Galaxy Mergers Analysis

Shy Genel
Gabriella Contardo
Siddhanth Vinay
Gehua Zhang

Abstract

Galaxies in our universe form hierarchically, continuously merging and absorbing smaller galaxies over cosmic time. Understanding how these mergers impact the resulting properties of galaxies can help us better understand the process of galaxy formation. In particular, this project focuses on the final shape of the galaxy at $z = 0$. By building models to predict the final shape of the galaxy using machine learning techniques such as Gradient Boosting, the project focuses on understanding what factors as well as features of the mergers impact the final shape of the galaxy. We see that the shape of the galaxy can be predicted with a reasonable accuracy when the galaxy is well resolved. We also observe that the stellar metallicity and stellar mass of the merging galaxy seem to be the most important features for determining the shape of the galaxy.

1 Introduction

From the significant research conducted over the time span of the last several decades, both in theoretical and observational astrophysics, the formation of galaxies from pools of dark matter halos has been corroborated by this research. From the Cold Dark Matter (CDM) model, we see that these subhalos form by a hierarchical merger of small progenitors and accretion of diffuse matter into their ultimate massive subhalos. These mergers and smooth accretion contribute approximately in a 2:1 ratio to the mass growth of the galaxy.

From this same CDM model, the stellar mass of the subhalo grows by internal activities such as gas cooling and stellar formation as well as external factors such as through the accretion of different galaxies. How both these factors relatively affect the stellar mass of the galaxy plays a crucial role in the morphology, kinematic support and the structure of the resultant halo. Since these events are usually triggered by earlier mergers of halos, this merger history of halos plays a crucial role in understanding the nature of formation of galaxies.

To study how these bottom-up mergers into the final resultant halo play a role in the properties of this resultant halo, this merger history is represented in the form of abstract merger trees, where a branch represents the subhalos that merge in a hierarchical fashion.

Halo merger trees are directional in-trees, where all edges (branches) are directed towards a single root vertex, with the direction representing time and the root corresponding to the final resultant subhalo. By this definition, these merger trees cannot contain disjoint branches or halos that split up as time progresses. Each merger consists of a main branch that follows the path that the primary subhalo takes as it evolves into its final subhalo, and a secondary subhalo that merges onto the primary subhalo, with the secondary subhalo having a lower mass than the primary subhalo that it merges into.

The goal of this project is to understand, through the use of these merger trees, how these mergers impact the resultant shape of the final subhalo at z=0. The project focuses on making use of the properties of the participant subhalos in these mergers and studying how these mergers can be used to predict the final shape of the subhalo, along with what properties of the secondary subhalo are important in influencing the final shape of the galaxy. To accomplish this, we build datasets using the merger history of subhalos and use popular machine learning techniques to build predictive models to predict the shape of the final subhalo using these techniques, following which techniques are used to assess the salient features of the secondary subhalos that were crucial to the final shape prediction.

2 Literature Review

https://arxiv.org/pdf/1609.09498.pdf defines the mass ratio of a merger as the mass ratio between the primary and the secondary subhalo at the time when the secondary subhalo reached its maximum stellar mass. We will use that definition as well, while dealing with

the following edge cases:

1. In case the stellar mass of the secondary subhalo was maximum at snapshot k and we have no information about the stellar mass of the primary subhalo at that snapshot, we take the primary subhalo's stellar mass from snapshot $k + 1$;
2. If we see that the mass ratio is lesser than one, implying that at the point of maximum stellar mass of the secondary subhalo, the secondary subhalo had a higher stellar mass than the primary subhalo, we store the inverse of the mass ratio, which is important for later sorting the dataset based on the mass ratios.

3 Data Creation and Analysis

3.1 Data Creation

1. Select all subhalos at redshift 0 with stellar mass $> 10^{10}$ solar masses
2. Obtain the axis ratios for these subhalos and convert the value to represent the shape or flatness of the subhalo by taking $\frac{M_1}{\sqrt{M_2 M_3}}$, where M_1, M_2 and M_3 are the mass tensor eigenvalues for the subhalo - gives a shape value between 0 and 1, which is the target for the regression problem.
3. Traverse the merger history tree of each subhalo and create the data as follows:
   (a) For each snapshot where there is a merger (in case of multiple mergers, we consider the most major merger) for a subhalo, and the stellar mass of both the primary and secondary subhalos' stellar mass is greater than 0, store the following properties,
      i. Subhalo mass, subhalo gas metallicity, subhalo SFR, subhalo star metallicity, subhalo half-mass radius type gas, subhalo half-mass radius type dark matter, subhalo half-mass radius type stellar, subhalo half-mass radius type sblack hole, subhalo mass type dark matter, subhalo mass type gas, subhalo mass type stellar, subhalo mass type black hole, subhalo spin for both the primary and the secondary subhalos involved in the merger
      ii. The mass ratio of the merger
      iii. The redshift corresponding to the merger
   (b) If there is no merger or either the primary or secondary merging galaxy has a stellar mass of 0, store no information about the merger
   (c) Given all these properties stored about the mergers of subhalos, create a dataframe of these mergers by storing it in the following way:
      i. Each row corresponds to all the mergers for one particular subhalo.

ii. The mergers for each subhalo are sorted by their mass ratios in ascending order, hence the most major merger's properties for each subhalo make up the first 28 ($= 13 * 2 + 2$) columns of the dataset. Since we expect the more major mergers to impact the final shape of the galaxy the most, sorting the dataset ordered by mass ratio is crucial for the final shape prediction since we want columns of the dataset to contain similar meaningful information based on this hypothesis or else our models will not be able to learn anything from the data.

iii. Pad the missing mergers information with -1s, such that if a galaxy has information about k mergers out of 100, it has the first $k * 28$ columns corresponding to the 28 properties of the first k mergers, sorted by mass ratio, and the remaining $(100 - k) * 28$ columns for that subhalo are padded by -1s.

