dexa.f$Age_cat == ">=50" & dexa.f$lmm_diag == "lmm")[2]/138,
table(dexa.m$Age_cat == "<50" & dexa.m$lmm_diag == "lmm")[2]/699,
table(dexa.m$Age_cat == ">=50" & dexa.m$lmm_diag == "lmm")[2]/404))
# plot
barplot(as.matrix(mydata3), col =  brewer.pal(n = 4, name = "RdBu"),
        main = "Muscle disease", beside = T, ylim = c(0,1))
legend(1,1, c("Women <50", "Women >=50", "Men <50",  "Men >=50"),
       cex = 1, fill= brewer.pal(n = 4, name = "RdBu"))
```

```
# Exploratory analysis
#---------------------
## 1. Normality of some main variables
par(mfrow = c(2,2))
boxplot(dexa.clean$Age, main = 'age')
boxplot(dexa.clean$BMI, main = 'BMI')
boxplot(dexa.clean$TotalBMD, main = 'Total BMD')
boxplot(dexa.clean$minT_gral, main = 'min T-score')

## 2. Correlation study for a set of main variables
cor.set <- cor(dexa.clean[,c(8:13,16,19,22,31,34,37,40,43,47,50:60,63,66:70)])
```

```r
  # Extract correlations between variables
  row_indic <- apply(cor.set, 1, function(x) sum(x > 0.3 | x < -0.3) > 1)
  # correlation plot
  cor.set<- cor.set[row_indic ,row_indic ]
  corrplot(cor.set, method="square", tl.cex = 0.55, tl.col = "black")
```

```r
# Cluster of observations -> K-means + Silhouette
# -----------------------------------------------
## Extract continuous variables + scale
dexa.clust <- dexa.clean[,c(8:70)]
dexa.clust <- scale(dexa.clust)

# Apply method = "silhouette"
fviz_nbclust(dexa.clust, kmeans, method = "silhouette", k.max = 10)

# Study composition of clusters -> SPLIT WAS REJECTED
# kmeans.clust <- kmeans(dexa.clust, centers = 2)
# dexa.clean$cluster <- kmeans.clust$cluster
# summary(dexa.clean[dexa.clean$cluster == 1,])
# summary(dexa.clean[dexa.clean$cluster == 2,])
```

```r
# CUSTOMIZED FUNCTIONS TO SELECT DATA + EXTRACT RESULTS
# ----------------------------------------------------
```

```r
# 1. Function to select sex
# -------------------------
# Argument "sex" can get the values "female/male"

select_sex <- function(sex) {
  if (sex == "female") {
    dataset <- dexa.f
  } else if (sex == "male") {
      dataset <- dexa.m
  }
  return(dataset) # returns dataset of interest (dexa.f/dexa.m)
}
```

```r
# 2. Function to select disease, for a selected sex
# -------------------------------------------------
# Argument "sick": "bone/fat/muscle" (depends on disease under study)

select_disease <- function(sex, sick) {
  print(paste0("Selected sex is: ", sex))
```

```
  # 1. Select sex (female/male)
  dataset <- select_sex(sex)

  # Selection of features depending on disease to predict (not-related)
  # + extract labelName (diagnose)
  if (sick == "bone") {
    disease <- dataset[-c(1,3:6,11:43)]
    labelName <- "bone_diag"
  } else if (sick == "fat") {
    disease <- dataset[-c(1:3,5,6,44:59)]
    labelName <- "lipo_diag"
  } else if (sick == "muscle") {
    disease <- dataset[-c(1,2,4:6,60:69)]
    labelName <- "lmm_diag"
  }
  print(paste0("Selected disease is: ", sick))
  mylist <- list(disease, labelName)
  return(mylist) # returns disease dataset + label name (diagnosis)
}
```

```
# 3. Normalization function
# -------------------------
normalize <- function(x) {
      return((x - min(x)) / (max(x) - min(x)))
}
```

```
# 4. Function to select type of formula
# -------------------------------------
# Argument "type_formula": values "all" (all variables). For "raw" data, also
# "basic"/"corr" (basic or correlated variables) can be used

select_formula <- function(sex,sick, type_formula) {
  # 1. Select sex & disease
  res.disease <- select_disease(sex, sick)
  labelName <- res.disease[[2]]
  disease <- res.disease[[1]]
  predictors <- names(disease[-1])

    if (type_formula == "all") {
    formula <- formula(paste(names(disease[labelName]),
                paste(names(disease[,predictors]),
                      collapse = "+"), sep = "~"))
    } else {
      if (type_formula == "basic") {
```

```
    if (sick == "fat") {
      formula <- formula(lipo_diag ~ Age + Height + Weight +
                           BMI + TotalBMD + minT_gral + TotalLg
                         + FFMI + Apendicularleanmas)
  }
    else if (sick == "bone") {
          formula <- formula(bone_diag ~ Age + Height + Weight +
                               BMI + TotalFg + FMI + FMR + FFMI +
                             Indexdistributionfat +
                             Apendicularleanmas)
    } else if (sick == "muscle") {
    formula <- formula(lmm_diag ~ Age + Height + Weight + BMI +
                       TotalBMD + minT_gral + TotalFg + FMI +
                       FMR + Indexdistributionfat)
  }

