CLUES on the Dynamics of the Local Group using Machine Learning

Edoardo Carlesi

September 9, 2020

1 Introduction

In this work we used several different *supervised machine learning algorithms* to tackle different issues related to the dynamics of the Local Group. Machine learning is a general term used to describe a host of algorithms that use large datasets to learn how to predict the value of a variable from a set of input values (in ML terminology, these are commonly referred to as *features*). The value can be discrete (in which case we are dealing with a *classification* problem) or a continuous one (a *regression* problem). We talk about *supervised learning* when the algorithm is trained using a dataset that also contains the desired output value, which is used to adjust its parameters. The goodness of the fit needs then to be gauged using a separate test dataset. In the case of *unsupervised learning* methods, on the other hand, there is no correct output value being provided. These methods are mostly used for classification and try to identify the correct metric that allows to separate the data into different homogeneous subclasses.

In this work the subject will be placed on the Local Group Dynamics, trying to infer - using different regression algorithms - the values of properties such as mass, mass ratio and tangential velocity using other variables (e.g. distance and radial velocity) as an input. The algorithms will be trained on different sets of LG-like objects identified in DM only simulations.

2 Machine Learning Methods

In this section we will introduce the main features of the different ML algorithms used in this work. We want to address the following questions:

- Can ML help us shedding some light on the known unknowns of the LG?
- Are ML models predictions comparable to those obtained with other methods?
- What is the best ML algorithm to address this kind of issues?

The benchmark method we will compare our results to is the *Timing Argument* (TA) estimate of the mass of the local group. We expect this estimate to be larger than the one obtained with ML methods, where $M_{\rm LG}$ is defined as the sum of $M_{\rm MW}$ and $M_{\rm M31}$ whereas the TA estimates the *total* bound mass of the system, that includes smaller galaxies that belong to the Local Group. In the case of the estimation of other parameters for the LG (v_{tan} and the mass ratio r_M), however, we will have no independent benchmark method to gauge the goodness of our results.
There are two main sets of free coefficients that need to be determined for an ML algorithm to function properly, which are referred to as the model *parameters* and *hyperparameters*. The

model's parameters are those that need to be fitted to the data, whereas the hyperparameters are those values that need to be tuned to specify the architecture of the algorithm (e.g. the number of decision trees in a random forest, the number of leaves in a decision tree and so on).
Given a dataset, we need to perform a train-test split, that is, separate the data into a first subsample (a fraction of $0.7 - 0.8$ of the total) that will be used to train (fit) the algorithm's parameters, whereas the remaining part of the data will be used to test the results and gauge the goodness of the fit.
The training operation itself has some hyperparameters that need to be tuned (e.g. the *learning rate* in the case of Stochastic Gradient Descent optimization); also, given that some fitting methods have a random component (e.g. bootstrapping with a Random Forest) they will not always produce the exact same results.

2.1 Linear Regression

Linear regression is a simple kind of approach where the value of a dependent variable is being put into linear relation to one (or more) other variables:

$$y = a_0 + x_1 * a_1 + ... + x_n * a_n$$

where the $x_1 ... x_n$ are the n dependent variables (features); the data is used to estimate the linear coefficients a_i that correlate the input to the output quantities. In order to apply linear regression to our data we need to make sure that the input variables are suitably rescaled, so that in the case of $M_{\rm LG}$ we will be using $\log_{10} M_{\rm LG}$ instead. In this work we'll mainly use LR as a benchmark to compare how different ML algorithms perform compared to the simplest possible approach.

2.2 Decision Tree

Decision trees are simple machine learning algorithms generally used for classification purposes, as they usually provide a discrete set of outcomes. In the case of regression, however, we can increase the number of final leaves and approximate a continuous variable with a large number output bins. Each node of the tree contains a condition on one (or more) feature and splits the set accordingly, and the thresholds that regulate these splits are tuned during the training procedure.

2.3 Random Forest

Random forest methods rely on decision trees and can be used for both classification and regression. They are a kind of *ensemble* method, that is, a ML method that does not rely on a single estimator but averages over several to provide a more robust prediction of the output variable. The basic idea is to combine a set of decision trees, with different properties (such as number of leaves) to produce a set of different outcomes for the final variable. In the case of classification, the value of the output is assigned by majority rule voting, that is, to the class chosen by the largest number of trees.