(d) As done with the above steps, similarly simultaneously create a dataframe where we consider only those mergers which happen at $z \leq 4$. We want to compare the performances of models on only major recent mergers vs using all the mergers from the merger history, to see if we can learn better using only the recent mergers, sorted by mass ratio.

(e) Upon testing the correlations of primary merger features to the final shape, we see that some of these features are highly correlated to the final shape and will not allow our model to learn any useful information regarding how the mergers impact the shape of the galaxy. Hence, we subset the dataset to contain columns corresponding to only the secondary features, mass ratio and redshift of each merger and try to use these features to predict the final shape of the galaxy.

Creation of this dataset on the IllustrisTNG100 simulation leads to a total of 3154 galaxies in the dataset, and 48192 galaxies in the IllustrisTNG300 simulation dataset.

3.2 Correlation Analysis

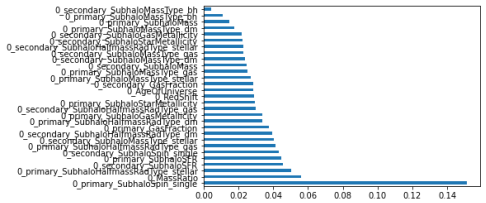
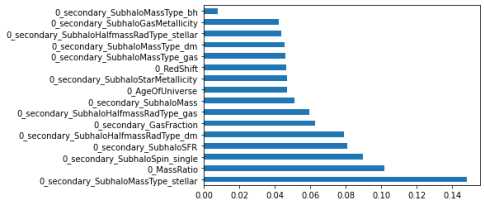
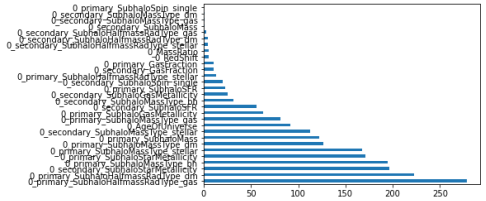
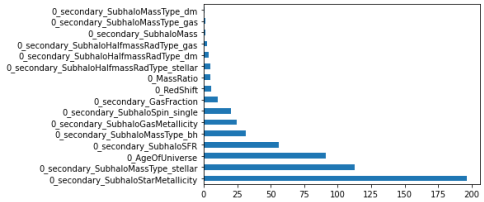
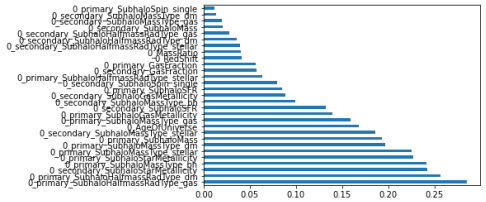
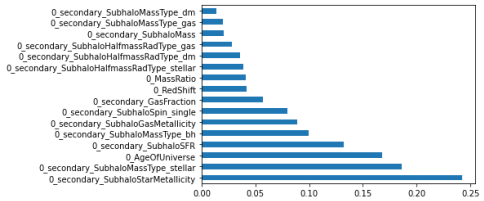
	Use Primary and Secondary Features	Use Only Secondary Features
Random Forest		
Univariate Selection		
Pearson's Rho		

Table 1

Table 1: Different measures of correlation between Galaxy shape subhalo features

1. To analyze the correlation of each features of the most major merger (29 features in total, 13 from each primary and secondary merger and 3 general features) with Galaxy shape, we have tried three methods: Random Forest Regressor's feature importance, Univariate Feature Selection and Pearson's Rho, each method will give us a score list of how features are related to Galaxy shape, we only keep top 10 ranked features. Then we calculate the occurrences of all features in three methods. If a feature appears three times in top 10 rank (it is recommended by all three methods), it is considered as highly correlated to Galaxy shape.

2. We observe a high correlation between Galaxy shape and primary features, rather than secondary features, which makes sense in Galaxy merging events due to a dominant role of primary features (in Mass, Star Metallicity, Star Mass, etc). However our research is to discover the impact of secondary features on Galaxy shape, we will exclude primary features in future analysis to explore the roles of secondary features.

3. Analyzing the correlation of secondary features and Galaxy shape gives us 7 highly correlated features: secondary_SubhaloSFR, secondary_SubhaloStarMetallicity, secondary_SubhaloMassType_stellar, secondary_GasFraction, secondary_SubhaloSpin_single, MassRatio and AgeOfUniverse. We then plot the value of these features with Galaxy shape, however the scatter plot looks arbitrary (hard to find an obvious linear relation of each feature w.r.t shape).

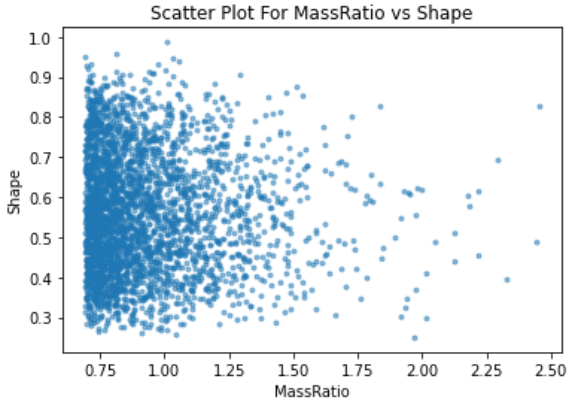


Figure 1

Figure 1: We do not observe any linear correlation of mass ratio and Galaxy shape

4. Then we analyze the multicollinearity inside secondary features, we see 'secondary_SubhaloMass is highly correlated to both secondary_SubhaloMassType_gas and secondary_SubhaloMassType_dm; secondary_SubhaloStarMetallicity is highly correlated to RedShift; AgeOfUniverse is built from RedShift and so they are also highly correlated.