  } else if (type_formula == "corr") {
    if (sex == "female") {
      if (sick == "fat") {
      formula <- formula(lipo_diag ~ FFMI + TLg + Age + TotalLg +
                           Apendicularleanmas + LLLg + BothLLg +
                           L1Z + RLLg + L1L4Z)
    }
    else if (sick == "bone") {
      formula <- formula(bone_diag ~ Age + RALg + BothALg +
                           LLLg + RLLg + BothLLg + TotalLg +
                           Apendicularleanmas + LALg + FFMI)
    }
    else if (sick == "muscle") {
      formula <- formula(lmm_diag ~ Weight + BMI + TrochBMD +
                           TotalFBMD + TotalFT + TrochT + TotalBMD
                         + NeckFBMD + NeckFT + LAFg)
    }
  }
  if (sex == "male") {
    if (sick == "fat") {
      formula <- formula(lipo_diag ~ Age + FFMI + WardsBMD +
                           WardsT + NeckFT + NeckFBMD + TLg +
                           minT_hip + minT_gral +
                           Apendicularleanmas)
    }
    else if (sick == "bone") {
      formula <- formula(bone_diag ~ BothALg + BothLLg + LALg +
                           RALg + RLLg + LLLg + TotalLg +
```

```
                              Apendicularleanmas + Age + TLg)
        }
        else if (sick == "muscle") {
          formula <- formula(lmm_diag ~ BMI + Weight + TotalBMD +
                              TrochBMD + TrochT + TotalFT + TotalFBMD +
                              NeckFT + NeckFBMD + minT_hip)
        }
      }
    }
  }
return(formula)
}
```

```
# 5. Function to select sex, disease, type of data and formula
# ------------------------------------------------------------
# This function creates sets of variables (original or synthetic) as
# well as the formula to use in further analysis
# Argument "data_type": "raw/pca/clust/mfa"

select_type_data <- function(sex, sick, data_type, type_formula = "all") {
  res.dis <- select_disease(sex, sick)
  labelName <- res.dis[[2]]
  disease <- res.dis[[1]] # 'disease' dataset contains features to
                          # predict disease of interest

  # Create 'pca.set' with numeric variables within 'disease'
  pca.set <-  disease[,!grepl("diag",names(disease))]
  pca.set <- pca.set[,!grepl("ID",names(pca.set))]

  # selection of original data
  if (data_type == "raw"){
    model.set <- disease
    predictors <- names(model.set)[names(model.set) != labelName]
    formula <- select_formula(sex, sick, type_formula)
    # selection of principal components
  } else if (data_type == "pca") {
      pca.res <- prcomp(pca.set, scale = T, center = T, rank. = 15)
      ## Normalization of data
      pca.norm <- lapply(as.data.frame(pca.res$x), normalize)
      pca.norm <- as.data.frame(pca.norm)
      model.set <- data.frame(disease[1], pca.norm)
      predictors <- names(model.set)[names(model.set) != labelName]
      formula <- formula(paste(names(model.set[labelName]),
                        paste(names(model.set[,predictors]),
```

```
                                collapse=" + "), sep=" ~ "))
      # selection of MFA dimensions (coordinates)
  } else if (data_type == "mfa") {
    if (sick == "fat") {
      res.mfa <- MFA(pca.set, group=c(1,3,33,10),
                     type=c(rep("c",4)), ncp=15,
                     name.group=c("age","antrop","bone","muscle"))
      model.set <- data.frame(disease[1], res.mfa$ind$coord)
      predictors <- names(model.set)[names(model.set) != labelName]
    } else if (sick == "bone") {
      res.mfa <- MFA(pca.set, group=c(1,3,16,10),
                     type=c(rep("c",4)), ncp=15,
                     name.group=c("age","antrop","fat","muscle"))
      model.set <- data.frame(disease[1], res.mfa$ind$coord)
      predictors <- names(model.set)[names(model.set) != labelName]
    } else if (sick == "muscle") {
      res.mfa <- MFA(pca.set, group=c(1,3,33,16),
                     type=c(rep("c",4)), ncp=15,
                     name.group=c("age","antrop","bone","fat"))
      model.set <- data.frame(disease[1], res.mfa$ind$coord)
      predictors <- names(model.set)[names(model.set) != labelName]
    }
formula <- formula(paste(names(model.set[labelName]),
                         paste(names(model.set[,predictors]),
                               collapse=" + "), sep=" ~ "))
    # extract clusters
  } else if (data_type == "clust") {
      hclust.res <- hclustvar(X.quanti = pca.set)
      clust.cut <- cutreevar(hclust.res, 15)
      model.set <- data.frame(disease[1], clust.cut$scores)
      predictors <- names(model.set)[names(model.set) != labelName]
     formula <- formula(paste(names(model.set[labelName]),
                         paste(names(model.set[,predictors]),
                               collapse=" + "), sep=" ~ "))
  }
res.list <- list(labelName, formula, predictors, model.set)
return(res.list)
}
```

```
# 6. Function to balance classes and split into train and test sets
# ------------------------------------------------------------------
# Imbalance of classes exists. SMOTE method will be applied
# Balance works well for "fat/muscle"; for "bone" disease, results
# are similar to original ones.
```

```r
# Formula "all" will be used in any case

balance_and_split <- function(sex, sick, data_type) {
  results <- select_type_data(sex, sick, data_type)
  # 1. Create formula & model.set
  labelName <- results[[1]]
  formula <- results[[2]]
  predictors <- results[[3]]
  model.set <- results[[4]]