In the case of regression, on the other hand, the final value is obtained averaging all the different outputs of each tree. The number of decision trees and the maximum depth of the tree (that is, the maximum number of layers of nodes) are probably the two most important hyperparameters, with the largest impact on the final result. Moreover, during the training procedure we can opt for a bootstrapping approach, where different trees are independently fitted to different subset of the training data. While on the one hand this procedure reduces the overfitting of the training data, it may lead to different results even with the same hyperparameters of the model.

2.4 Gradient Boosting

Gradient boosting is another ensemble method where the number of estimators, however, is not chosen in advance as an hyperparameter, but rather adjusted dynamically at each training step in order to minimize the residuals (the difference between the predicted and the real value). When using decision trees as estimators, this resembles a random forest, with the exception that all the estimators are trained on the same data at the same time.

3 The Datasets

In this work we have tested our methods on three different samples of simulated LGs, the simulation datasets are:

- Dataset-1 (LGF): 1000 simulations of a $100h^{-1}$Mpc box (512^3 particles in the high-res region)
- Dataset-2 (SmallBox): 5 simulations in a $100h^{-1}$Mpc box (1024^3 particles)
- Dataset-3 (BigMD): $2.5h^{-1}$Gpc box

In Dataset-1 LG-like pairs have been searched for within a sphere of $15h^{-1}$Mpc from the box center. We have performed a first broad selection of LG pairs based on the following LG model:

- Halo masses within $(0.5 - -4.0) \times 10^{12}h^{-1}\mathrm{M}_{\odot}$
- Mass ratio M_{M31} to $M_{\mathrm{MW}} < 4$
- Separation $(0.5 - 1.3)h^{-1}$Mpc
- Isolation, i.e. no other halo of mass $\geq M_{\mathrm{MW}}$ within $2h^{-1}$Mpc
- $v_{rad}< 0$ km s^{-1}

Using this model, in Dataset-1 and Dataset-2 we find $\approx$ 2000 LG pairs, whereas in Dataset-3 the total number is $\approx 2x10^6$. In Fig. 3 we show the distributions of total masses (in $\log_{10}$), separation between the two main halos (R) and normalized v_{tan} and v_{rad}, that are defined as

$$v_{\mathrm{rad}}{}^{norm} = \log_{10}\left(-v_{\mathrm{rad}}/100.0\right); \quad v_{\mathrm{tan}}{}^{norm} = \log_{10}\left(v_{\mathrm{tan}}/100.0\right)$$

To end with, in Fig. 3 we plot the distributions for the masses of MW and M31 in the LG pairs identified in the three simulations, the median values and their 20/80 percentiles are shown in Table 3. We notice the strong influence of the M_{MW} mass threshold $> 0.5 \times 10^{12}h^{-1}\mathrm{M}_{\odot}$ and the condition $M_{\mathrm{M31}}> M_{\mathrm{MW}}$. However, the three distributions are very similar in terms of mean and scatter and we do not notice any systematic effect connected to any of the Dataset.

We can see that there is a substantial overlap of all the distribution among the three datasets, meaning that we will not expect the three datasets to produce different results for the ML algorithms despite the differences in particle resolution and sample size.