5. After above steps, we believe some machine learning methods could be applied but before that, we need to deal with data distribution and find a way to scale the data.

3.3 Data Distribution and Scaling

1. From the original data distribution, we see a highly unbalanced data structure. Using MinMaxScaler, MaxAbsScaler or log1p would also keep that structure. The unbalanced structure is due to the large amount of numbers close to zero. Another problem is that each column would have a different size of zero values so we cannot just take them out. To solve this we decide to separate zero values before scaling then assign them a customized value. For other non-zero values, we decide to use log10 scaling. After exploring the data, we find no negative values in the dataset, the smallest value is 0 (which means after scaling the original 0 value should also be the smallest value in our dataset). So our approach is, after log10 scaling we assign a small value for the original 0 values, this small value equals the log value of the smallest value of this column minus one. In this approach, each column will have a different zero replacement.

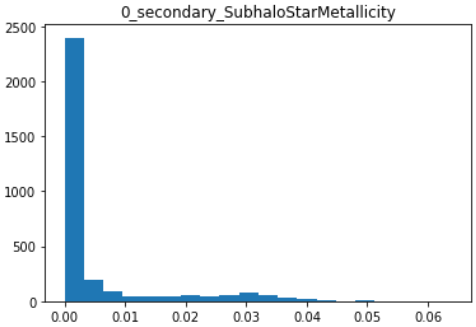


Figure 2: No scaling distribution for SubhaloStarMetallicity

2. After scaling, the unbalanced feature is more likely to have a duplex plot, the left part is derived from zero values and the right part contains more information to train the model. Our scaling method separates these two parts and hope it would be better for the model to learn, since we do believe zero values contain some information but it is better for models to interpret them differently compared to non-zero values.

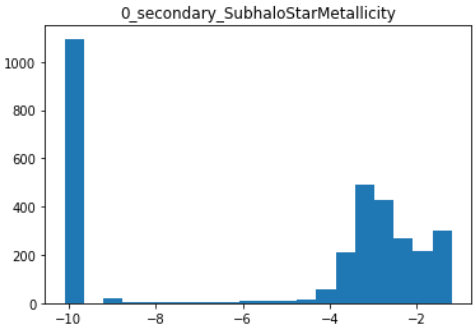


Figure 3: After scaling zero values the distribution for SubhaloStarMetallicity

3. As for the missing value (often occurs at minor mergers), we assign it with -1 in our original dataset, which is also not supported by log scaling. Following the same idea we replace it with the minimum value of this column minus 2. If this column contains both zero values and missing values, its distribution will be split into three parts. The left part is missing values, the middle part is zero values, the right part contains log10 scaled data.

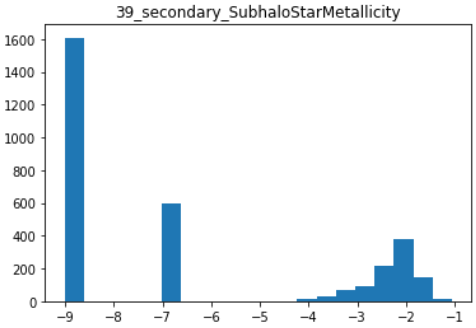


Figure 4: After scaling zero and missing values the distribution for SubhaloStarMetallicity

We will apply this scaling method in the Neural Network and Symbolic Regression model.

4 Methodology and models used

4.1 Models Tried

1. Lasso Regression - hypothesis that the lasso regression model will suppress most of the features' coefficients to 0, while retaining only few features relevant for prediction. However, the lasso regression model performed poorly on the dataset and was discarded with an R2 score of only 0.14 on the test data.

2. Random forest - Tree based models do not require any scaling of the data and the feature importances can give us a good idea of what features were relevant for prediction. However, we see that the random forest model also fails to learn well on the data, with an R2 score of 0.26 on the test data.

3. Gradient boosting - No need to scale data, model gives us feature importances of those features that are relevant for prediction. We see that the gradient boosting model performs the best on the data and decide to use this model for further analysis and prediction, with the best models obtaining an R2 score of 0.37 on the test data.

4. Neural Networks - Need to scale the data. mse as a loss function would result in a prediction of the mean of y_train because of the non-uniform distribution in the 'target' to predict (the shape). We use customized loss functions to penalize the model to predict the mean. We have both tried SGD and Adam as our optimizers.

5. Symbolic Regression - Scaling the data might have an impact on the results. We also use customized loss functions. Need to specify math operations that are allowed in model. To avoid too many input features we use most major merger's features. Also we build a dataset which uses some selected features from different mergers.

4.2 Metrics Used

1. R2 score: This is a statistical measure that represents the proportion of the variance of the target variable that's explained by the regression model.