  # 2. Balance data with SMOTE
  # ----- perc.over will probably depend on disease -----
  if (sick == "bone") { # do nothing
      balancedData <- model.set
  } else if (sick == "muscle") {
      if (sex == "male") {
        set.seed(123)
        balancedData <- SMOTE(formula, data= model.set,
                              perc.over = 300, perc.under = 150)
      } else { # sex = female
        set.seed(123)
        balancedData <- SMOTE(formula, data= model.set,
                              perc.over = 150)
      }
    } else { # sick = fat
        set.seed(123)
        balancedData <- SMOTE(formula, data= model.set,
                              perc.over = 150)
    }
  # 3. Split into training and test Sets
    set.seed(123)
    smp_size <- floor(2/3 * nrow(balancedData))

    set.seed(123) # split with index
    train_ind <- sample(seq_len(nrow(balancedData)), size = smp_size)
    trainSet <- balancedData[train_ind, ]
    testSet <- balancedData[-train_ind, ]

    my.split.list <- list(labelName, formula, predictors, model.set, trainSet,
                          testSet, balancedData)
return(my.split.list)
}
```

```r
# 7. Function to apply logistic regression to (all/selected) cases
# ----------------------------------------------------------------
log_results <- function(sex, sick, data_type, type_formula = "all") {
  # Create variables to use
  split <- balance_and_split(sex, sick, data_type)
  labelName <- split[[1]]
  predictors <- split[[3]]
  trainSet <- split[[5]]
  testSet <- split[[6]]
  results.selection <- select_type_data(sex, sick, data_type,
                                        type_formula)
  formula <- results.selection[[2]]

  # 2. Fit log models
  set.seed(123)
  log.mod <- multinom(formula, data = trainSet)

  # 3. Predict
  predLog <- predict(log.mod, newdata = testSet, type = "class")
  predProb <- predict(log.mod, newdata = testSet,
                           type = "prob")
  # 4. Extract accuracy
  c <- confusionMatrix(predLog, testSet[,labelName])
  if (sick == "bone") { # calculate AUC multicurve
    #library(pROC)
    multiRoc <- multiclass.roc(testSet[,labelName],
                               predProb, plot = F, percent = T)
    auc <- multiRoc$auc
    auc <- round(auc*0.01, 3)
  } else {
    auc2 <- roc(testSet[,labelName],predProb,
            smoothed = TRUE, plot=T, auc.polygon=T,
            max.auc.polygon=TRUE, grid = T,
            print.auc=T)
    auc <- round(auc2$auc,3)
  }
  my.res.list <- list(c, auc, log.mod, labelName, formula, predictors, trainSet,
                      testSet)
  return(my.res.list)
}
```

```r
# Loop to get results for all combinations of sex/disease/data type/ formula
# NOTE: formulas of type "basic" and "corr" can just be applied to "raw" data
sex.list <- c("female", "male")
```

```
disease.list <- c("bone", "fat", "muscle")
data.type.list <- c("pca", "clust", "mfa") #, "raw")
form.list <- c("all") # , "basic", "corr")
i <- 1
save.results <- data.frame()
final.table.log <- data.frame()
save.mods.log <- list()# save models

# loop to extract results for each case
for (sex in sex.list) {
  for (disease in disease.list) {
    for (data in data.type.list) {
      for (form in form.list) {
      log.results <- log_results(sex, disease, data, form)
      c <- log.results[[1]]
      auc <- log.results[[2]]
      acc <- round(c$overall[1]*100,3)
      sens <- round(c$byClass[1]*100, 3)
      spec <- round(c$byClass[2]*100, 3)

      # save model in list
      save.mods.log[[i]] <- log.results[[3]]
      i <- i + 1

      # save combination of sex, disease, data, formula
      combo <- paste0(sex, " ",disease, " ", data, " ", form)

      # save dataframe of results
      save.results <- cbind(combo, acc, sens, spec, auc)
      final.table.log <- rbind(final.table.log, save.results)
      }
    }
  }
}
#final.table.log
#str(save.mods.log)
```

```
# 8. Loop to apply stepwise to log models
#---------------------------------------
# NOTE: stepAIC() cannot be used inside a customized function

sex.list <- c("female", "male")
disease.list <- c("muscle", "bone", "fat")
data.type.list <- c("raw","clust", "pca","mfa")
```

```
form.list <- c("all") #, "corr", "basic")
save.mods.step <- list()
save.auc <- c()
combo <- c()
final.table.bc.step <- data.frame()
i <- 1

for (sex in sex.list) {
  for (disease in disease.list) {
      for (data in data.type.list) {
        for (form in form.list) {
        log.results <- log_results(sex, disease, data, form)
        labelName <- log.results[[4]]
        formula <- log.results[[5]]
        trainSet <- log.results[[7]]
        testSet <- log.results[[8]]

       # Extract log model + apply stepwise
        log.mod <- log.results[[3]]
        #library(MASS)
        set.seed(params$models.seed)
        step.mod <- stepAIC(log.mod, direction = "backward", trace = 0)

        # predictions
        predClass <- predict(step.mod, newdata = testSet)
        predProb <- predict(step.mod, newdata = testSet, type = "prob")
        c <- confusionMatrix(predClass, testSet[,labelName])
        acc <- round(c$overall[1]*100,3)
        sens <- round(c$byClass[1]*100, 3)
        spec <- round(c$byClass[2]*100, 3)