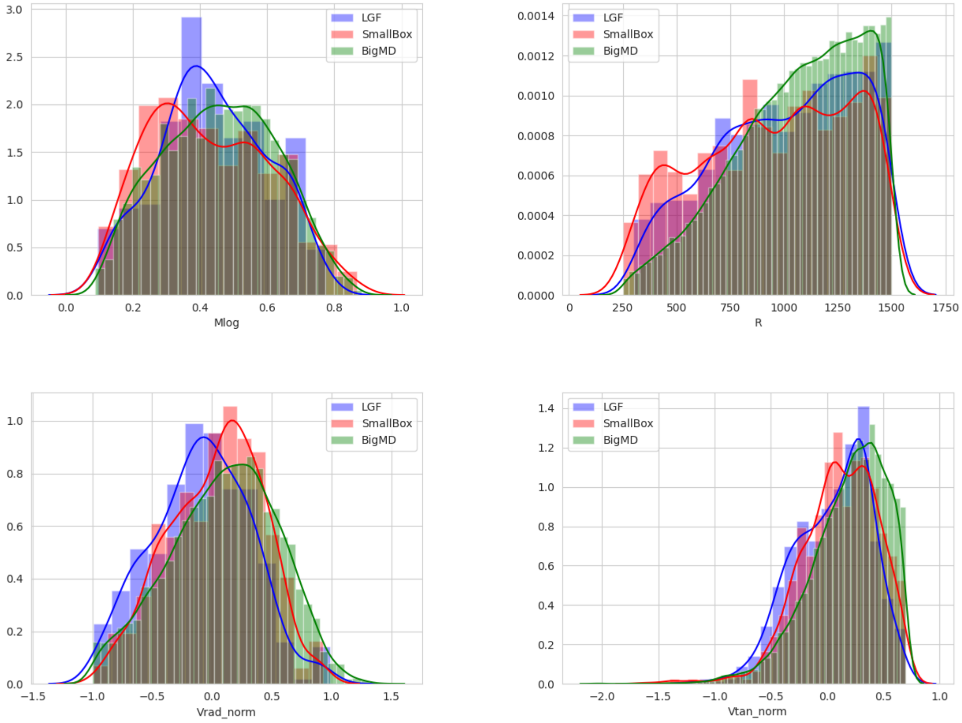

Figure 1: Distribution of total masses, distances, radial and tangential velocities for all LG-like objects within the three simulation samples. Total LG masses are divided by 10^{12} and then we take the $\log_{10}$, for radial and tangential velocities we divide by 100 km $s^{-1}$and then take the $\log_{10}$ (multiplying by -1 in the case of v_{rad}).

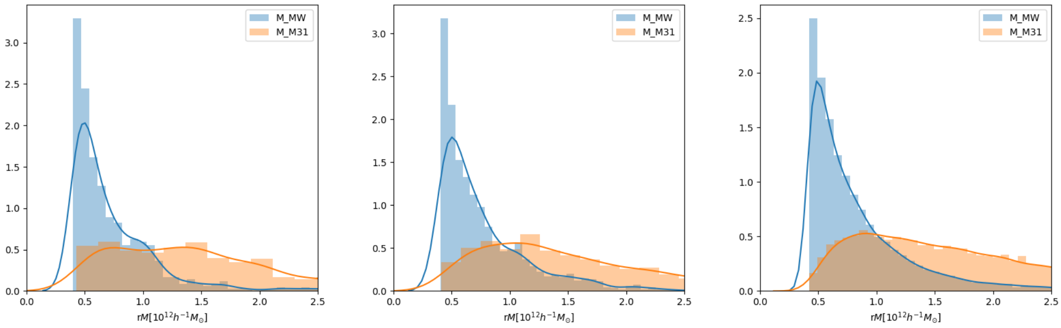

Table 1: Median masses of MW and M31 with their 20th and 80th percentiles, in 10^{12} $h^{-1}\mathrm{M}_{\odot}$ units.

Dataset	M_{MW}	M_{M31}
DS1	$0.597^{+0.352}_{-0.137}$	$1.358^{+0.682}_{-0.612}$
DS2	$0.642^{+0.405}_{-0.174}$	$1.394^{+0.936}_{-0.542}$
DS3	$0.682^{+0.423}_{-0.188}$	$1.552^{+0.941}_{-0.635}$