2. RMSE: Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

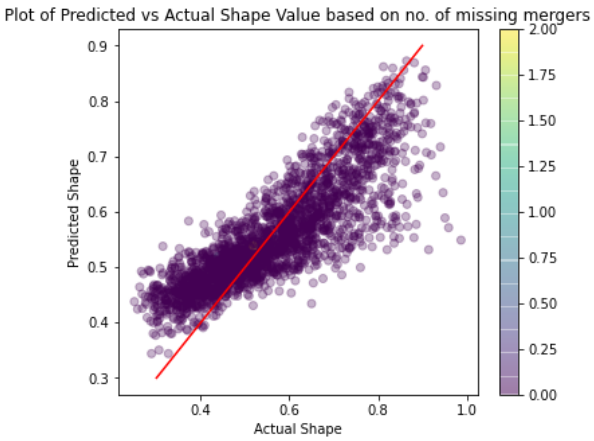
5 Model Fitting

5.1 Gradient Boosting

5.1.1 First Approach

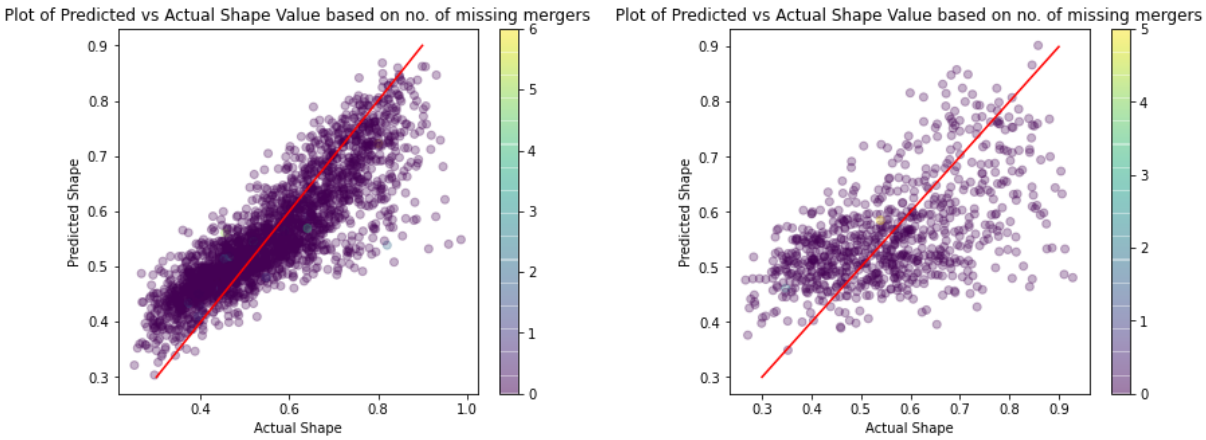
We take the 2 datasets sorted by mass ratio with all mergers and those mergers that are $z < 4$. We subset the data into considering only the top 10, 20, 40 and 60 mergers and thus build models to predict the shape using these datasets, to evaluate which dataset performs better.

Train Graph

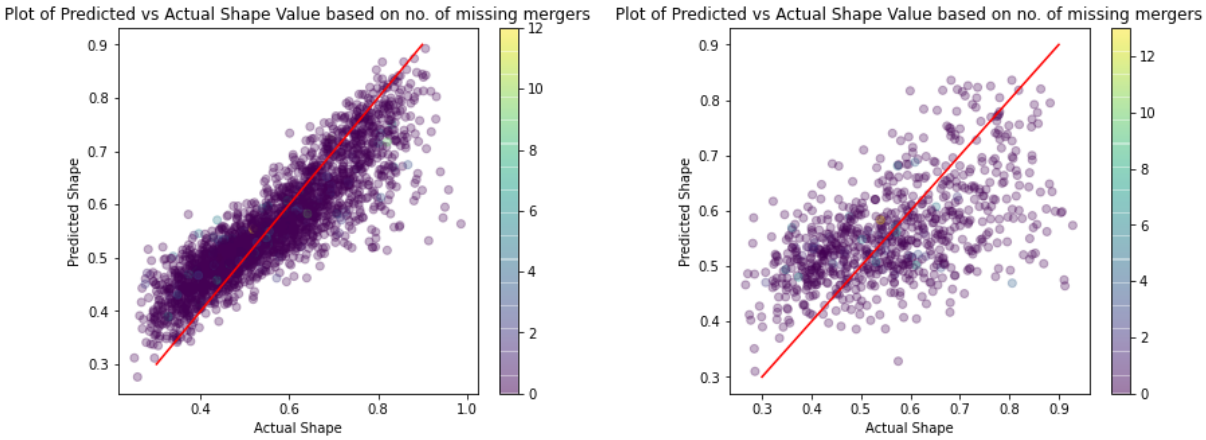


Test Graph

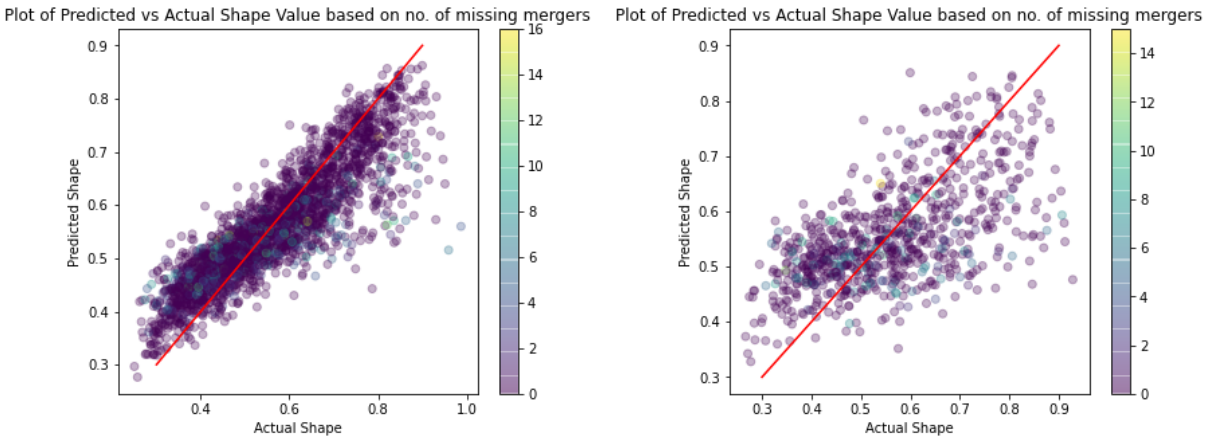
dataset = 10, Train R2 = 0.67, Train RMSE = 0.084,
Test R2 = 0.25, Test RMSE = 0.124
Least major mergers' features do not have high importance values,
except for secondary_subhaloMassType_dm_10