        # save model in list
        save.mods.step[[i]] <- step.mod
        i <- i + 1

        # extract number of features after stepwise
        num.coefs <- length(step.mod$coefnames)

        # print combination of sex, disease, data used, features (formula)
        combo <- paste0(sex, " ",disease, " ", data, " ", form)

        if (disease == "bone") { # calculate AUC multicurve
          multiRoc <- multiclass.roc(testSet[,labelName],
                                     predProb, plot = F, percent = T)
```

```
          auc <- multiRoc$auc
          auc <- round(auc*0.01, 3)

        } else {
            auc2 <- roc(testSet[,labelName],predProb,
            smoothed = TRUE, plot=T, auc.polygon=T,
            max.auc.polygon=TRUE, grid = T,
            print.auc=T) # ci would compute CI
            auc <- round(auc2$auc,3)
    }
      save.auc <- cbind(combo, num.coefs, acc, sens, spec, auc)
      final.table.bc.step <- rbind(final.table.bc.step, save.auc)
      }
    }
  }
}
names(final.table.bc.step) <- c("model", "num. features", "Accuracy",
                                "Sensitivity", "Specificity", "AUC")
# step.mod # step models are stored here
# final.table.bc.step # final results are stored here
```

```
# Extract features from selected models (created by stepwise)
#-------------------------------------------------------------
```

```
# 9. Function to apply LASSO to logistic models
# ----------------------------------------------
# NOTE: LASSO did not let extract AUC values or sets of variables, so it
# was finally not used
# lasso_results <- function(sex, disease, data, form)  {
#   set.seed(params$models.seed)
#       log.results <- log_results(sex, disease, data, form)
#       labelName <- log.results[[4]]
#       formula <- log.results[[5]]
#       predictors <- log.results[[6]]
#       trainSet <- log.results[[7]]
#       testSet <- log.results[[8]]
#
#       # 0. Create predictor matrix + extract response variable
#       X <- model.matrix(formula, data = trainSet)[,-1]
#       Xtest <- model.matrix(formula, data = testSet)[,-1]
#       y <- trainSet[,labelName]
#
#       ## Extract lambda for best fit by CV
#       set.seed(params$models.seed)
```

```
#         if (disease == "bone") {
#         family.type <- "multinomial"
#       } else {
#         family.type <- "binomial"
#       }
#       cv.lasso <- cv.glmnet(X, y, alpha = 1, type.measure = "mse",
#                             nfold = 10, family = family.type)
#       bestLam <- cv.lasso$lambda.min
#       ## Fit models
#       grid <- 10^seq(10, -2, length = 100)
#       set.seed(params$models.seed)
#       lasso.mod <- glmnet(X, y, alpha = 1, lambda = grid,
#                           type.multinomial = "grouped",
#                           family = family.type)
#       ## confusionmatrix
#       predClass <- predict(lasso.mod, newx= Xtest,
#                                   s = bestLam, type = "class")
#       predClass <- as.factor(predClass)
#       c <- confusionMatrix(predClass, testSet[,labelName])
#       my.res.list <- list(lasso.mod, c, l )
#   return (my.res.list)
# }
```

```
# 10. Function to apply ML models, using default tuning method
# ------------------------------------------------------------
ml_results <- function(sex, sick, data_type, type_formula = "all",
                       ml_method) {
  # Balance and split + Extract necessary information
  split <- balance_and_split(sex, sick, data_type)
  labelName <- split[[1]]
  predictors <- split[[3]]
  trainSet <- split[[5]]
  testSet <- split[[6]]

  # Set training parameters
  myControl <- trainControl(method="boot",
                            number=25,
                            classProbs = T)
  myMetric <- "ROC"
  preProc <- c("center", "scale")

  # Fit ML model/s. Parameters will depend on model
  if (ml_method == "svmLinear")  {
      set.seed(params$models.seed)
```

```
    ml.model <- train(x = trainSet[,predictors],
                y = trainSet[,labelName],
                method = ml_method,
                metric = myMetric,
                trControl = myControl,
                preProc = preProc)
} else {
  set.seed(params$models.seed)
    ml.model <- train(x = trainSet[,predictors],
                y = trainSet[,labelName],
                method = ml_method,
                metric = myMetric,
                trControl = myControl)
}
# Some models let us extract the "best tune"
#best.tune <- ml.model$bestTune

# Predict
ml.predClass <- predict(ml.model, testSet[,predictors])

# Performance
c <- confusionMatrix(ml.predClass, testSet[,labelName])

## auc -> different function for binary and multi-class
predProb <- predict(ml.model, newdata = testSet[,predictors], type = "prob")

  if (sick == "bone") {
    multiRoc <- multiclass.roc(testSet[,labelName],
                               predProb, plot = F, percent = T)
    auc <- multiRoc$auc
    auc <- round(auc*0.01, 3)
  } else {
   auc2.plot <- roc(testSet[,labelName],predProb[,1],
          smoothed = TRUE, plot=T, auc.polygon=T,
          max.auc.polygon=TRUE, grid = T,
          print.auc=T)
    auc <- round(auc2.plot$auc,3)
  }
# Extract + store results
acc <- round(c$overall[1]*100,3)
sens <- round(c$byClass[1]*100,3)
spec <- round(c$byClass[2]*100,3)

combo <- paste0(sex, " ",sick, " ", data_type, " ", ml_method)
```

```
  res.list.ml <- list(combo, acc, sens, spec, auc) #, best.tune)
  return(res.list.ml)
}
```