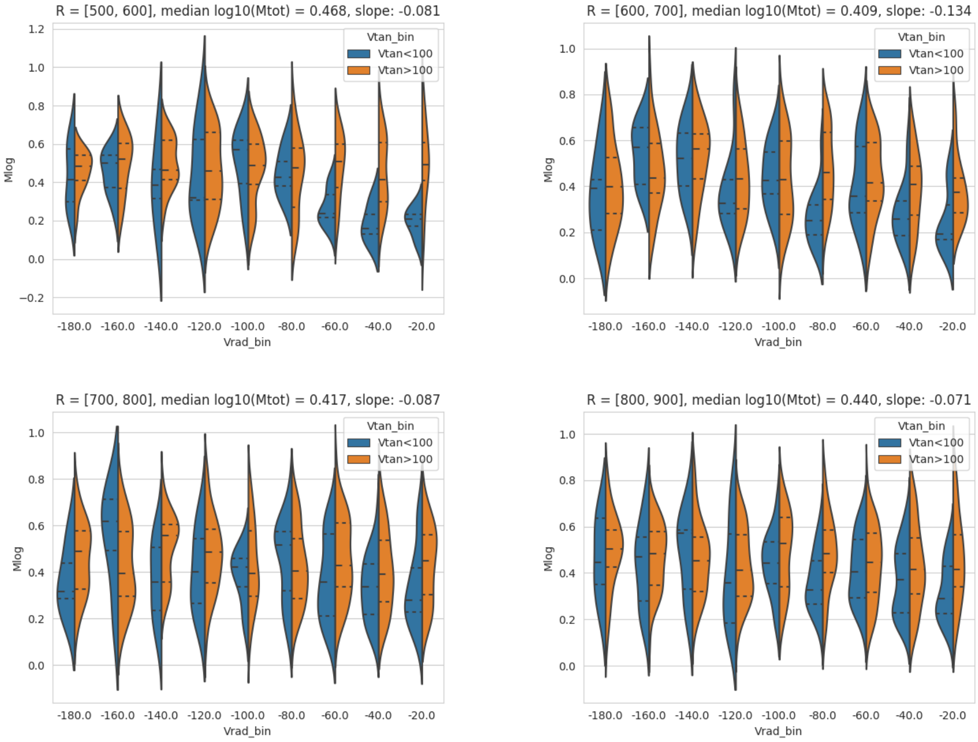

Figure 2: Violin plots of mass distributions at three different radial bins. These are obtained as follows: (1) we select ALL LG pairs that fall within a given radial bin (2) these pairs are binned according to their radial velocity, from -200 to 0 km s^{-1} (3) each bin is again divided into high tangential velocity pairs (> 100 km s^{-1}, orange area under the curve) and low tangential velocity pairs (< 100 km s^{-1}, blue area under the curve). Then we plot the $\log_{10}$ of the total mass of the two halos divided by $10^{12}h^{-1}\mathrm{M}_{\odot}$ at each v_{rad} bin. On top of each distribution the 25th and 75th percentiles are marked with dotted lines whereas the solid lines are the median values. These plots are obtained using Dataset-3.

3.1 Preliminary Analysis

To have a first understanding of how the features are correlated to the output value, we perform the analysis shown in Fig. 3.1, using Dataset-3. First, we group our halo sample by inter-halo distance R into four different sub-samples. Each sub-sample is then binned in v_{rad} and for each v_{rad} bin we plot the distribution of $M_{\rm LG}$. To further highlight any possible dependence on the v_{tan} variable, the $M_{\rm LG}$ distributions in each radial velocity bin are split into high v_{tan} ($v_{tan}>$ 100 km s^{-1}) and low v_{tan} ($v_{tan}<$ 100 km s^{-1}). This means that:

- There is a weak negative correlation (for each r-bin subsample) between v_{rad} and $M_{\rm LG}$. This is a consequence of the fact that we expect larger (negative) v_{rad} values to be correlated with a stronger gravitational pull and thus larger masses for the system
- The separation into low v_{tan} and high v_{tan} values does not show any particular trend, except at low v_{rad} values where median $M_{\rm LG}$ masses for the low v_{tan} samples consistently lie below the high v_{tan} values
- The $M_{\rm LG}$ values does not seem to be correlated with R, as median $M_{\rm LG}$ values do not vary much among different subsamples
- $M_{\rm LG}$ values have a large variation in each v_{rad} bin

From this we already see that a prediction of the $M_{\rm LG}$ value from these variables alone will be subject to a large scatter due to the nature of the data, which shows at best some weak correlations among the different features.

3.2 The Timing Argument

We can use the dynamics of the MW-M31 system to estimate its total mass. To do this, we need to solve numerically the parametric equations

$$r = \frac{R_{max}}{2}(1 - \cos\theta) \tag{1}$$

$$v = \frac{\sin\theta}{1 - \cos\theta}\sqrt{\frac{2GM_{tot}}{R_{max}}} \tag{2}$$

which hold in the case of a radial orbit (i.e. $v_{tan}\approx$ 0). Solving for this equations using r and v from the simulated halo pairs allows us to get an estimate of the total LG bound mass, which we can compare to the true M_{tot} =$M_{\rm MW}$ + $M_{\rm M31}$.