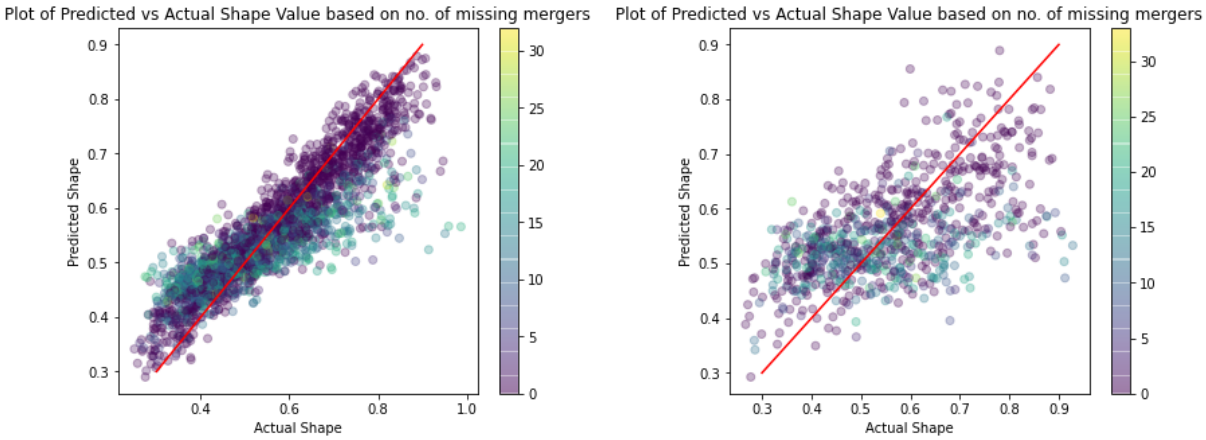
dataset = 10, $z < 4$, Train R2 = 0.702, Train RMSE = 0.082,
Test R2 = 0.302, Test RMSE =0.119
Least major mergers' features do not have high importance values,
except for mass_ratio_10



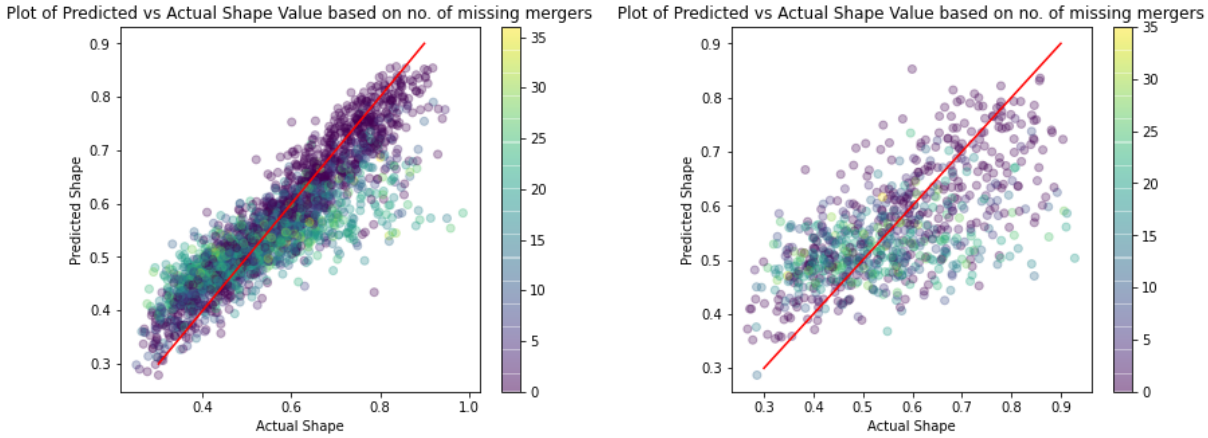
dataset = 20, Train R2 = 0.74, Train RMSE = 0.076,
Test R2 = 0.31, Test RMSE =0.119
Least major mergers' features do not have high importance values,
except for secondary_subhaloGasMetallicity_20 and redshift_18