```
# Loop to extract results for ML models

sex.list <- c("female", "male")
disease.list <- c("bone","fat", "muscle")
data.list <- c("raw", "pca", "clust", "mfa")
form.list <- c("all")
method.list <- c("C5.0","rf","treebag", "svmLinear", "gbm")
res.vector <- data.frame()
res.ml.table <- data.frame()

for (sex in sex.list) {
  for (disease in disease.list) {
    for (data in data.list) {
      for (form in form.list) {
        for (method in method.list) {
          set.seed(params$models.seed)
          ml.res <- ml_results(sex, disease, data, form, method)

          # extract results - will depend on model
          #best.tune <- ml.res[[6]] # for C5.0, rf, gbm

      # Example for gbm
      res.vector <- cbind(ml.res[[1]], ml.res[[2]], ml.res[[3]], ml.res[[4]],
                          ml.res[[5]])
      res.ml.table <- rbind(res.ml.table, res.vector)
        }
      }
    }
  }
}

# extract results - without tuning parameters
names(res.ml.table) <- c("model", "Acc.", "Sens.", "Spec.", "AUC")
res.ml.table
```

```
# 11. Function for stacking models -> only lipodystrophy and muscle mass
# ------------------------------------------------------------------------
# NOTE: Function can just be applied to lipodystrophy and low muscle mass

stacking_models <- function(sex, sick, data_type, type_formula = "all") {
```

```
# Balance and split.
split <- balance_and_split(sex, sick, data_type)
labelName <- split[[1]]
predictors <- split[[3]]
trainSet <- split[[5]]
testSet <- split[[6]]

# Set training parameters
myControl <- trainControl(method="boot",
                          number=25,
                          classProbs = T)
stackControl <- trainControl(method="repeatedcv",
                          number=10, repeats = 3,
                          savePredictions = TRUE,
                          classProbs = T,
                          verboseIter = TRUE)
myMetric <- "ROC"

# Fit (list of) models
set.seed(params$models.seed)
model.list <- caretList(x = trainSet[,predictors],
                        y = trainSet[,labelName],
                        trControl = stackControl,
                        methodList = c("C5.0","nnet", "glm", "gbm",
                         "svmLinear"),
                        metric = myMetric)

# Extract results of training
set.seed(params$models.seed)
stack.res <- resamples(model.list)
models.summary <- summary(stack.res) # summary of models
models.corr <- modelCor(stack.res) # correlation - desired to be low

# Stack models using a supervisor (glm)
# Stacking needs a differnt type of control to the one used for fitting
set.seed(params$models.seed)
stack.mod <- caretStack(model.list, method="glmnet",
                        metric=myMetric,
                        trControl=myControl)

# Predictions + accuracy -> for ensemble with supervisor
predClass.s <- predict(stack.mod,
                           newdata = testSet[,predictors])
predProb.s <- predict(stack.mod,
```

```
                        newdata = testSet[,predictors], type = "prob")
  c.s <- confusionMatrix(predClass.s, testSet[,labelName])

  # Extract AUC value
  aucRoc.s <- roc(testSet[,labelName],predProb.s, smoothed = TRUE, plot=T,
                  auc.polygon=T, max.auc.polygon=TRUE, grid = T,
                  print.auc=T)
  auc.s <- round(aucRoc.s$auc, 3)

  # Extract + store results
  acc.s <- round(c.s$overall[1]*100,3)
  sens.s <- round(c.s$byClass[1]*100,3)
  spec.s <- round(c.s$byClass[2]*100,3)
  combo <- paste0(sex, " ",sick, " ", data_type)
  res.list.ensemble <- list(combo, models.summary, models.corr, acc.s,
                            sens.s, spec.s, auc.s)
  return(res.list.ensemble)
}
```

```
# Loop to extract results for stacking method (fat and muscle only)
sex.list <- c("female", "male")
disease.list <- c("fat", "muscle")
data.list <- c("mfa", "pca", "clust", "all")
model.summary <- list()
model.corr <- list()
stack.res <- data.frame()
table.stack <- data.frame()
i <- 1

for (sex in sex.list){
  for (sick in disease.list) {
    for (data in data.list) {
      stacking <- stacking_models(sex, sick, data)
      # 1. combo, 2. models. summary, 3. models corr, 4. acc stack,
      # 5. sens stack, 6. spec. stack, 7. auc stack
      model.summary[[i]] <- stacking[[2]] # # save list of model summaries
      model.corr[[i]] <- stacking[[3]] #  # save list of model correlations

      # get stacking results
      stack.res <- cbind(stacking[[1]], stacking[[4]], stacking[[5]],
                         stacking[[6]], stacking[[7]])
      # store in table
      table.stack <- rbind(table.stack, stack.res)
      i <- i + 1
```

```
    }
  }
}
# extract results - without tuning parameters
names(table.stack) <- c("model", "Acc.stack", "Sens.stack", "Spec.stack", "AUC stack")
#table.stack
```

```
# 12. Stacking models by majority vote -> alternative for bone disease
# ------------------------------------------------------------------------
# create list of models to use
mod.list <- list("C5.0", "treebag", "gbm", "rf", "svmLinear")
sex.list <- c("male") #, "female")
data.list <- c("all") #, "pca", "clust", "mfa")

myControl <- trainControl(method="boot",
                          number=25,
                          classProbs = T)
myMetric <- "ROC"
res.bone <- data.frame()
final.table.ensembl.bone <- data.frame()
pred.list <- list()
predProb.list <- list()
i <- 1