In Fig. 3.2 we compare the true values to those estimated with a simple application of the TA, fitting $M_{true} - M_{TA}$ to a simple linear relation

$$M_{true} = a + bM_{TA}$$

where, in the ideal case, the slope b should be close to one; moreover, we plot the distribution of

$$r_m = \log_{10}\frac{M_{TA}}{M_{true}}$$

As we see in Fig. 3.2 the TA tends to overpredict the total LG mass by a factor $\approx$ 0.411 in $\log_{10}$. This is due both to the assumption of purely radial motion and the fact that we are comparing the *total* bound mass of LG (estimated in the TA) to the one given by the simple

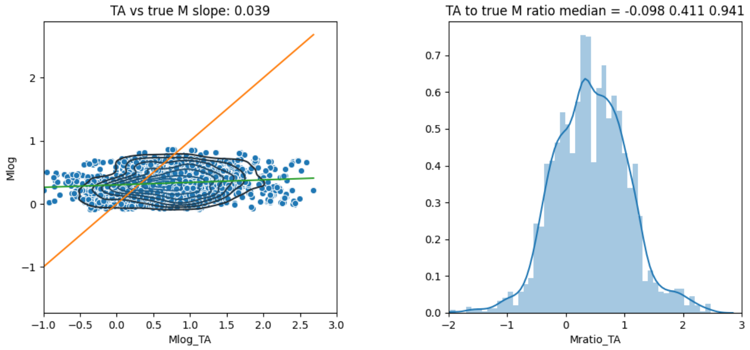

Figure 3: TA mass versus true mass and $\log_{10}$ of the *ratio* of the TA LG mass divided by the true simulated mass.

sum of $M_{\rm MW}$ and $M_{\rm M31}$. In the same figure we show the correlation between the TA-estimated mass and the simulated one, finding a very small correlation coefficient of ≈ 0.039. This can be expected, as we have seen in Fig. 3.1 that halo pairs with approximately the same v_{tan} and r are associated with a very large scatter in mass. We see that these baseline results, used only using the radial velocity v_{rad} and the halo separation r provide a result which is, on average, correct within a factor of 2.5, with a very small positive correlation between the estimated and true values.

3.3 Training

Since we find that no substantial improvement can be found training our algorithms with more than 1000 halo pairs, in what follows we will focus on halo samples drawn from Dataset-2, which have the highest mass resolution. We split the sample into a training set of ≈ 1500 halos and a test set of ≈ 500. Each train set is used to fit a linear regression, a decision tree, a random forest and gradient boosted trees on the following combination of features and dependent variables:

- $(v_{rad}, r, v_{tan}) \rightarrow M_{\rm LG}$
- $(v_{rad}, r, v_{tan}) \rightarrow M_{\rm MW}$
- $(v_{rad}, r, v_{tan}) \rightarrow M_{\rm M31}$
- $(v_{rad}, r, v_{tan}) \rightarrow M_{\rm MW}/\ M_{\rm M31}$
- $(v_{rad}, r, M_{\rm LG}) \rightarrow v_{tan}$

Mass and velocity variables are suitably rescaled by $\log_{10}$ and a proper normalization factor, $10^{12} h^{-1} {\rm M}_{\odot}$ in the case of $M_{\rm LG}$, $M_{\rm MW}$ and $M_{\rm M31}$, and $100\ \ {\rm km}\ s^{-1}$ in the case of v_{tan} and v_{rad}. This is extremely important in particular for linear regression.

In addition to the total sample of halos, obtained selecting LG pairs with the LG model defined above, we constructed additional training samples. In fact, due to the fact that halo masses are skewed are smaller values, any attempt at reconstruction with ML methods might as well be skewed towards lower masses, which would turn out to be right more often. Thus, some samples with a uniform mass distribution in $\log_{10}$ have been constructed drawing from an equal number of object per mass bin. This can be shown to produce some minor improvement on the results, however, for simplicity in the following we will focus only on results obtained with the full halo sample.