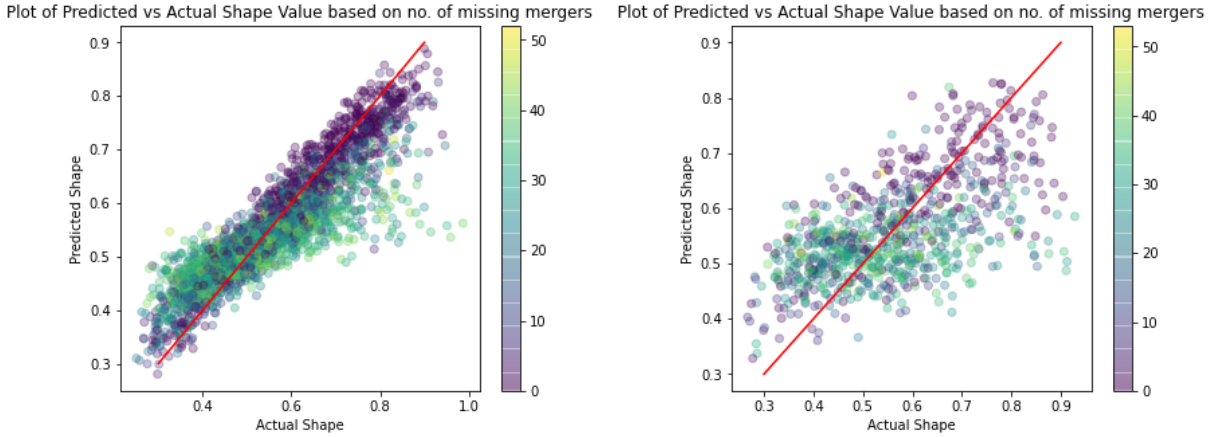
dataset = 20 , $z < 4$, Train R2 = 0.74, Train RMSE = 0.075,
Test R2 = 0.36, Test RMSE =0.115
Least major mergers' features do not have high importance values,
except for secondary_subhaloGasMetallicity_19



dataset = 40, Train R2 = 0.76, Train RMSE = 0.073,
Test R2 = 0.35, Test RMSE =0.116
Least major mergers' features do not have high importance values,
except for mass_ratio_40 and secondary_subhaloGasMetallicity_24



dataset = 40, $z < 4$, Train R2 = 0.76, Train RMSE = 0.073,
Test R2 = 0.4, Test RMSE =0.111
Least major mergers' features do not have high importance values,
except for mass_ratio_40, secondary_subhaloGasMetallicity_26 and secondary_subhaloGasMetallicity_1



dataset = 60, Train R2 = 0.76, Train RMSE = 0.072,
Test R2 = 0.36, Test RMSE =0.115
Least major mergers' features that appear are mass_ratio_58, mass_ratio_60,
redshift_60, mass_ratio_50 and mass_ratio_47

dataset = 60, $z < 4$, Train R2 = 0.76, Train RMSE = 0.073,
Test R2 = 0.37, Test RMSE =0.113
Least major mergers' features do not have high importance values
except for mass_ratio_45 and secondary_subhaloMassType_stellar_19

Table 2

Results on using 10, 20, 40 and 60 mergers, including and excluding $z > 4$ mergers, as described in Table 2, show that $z < 4$ mergers generalize better, and the model with 60 mergers gave best results. This increased performance on increasing the number of mergers could potentially be due to the following hypotheses:

1. Having more history helps make better predictions, even if these mergers are sorted by mass ratio due to the fact that the more mergers there are, the higher the shape value of the subhalo is.

2. Even the lesser major mergers (based on the mass ratio of the merger) can provide information about the resultant shape value.

3. The number of mergers contains some information about the age or mass of the galaxy which could help with the shape prediction.

We also notice a clear bimodality in the predictions, with the predictions having two clear lines - one along the 1:1 line and a second flatter set of predictions. To analyze the cause for this bimodality, we plotted the predictions colored by the 14 features of the most major merger as well as the top 10 features with the highest feature importance for prediction to see if the bimodality is correlated to the value of any of these features. We did not however notice any correlation between the values of any of these features and the bimodality.

We also hypothesize that the number of mergers subhalo has had in its merger history could explain the bimodality in predictions. By looking at the prediction plots, (number of missing mergers in the graph), we can clearly see that the bimodality is correlated to the number of mergers.

5.1.2 Second Approach

Following this we consider the following hypothesis : the number of missing mergers contains some information about the final shape of the galaxy. To test this, we build a model using the first 30 mergers and the number of mergers as a feature and compare it to a model just using 30 mergers to predict the shape.

Model 1: Just using the first 30 $z < 4$ mergers for prediction:

Figure 5A Figure 5B

Train R2 = 0.76, Train RMSE = 0.073, Test R2 = 0.37, Test RMSE = 0.114

Figure 5: Prediction plots on the train and test set when we exclude the number of mergers feature from our prediction model

Model 2: Using the first 30 $z < 4$ mergers for prediction including the n_missing mergers column:

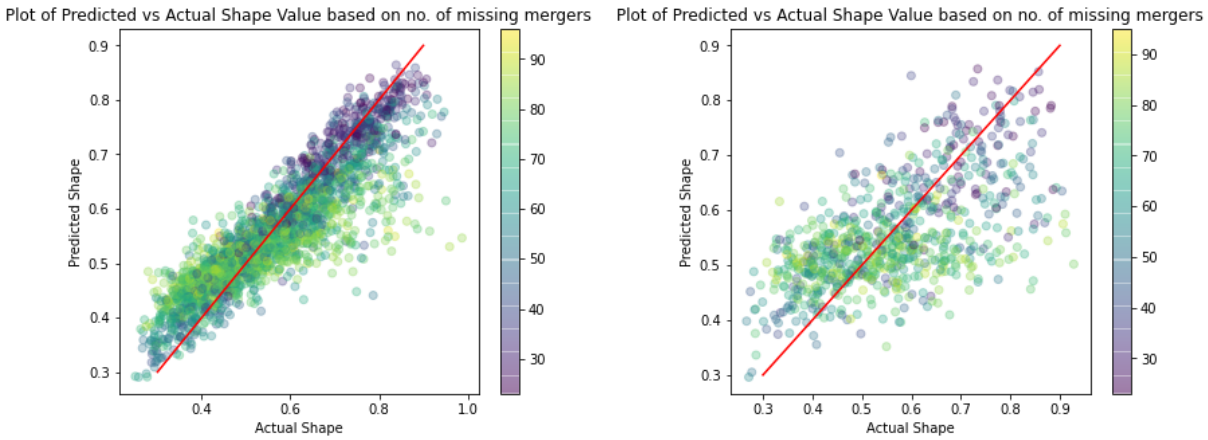


Figure 6A Figure 6B

Train R2 = 0.76, Train RMSE = 0.073, Test R2 = 0.36, Test RMSE = 0.114

Figure 6: Prediction plots on the train and test set when we include the number of mergers feature to our prediction model

From these results, it is clear that the number of mergers does not actually contain any information about the final shape of the galaxy and that there is just a correlation of this with the bimodality.