# 1. Fit different models (5)
for (sex in sex.list) {
  for (data in data.list) {
    split <- balance_and_split(sex, "bone", data)
    labelName <- split[[1]]
    predictors <- split[[3]]
    trainSet <- split[[5]]
    testSet <- split[[6]]

    for (sel.model in mod.list) {
      # 1. fit models + predict
      mod.fit <- train(y = trainSet[,labelName],
                       x = trainSet[,predictors],
                       trControl = myControl,
                       method = sel.model)
      # store predictions of each model (class)
      pred.list[[i]] <- predict(mod.fit, testSet[,predictors])
      predProb.list[[i]] <- predict(mod.fit, testSet[,predictors],
                                    type = "prob")
      i <- i + 1
```

```
}

# Extract majority vote
maj.vote <- cbind.data.frame(pred.list[[1]], pred.list[[2]],
                  pred.list[[3]],pred.list[[4]],
                 pred.list[[5]])
names(maj.vote) <- c("C5.0", "treebag", "gbm", "rf", "svmLinear")
maj.vote$maj.vote <- apply(maj.vote,1,function(x)
  X = names(which.max(table(x))))
maj.vote$maj.vote <- as.factor(maj.vote$maj.vote)
maj.vote$real.case <- testSet[,labelName]

# compare maj.vote with real output - confusion matrix
c.bone <- confusionMatrix(maj.vote$maj.vote, testSet[,labelName])
acc.ensemb.bone <- round(c.bone$overall[1]*100, 3)

# 5. Apply average probability-> to calculate AUC
all.probs <- data.frame(predProb.list[[1]], predProb.list[[2]],
                  predProb.list[[3]],predProb.list[[4]],
                  predProb.list[[5]])
## extract each class' probability
normal.prob <- data.frame(all.probs[1], all.probs[4], all.probs[7],
                          all.probs[10], all.probs[13])
osteopenia.prob <- data.frame(all.probs[2], all.probs[5],
                              all.probs[8],all.probs[11],
                              all.probs[14])
osteoporosis.prob <- data.frame(all.probs[3], all.probs[6],
                                all.probs[9],all.probs[12], all.probs[15])

# calculate average probability, for each cases and for all models
av.normal <- apply(normal.prob, 1, mean)
av.osteopenia <- apply(osteopenia.prob, 1, mean)
av.osteoporosis <- apply(osteoporosis.prob, 1, mean)
av.probs <- cbind(av.normal, av.osteopenia, av.osteoporosis)
colnames(av.probs) <- c("normal", "osteopenia", "osteoporosis")

# Calculate AUC with the extracted probabilities
multiRoc.stack <- multiclass.roc(testSet[,labelName],
                                 av.probs, plot = F, percent = T)
auc.bone.stack <- multiRoc.stack$auc

# Save final results
combo <- paste0(sex, " ","bone", " ", data)
res.bone <- cbind(combo, acc.ensemb.bone, auc.bone.stack)
```