Table 2: Slope c of the predicted versus true value; median μ and scatter $\sigma$for the $\log_{10}$ of the ratio predicted over true computed for M_{LG}, M_{MW}, M_{M31} and r_M training four different ML models: Linear Regression (LR), Decision Tree (DT), Random Forest (RF) and Gradient Boosted Trees (GBT).

	M_{LG}			
	LR	DT	RF	GBT
c	0.033	0.44	0.346	0.43
μ	0.05	0.002	0.005	0.006
σ	0.27	0.282	0.232	0.230

M_{MW}			
LR	DT	RF	GBT
0.006	0.479	0.315	0.427
-0.407	-0.005	-0.148	-0.122
0.434	0.388	0.519	0.456

	M_{M31}			
	LR	DT	RF	GBT
c	0.042	0.448	0.348	0.443
μ	-0.087	-0.008	-0.069	-0.069
σ	0.401	0.411	0.392	0.398

r_M			
LR	DT	RF	GBT
0.029	0.413	0.297	0.367
0.048	-0.005	0.021	0.012
0.188	0.18	0.158	0.150

4 Results

To evaluate the goodness of our algorithms we focus on two quantities, the *slope* c of the ML-value versus the true ones and the distribution of the $\log_{10}$ ratios of the predicted to the true values, focusing on the mean μ and the standard deviation σ.

4.1 Mass of the Local Group

We first apply our machinery to the issue of the LG mass, using three features (v_{rad}, r and v_{tan}) to predict four quantities, the mass ratio r_M, the total mass of the M_{LG} and the individual MW and M31 masses (which can be of course derived from the previous two). In Table 2 we show the results for the four algorithms applied to the test data.

- Linear regression shows a very weak correlation between true values and predicted ones, with $c < 0.05$

- Tree-based methods (decision trees, random forest and gradient boosted trees) perform substantially better than the simple LR approach, as can be seen by a substantially higher c value (within $0.25 - 0.48$). In all the cases, the $\log_{10}$ of $\frac{predicted}{true}$ are centered around $\mu \approx 0.0$ with $\sigma \approx 0.2 - 0.4$. Decision trees and gradient boosted trees have a similar performace, whereas RF has on average a smaller c

- In the case of LG masses (total and individual) ML methods are not biased, which is a clear advantage compared to the naive TA approach which overpredicts M_{LG} by a factor of ≈ 2.5 on average

4.2 Tangential Velocity

We apply the same machinery to a set of different features (v_{rad}, r, $\log_{10} M_{\mathrm{LG}}$) to train our ML algorithms for the pourpose of predicting v_{tan}.

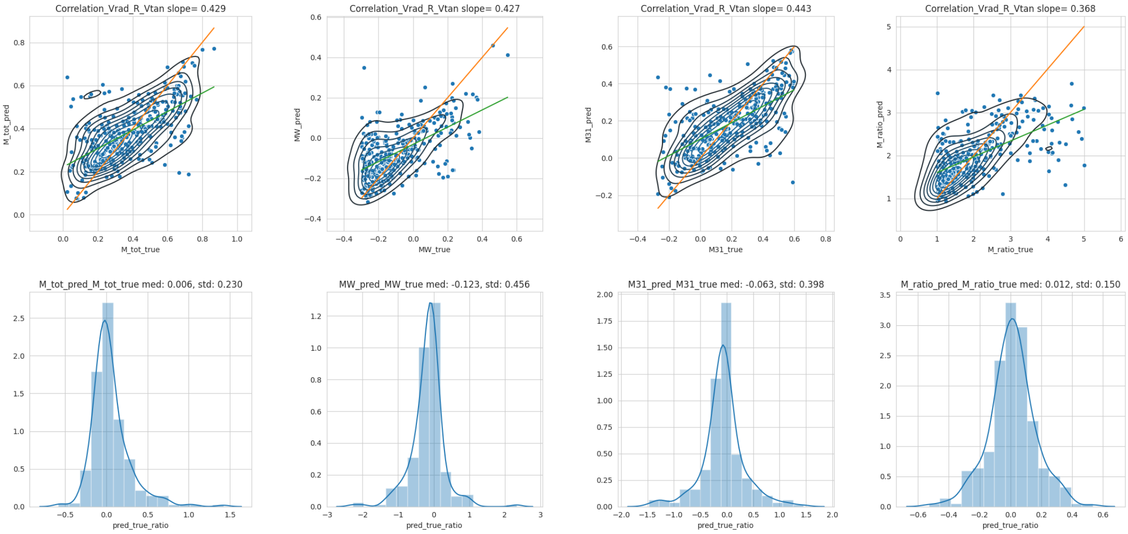

Figure 4: GBT algorithm: $M_{\rm LG}$, $M_{\rm MW}$, $M_{\rm M31}$ and r_M obtained for the test dataset. The upper plots show the distributions of predicted vs. true values with the best fit slope and the ideal case $c = 1.0$. In the lower panels we show the $\log_{10}$ values of $\frac{predicted}{true}$.