Model 3: Following this, we want to test whether the number of mergers of a subhalo can be predicted from features of the subhalo itself. To test this, we only use the 7 most major mergers from a subhalo's merger history since all subhalos have at least 7 mergers and using more mergers than that will leak information about the number of mergers a galaxy has (since these columns are padded by -1s). Using this dataset, we try to predict the number of mergers

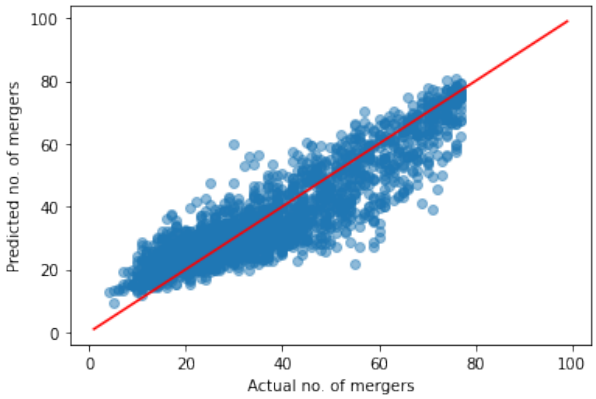


Figure 7A

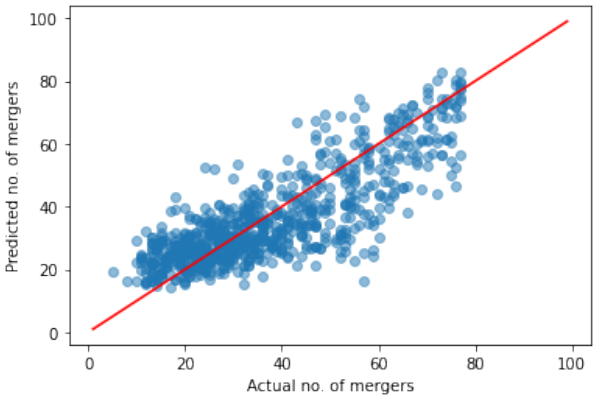


Figure 7B

Train R2 = 0.81, Train RMSE = 7.88, Test R2 = 0.66, Test RMSE = 10.27

Figure 7: Prediction plots for predicting the number of mergers given the 7 most major mergers, on train (7A) and test data (7B) points

From the R2 score, RMSE value and figure 3, we can see that just using the information of the first 7 mergers contains enough information to predict the number of mergers in a galaxy with a high R2 score of 0.66 on the test data.

Model 4: Based on the above results, we decide to test whether the number of mergers can simply be predicted by the final stellar mass of the galaxy instead of its merger history by building a model to test this:

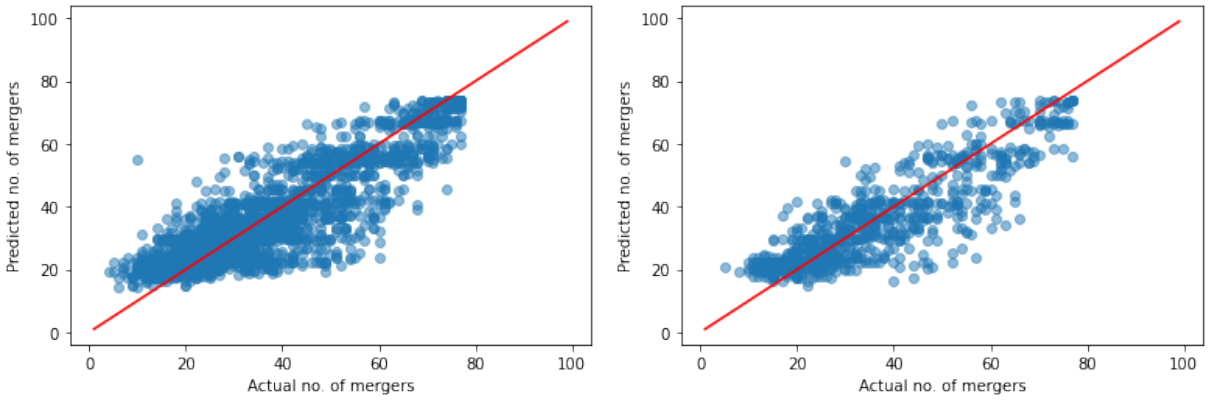


Figure 8A Figure 8B

Train R2 = 0.77, Train RMSE = 8.63, Test R2 = 0.75, Test RMSE = 8.81

Figure 8: Prediction plots for predicting the number of mergers given the stellar mass of the subhalo, on train (8A) and test data (8B) points

Looking at the R2 score and the prediction plots from figure 4, we see that just using the final stellar mass of the galaxy can predict the number of mergers with an R2 score of 0.75 on the test data, which is even higher than the R2 score obtained by using the merger history information. Thus, this helps us conclude that the number of mergers a subhalo has undergone is very highly correlated with the final stellar mass of that subhalo.