```
    final.table.ensembl.bone <- rbind(final.table.ensembl.bone, res.bone)
  }}
# final.table.ensembl.bone
```

References

Beraldo, RA, Helena Siqueira Vassimon, Davi Casale Aragon, Anderson Marliere Navarro, FJ Albuquerque De Paula, and Maria Cristina Foss-Freitas. 2015. "Proposed Ratios and Cutoffs for the Assessment of Lipodystrophy in Hiv-Seropositive Individuals." *European Journal of Clinical Nutrition* 69 (2). Nature Publishing Group: 274.

Bolland, Mark J, Andrew B Grey, Greg D Gamble, and Ian R Reid. 2007. "Low Body Weight Mediates the Relationship Between Hiv Infection and Low Bone Mineral Density: A Meta-Analysis." *The Journal of Clinical Endocrinology & Metabolism* 92 (12). Oxford University Press: 4522–8.

Bonnet, E, C Delpierre, A Sommet, F Marion-Latard, R Herve, C Aquilina, E Labau, et al. 2005. "Total Body Composition by Dxa of 241 Hiv-Negative Men and 162 Hiv-Infected Men: Proposal of Reference Values for Defining Lipodystrophy." *Journal of Clinical Densitometry* 8 (3). Elsevier: 287–92.

Carr, Andrew, Birgit Grund, Jacqueline Neuhaus, Ann Schwartz, Jose I Bernardino, David White, Sharlaa Badel-Faesen, et al. 2015. "Prevalence of and Risk Factors for Low Bone Mineral Density in Untreated Hiv Infection: A Substudy of the Insight Strategic Timing of Antiretroviral Treatment (Start) Trial." *HIV Medicine* 16. Wiley Online Library: 137–46.

Chavent, Marie, Robin Genuer, and Jerome Saracco. 2016. "Combining Clustering of Variables and Feature Selection Using Random Forests." *arXiv Preprint arXiv:1608.06740*.

Chavent, Marie, Vanessa Kuentz, Beno^it Liquet, and L Saracco. 2011. "ClustOfVar: An R Package for the Clustering of Variables." *arXiv Preprint arXiv:1112.0295*.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. "SMOTE: Synthetic Minority over-Sampling Technique." *Journal of Artificial Intelligence Research* 16: 321–57.

Chiquette, Elaine, Elif A Oral, Abhimanyu Garg, David Araújo-Vilar, and Praveen Dhankhar. 2017. "Estimating the Prevalence of Generalized and Partial Lipodystrophy: Findings and Challenges." *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy* 10. Dove Press: 375.

Cruz-Jentoft, Alfonso J, Gülistan Bahat, Jürgen Bauer, Yves Boirie, Olivier Bruyère, Tommy Cederholm, Cyrus Cooper, et al. 2018. "Sarcopenia: Revised European Consensus on Definition and Diagnosis." *Age and Ageing* 48 (1). Oxford University Press: 16–31.

Deane-Mayer, Zachary A, and JE Knowles. 2016. "CaretEnsemble: Ensembles of Caret Models." *R Package Version* 2 (0).

Dormann, Carsten F, Jane Elith, Sven Bacher, Carsten Buchmann, Gudrun Carl, Gabriel

Carré, Jaime R García Marquéz, et al. 2013. "Collinearity: A Review of Methods to Deal with It and a Simulation Study Evaluating Their Performance." *Ecography* 36 (1). Wiley Online Library: 27–46.

Freitas, Paula, Ana Cristina Santos, Davide Carvalho, Jorge Pereira, Rui Marques, Esteban Martinez, António Sarmento, and José Luís Medina. 2010. "Fat Mass Ratio: An Objective Tool to Define Lipodystrophy in Hiv-Infected Patients Under Antiretroviral Therapy." *Journal of Clinical Densitometry* 13 (2). Elsevier: 197–203.

Hain, Sharon F. 2006. "DXA Scanning for Osteoporosis." *Clinical Medicine* 6 (3). Royal College of Physicians: 254–58.

Hand, David J, and Robert J Till. 2001. "A Simple Generalisation of the Area Under the Roc Curve for Multiple Class Classification Problems." *Machine Learning* 45 (2). Springer: 171–86.

Hastie, Trevor, and Junyang Qian. 2014. "Glmnet Vignette." *Retrieve from Http://Www. Web. Stanford. Edu/~ Hastie/Papers/Glmnet_ Vignette. Pdf. Accessed September* 20: 2016.

Ioannidis, John PA, Thomas A Trikalinos, Matthew Law, Andrew Carr, and others. 2003. "HIV Lipodystrophy Case Definition Using Artificial Neural Network Modelling." *Antiviral Therapy* 8 (5). MTM PUBLICATIONS: 435–42.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

Kassambara, Alboukadel, and Fabian Mundt. 2017. "Package 'Factoextra'." *R Topics Documented* 75.

Kendler, David L, Joao LC Borges, Roger A Fielding, Akira Itabashi, Diane Krueger, Kathleen Mulligan, Bruno M Camargos, et al. 2013. "The Official Positions of the International Society for Clinical Densitometry: Indications of Use and Reporting of Dxa for Body Composition." *Journal of Clinical Densitometry* 16 (4). Elsevier: 496–507.

Kiang, Melody Y. 2003. "A Comparative Assessment of Classification Methods." *Decision Support Systems* 35 (4). Elsevier: 441–54.

Kilic, Niyazi, and Erkan Hosgormez. 2016. "Automatic Estimation of Osteoporotic Fracture Cases by Using Ensemble Learning Approaches." *Journal of Medical Systems* 40 (3). Springer: 61.

Kuhn, Max, and others. 2008. "Building Predictive Models in R Using the Caret Package." *Journal of Statistical Software* 28 (5): 1–26.

Lantz, Brett. 2015. *Machine Learning with R*. Packt Publishing Ltd.

Lee, Cathy Nga Yan, Simon Ching Lam, Alan Yat Kwan Tsang, Bernadette Ting Yan Ng, Joyce Chung Yin Leung, and Andy Chun Yin Chong. 2015. "Preliminary Investigation on Prevalence of Osteoporosis and Osteopenia: Should We Tune Our Focus on Healthy Adults?" *Japan Journal of Nursing Science* 12 (3). Wiley Online Library: 232–48.

Lê, Sébastien, Julie Josse, François Husson, and others. 2008. "FactoMineR: An R Package

for Multivariate Analysis." *Journal of Statistical Software* 25 (1). Los Angeles: 1–18.