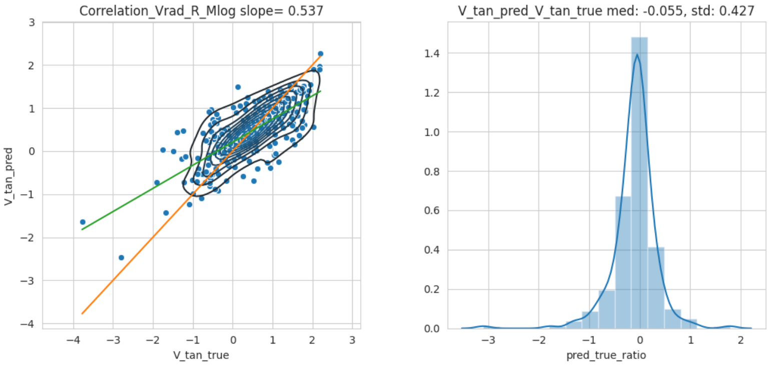

Figure 5: $\log_{10}$ of v_{tan} of Andromeda in the MW frame of reference.

Table 3: Slope c of the predicted versus true value; median μ and scatter $\sigma$for the $\log_{10}$ of the ratio predicted over true computed for v_{tan} using four different ML models: Linear Regression (LR), Decision Tree (DT), Random Forest (RF) and Gradient Boosted Trees (GBT).

	v_{tan}			
	LR	DT	RF	GBT
c	0.145	0.45	0.45	0.54
μ	-0.233	-0.014	-0.123	-0.054
σ	0.478	0.428	0.451	0.427

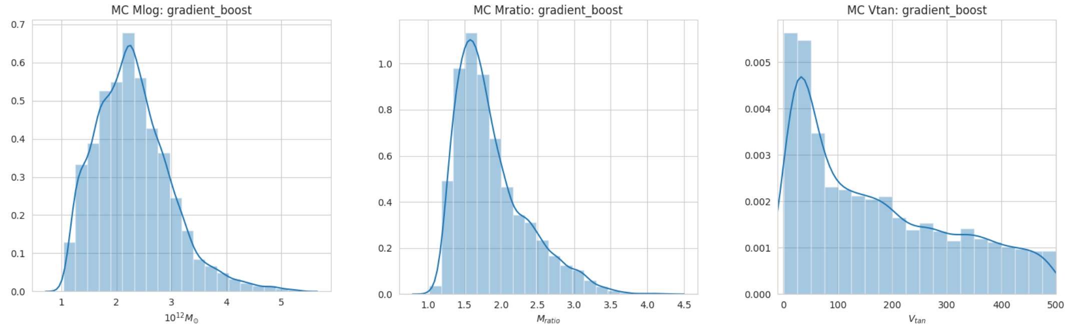

Figure 6: Total mass of the Local Group, mass ratio of $M_{\rm M31}$ to $M_{\rm MW}$ and v_{tan} obtained using the trained GBT model and 10^4 MC realisations as an input.

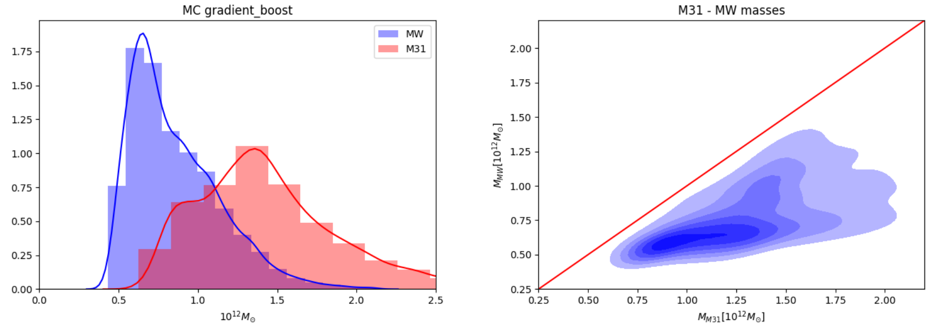

Figure 7: Distributions for individual $M_{\rm MW}$ and $M_{\rm M31}$ obtained with 10^4 MC realisations and a trained GBT model.

