Component 3

For tackling the problem of predicting the body fat percentage of an individual, I have used 2 well known, basic ML techniques: a linear regression and a decision tree.

**Linear regression** is an algorithm that predicts the output corresponding to a given input by making a weighted sum of the given parameters. It assigns weights to each input parameter and adjusts these values to minimize the sum of the squared differences between the predicted values and the ground truth. The major disadvantage of this method is that it is only suitable for linear hypothesis and does not accommodate other kinds of hypotheses. In case we have such a linear hypothesis, this approach is resilient to outlier data as it will adjust the weights to minimize the difference according to most of the data, not of exceptions.

**Decision tree** is another algorithm that can be used for regression tasks, but instead of assigning weights to the given input, it tries to make decisions by narrowing down the search space by querying input parameters based on standard deviation and coefficient of deviation. It goes down with branching until a desired standard deviation is reached or when the dataset can no longer be split. A decision tre can be seen as piecewise constant approximations: not smooth and not continuous. The advantage of this approach is that it computes the prediction in logarithmic time and it’s also easy to understand as it can be visualized. Here we can encounter the problem of overfitting in case we develop a tree that is too deep, but also underfit the data in case the tree is too shallow. Moreover, outliers can highly influence the process of creating such a decision tree as its creation is done by metrics like standard deviation which can create unwanted branches.

For training the models, I have used a dataset of 231 entries that has been split into 90-10% for training-testing. After training, the scoring of the models are:

* 0.9967503902441642 for the linear regression
* 0.998887380671577 for the decision tree

This scoring represents the coefficient of determination which is the R^2 score defined as (1 – u/v) where u = the residual sum of squares ((y\_true - y\_pred)\*\* 2).sum() and v = total sum of squares ((y\_true - y\_true.mean()) \*\* 2).sum().

These numbers are incredibly good metrics for how simple the models are, but due to the reduced size of the dataset, it is understandable.

After performing cross validation of the 2 methods with the k-fold method where k = 4, due to the reduced size of the data asset we got:

0.9584 accuracy with a standard deviation of 0.0523 for both models