Lima, Ana Lucia Lei Munhoz, Priscila Rosalba D de Oliveira, Perola Grimberg Plapler, Flora Maria D Andrea Marcolino, Eduardo de Souza Meirelles, André Sugawara, Riccardo Gomes Gobbi, Alexandre Leme Godoy dos Santos, and Gilberto Luis Camanho. 2011. "Osteopenia and Osteoporosis in People Living with Hiv: Multiprofessional Approach." *HIV/AIDS (Auckland, NZ)* 3. Dove Press: 117.

McClung, Michael R. 2003. "Prevention and Management of Osteoporosis." *Best Practice & Research Clinical Endocrinology & Metabolism* 17 (1). Elsevier: 53–71.

Montessori, Valentina, Natasha Press, Marianne Harris, Linda Akagi, and Julio SG Montaner. 2004. "Adverse Effects of Antiretroviral Therapy for Hiv Infection." *Cmaj* 170 (2). Can Med Assoc: 229–38.

Nasi, Milena, Sara De Biasi, Lara Gibellini, Elena Bianchini, S Pecorini, V Bacca, Giovanni Guaraldi, Cristina Mussini, Marcello Pinti, and Andrea Cossarizza. 2017. "Ageing and Inflammation in Patients with Hiv Infection." *Clinical & Experimental Immunology* 187 (1). Wiley Online Library: 44–52.

Neto, Pinto, Lauro Ferreira da Silva, Marina Cerqueira Sales, Eduarda Sobral Scaramussa, Clara Junia Calazans da Paz, and Renato Lirio Morelato. 2016. "Human Immunodeficiency Virus Infection and Its Association with Sarcopenia." *Brazilian Journal of Infectious Diseases* 20 (1). SciELO Brasil: 99–102.

Pages, Jérôme. 2004. "Multiple Factor Analysis: Main Features and Application to Sensory Data." *Revista Colombiana de Estadística* 27 (1). Universidad Nacional de Colombia: 1.

Pinnetti, Carmela, Lupi Federico, Patrizia Lorenzini, Chiappetta Domenico, Bellagamba Rita, Loiacono Laura, Mauro Zaccarelli, et al. 2014. "Relationship Between Body Mass Index and Bone Mineral Density in Hiv-Infected Patients Referred for Dxa." *Journal of the International AIDS Society* 17. Wiley Online Library: 19569.

Powderly, William G. 2012. "Osteoporosis and Bone Health in Hiv." *Current HIV/AIDS Reports* 9 (3). Springer: 218–22.

R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Ripley, Brian, William Venables, and Maintainer Brian Ripley. 2016. "Package 'Nnet'." *R Package Version* 7: 3–12.

Robin, Xavier, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. 2011. "PROC: An Open-Source Package for R and S+ to Analyze and Compare Roc Curves." *BMC Bioinformatics* 12 (1). BioMed Central: 77.

Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20. Elsevier: 53–65.

Santos, André P dos, Anderson M Navarro, Andiara Schwingel, Thiago C Alves, Pedro P Abdalla, Ana Claudia R Venturini, Rodrigo C de Santana, and Dalmo RL Machado. 2018.

"Lipodystrophy Diagnosis in People Living with Hiv/Aids: Prediction and Validation of Sex-Specific Anthropometric Models." *BMC Public Health* 18 (1). BioMed Central: 806.

Sarkar, D., and V. Natarajan. 2019. *Ensemble Machine Learning Cookbook: Over 35 Practical Recipes to Explore Ensemble Machine Learning Techniques Using Python*. Packt Publishing. https://books.google.at/books?id=dCWGDwAAQBAJ.

Schtscherbyna, Annie, Maria Fernanda Miguens Castelar Pinheiro, Laura Maria Carvalho de Mendonça, Carla Gouveia, Ronir Raggio Luiz, Elizabeth Stankiewicz Machado, and Maria Lucia Fleiuss de Farias. 2012. "Factors Associated with Low Bone Mineral Density in a Brazilian Cohort of Vertically Hiv-Infected Adolescents." *International Journal of Infectious Diseases* 16 (12). Elsevier: e872–e878.

Schubach, Max, Matteo Re, Peter N Robinson, and Giorgio Valentini. 2017. "Imbalance-Aware Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants." *Scientific Reports* 7 (1). Nature Publishing Group: 2959.

Shafiee, Gita, Abbasali Keshtkar, Akbar Soltani, Zeinab Ahadi, Bagher Larijani, and Ramin Heshmat. 2017. "Prevalence of Sarcopenia in the World: A Systematic Review and Meta-Analysis of General Population Studies." *Journal of Diabetes & Metabolic Disorders* 16 (1). BioMed Central: 21.

Tarca, Adi L, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. 2007. "Machine Learning and Its Applications to Biology." *PLoS Computational Biology* 3 (6). Public Library of Science: e116.

Tay, Wei-Liang, Chee-Kong Chui, Sim-Heng Ong, and Alvin Choong-Meng Ng. 2013. "Ensemble-Based Regression Analysis of Multimodal Medical Data for Osteopenia Diagnosis." *Expert Systems with Applications* 40 (2). Elsevier: 811–19.

Tian, Limin, Ruifei Yang, Lianhua Wei, Jing Liu, Yan Yang, Feifei Shao, Wenjuan Ma, Tingting Li, Yu Wang, and Tiankang Guo. 2017. "Prevalence of Osteoporosis and Related Lifestyle and Metabolic Factors of Postmenopausal Women and Elderly Men: A Cross-Sectional Study in Gansu Province, Northwestern of China." *Medicine* 96 (43). Wolters Kluwer Health.

Torgo, Luis, and Maintainer Luis Torgo. 2013. "Package 'Dmwr'." *Comprehensive R Archive Network*.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Fourth. New York: Springer. http://www.stats.ox.ac.uk/pub/MASS4.

Von Luxburg, Ulrike, and others. 2010. "Clustering Stability: An Overview." *Foundations and Trends in Machine Learning* 2 (3). Now Publishers, Inc.: 235–74.

Yoo, Tae Keun, Sung Kean Kim, Deok Won Kim, Joon Yul Choi, Wan Hyung Lee, and Eun-Cheol Park. 2013. "Osteoporosis Risk Prediction for Bone Mineral Density Assessment of Postmenopausal Women Using Machine Learning." *Yonsei Medical Journal* 54 (6): 1321–30.