4.3 Montecarlo

Once these ML models have been trained, they can be used to look at the predictions for the total and individual masses, the mass ratio and the tangential velocity of the Local Group, using observational data. We generate 10000 combinations of mass, v_{tan} and v_{rad} drawing from a normal distribution with values:

- $R : \mu = 525, \sigma = 50 \quad (h^{-1}\text{kpc})$
- v_{rad}: $\mu = -110, \sigma = 10 \quad (\text{ km } s^{-1})$
- v_{tan}: $\mu = 100, \sigma = 50 \quad (\text{ km } s^{-1}$, with $v_{tan} > 0)$

the resulting pairs are then fed to the different algorithms. The results for all the algorithms are shown in Table 4 whereas in Fig. 4.3 we only show the results for the GBT.

5 Conclusions

We have shown that ML methods can be applied to the Local Group to estimate the value of variables such as mass and v_{tan} given other observables. We find that:

- The scatter in the predicted values is still very large (0.2 - 0.4 in $\log_{10}$ of the mass), but this is an intrinsic feature of the data, as shown in the preliminary analisys

Table 4: Median values ± the 20th and 80th percentiles for $M_{\rm LG}$, $M_{\rm MW}$, $M_{\rm M31}$, r_M and v_{tan} obtained from the a Montecarlo distribution within the observational values applied to four different ML models: Linear Regression (LR), Decision Tree (DT), Random Forest (RF) and Gradient Boosted Trees (GBT).

	LR	DT	RF	GBT
$M_{\rm LG}$	2.35 ± 0.11	$1.87^{+0.88}_{-0.18}$	$2.20^{+0.39}_{-0.48}$	$2.22^{+0.53}_{-0.45}$
$M_{\rm MW}$	0.86 ± 0.22	$0.86^{+0.24}_{-0.29}$	$0.81^{+0.19}_{-0.16}$	$0.77^{+0.35}_{-0.15}$
$M_{\rm M31}$	1.44 ± 0.12	$1.30^{+0.72}_{-0.28}$	$1.19^{+0.51}_{-0.12}$	$1.35^{+0.43}_{-0.26}$
r_M	1.82 ± 0.22	$2.01^{+0.20}_{-0.48}$	$1.64^{+0.56}_{-0.18}$	$1.75^{+0.54}_{-0.28}$
v_{tan}	195^{+57}_{-43}	101^{+201}_{-85}	146^{+154}_{-94}	135^{+181}_{-101}

- DTs and GBTs result in a better correlation between the expected value and the true value ($c \approx 0.4$), while LR performs very poorly

- Mass estimates obtained with ML outperform the basic TA results in that they bear a stronger correlation to the real values (for TA $c \approx 0.03$ while tree-based methods produce $c > 0.25$) and their distribution is not s is not skewed towards higher masses (in TA $\mu = 0.411$ whereas for ML we find in general $\mu \approx 0.0$)

- In the case of v_{tan}, there is a small systematic underestimation of the real values ($\mu < 0$)

- Using a Montecarlo approach, we find that our $M_{\rm LG}$ estimates obtained with different methods produce consistent results. In the case of LR, the predictions have a very small scatter, which however is not a proxy for the goodness of the approach, as we know that LR has a very small c coefficient. Tree-based methods on the other hand seem to produce robust predictions for masses and v_{tan}, albeit with a larger scatter, which should be expected as it is a property of the training data

To improve these results, we argue that is not necessary to add more data (as these models do not show to improve for halo samples above 10^3) but rather additional features. For example, it can be shown that adding energy E as a feature the correlations c improve up to a factor of ≈ 0.6, however, since energy is not a good observable for the system we did not discuss its effects in detail.