Thus, although in section 1 where we see the bimodality in the prediction plots of the shape seems to be explained by the number of mergers in a subhalo's merger history, from the results in section 2, we can see that the number of mergers of a subhalo does not contain any information about its final shape. From sections 3 and 4, we see that these number of mergers can not only be reasonably predicted by considering the information from the 7 most major merger of a subhalo, but also just from the stellar mass of the subhalo - which implies that the bimodality in predictions seen in section 1 can also simply be explained by the stellar mass of the subhalo.

Model 5: From the above hypothesis, we decide to see if the bimodality in the shape predictions can be explained by the final stellar mass of the galaxy.Prediction plots colored by log10(stellar mass) on the train and test data for the dataset containing the top 60 $z < 4$ mergers.

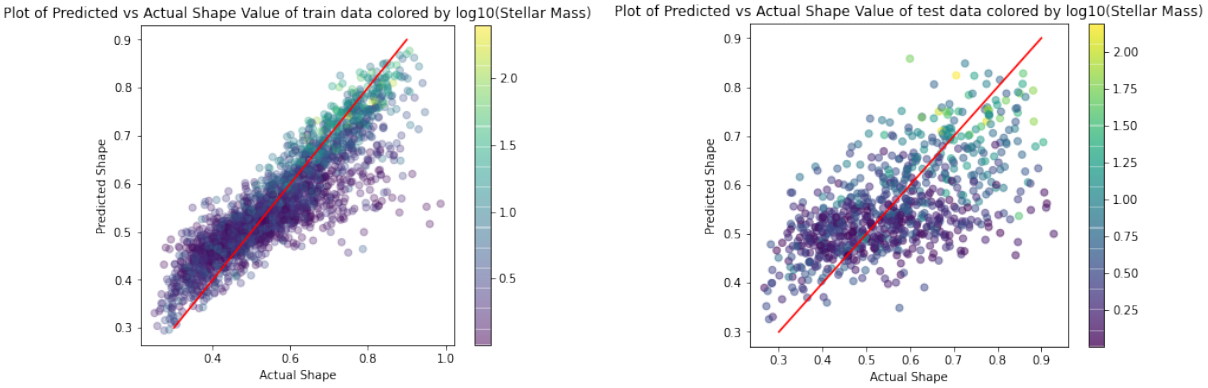


Figure 9A

Figure 9B

Train R2 = 0.77, Train RMSE = 8.63, Test R2 = 0.75, Test RMSE = 8.81

Figure 9: Prediction plots colored by log10(stellar mass) on the train and test data for the dataset containing the top 60 $z < 4$ mergers

Looking at the plots from Figure 5, we can clearly see that this bimodality in predictions can also be the stellar mass of the graph, with values below a stellar mass of $10^{0.5}$ having flatter and poorer predictions.

Model 6: Based on the information above, we build a model for only those galaxies with stellar mass $> 10^{0.5}$ to evaluate the predictions, using the dataset with the top 60 $z < 4$ mergers. We end up losing close to 2000 galaxies as a result of this cutoff and end up with only around 1238 galaxies in the resultant dataset.

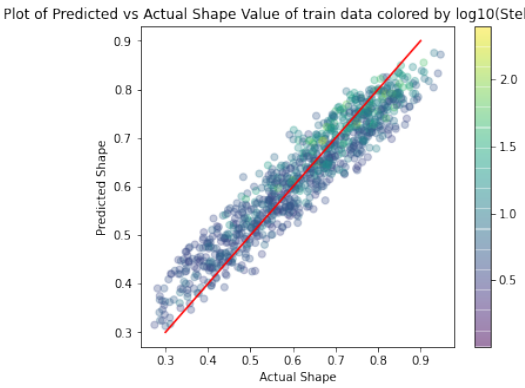


Figure 10A

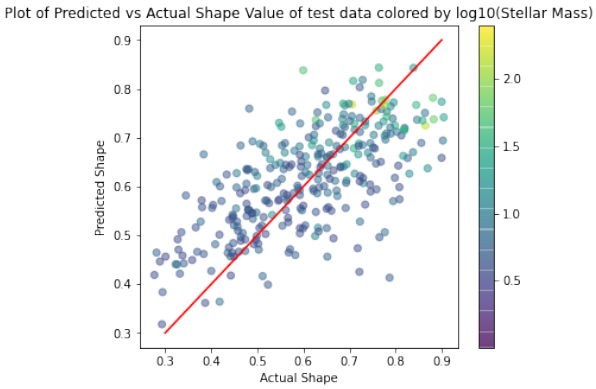


Figure 10B

Train R2 = 0.88, Train RMSE = 0.0523, Test R2 = 0.43, Test RMSE = 0.108
5-fold cross-validation score: 0.44

Figure 10: Prediction plots on the train and test data when we build a prediction model for the shape of only those galaxies with a stellar mass $> 10^{0.5}$

Looking at these results, we see that the model just using these subhalos has a better R2 of 0.43 on the test data when compared to using the full data, and the bimodality in predictions is eliminated.

From these results seen, there are two possible hypotheses for these results - either for low mass galaxies where the model predictions are poor, the shape is predicted by other features unavailable in the dataset, or that due to the fact that these galaxies have only a few mergers, there simply isn't enough information contained in these few mergers of these low mass galaxies to predict their final shape. Try to sharpen this (i know what you mean but am not sure an external reader will be able to distinguish the two options from this text)

To test the hypothesis, we use the IllustrisTNG300 simulation which is a lower resolution simulation and contains a lot more galaxies, and we build models to predict the shape in this simulation.

Model 7: We first build a model on the entire IllustrisTNG300 dataset which contains 16 times more data points than the counterpart IllustrisTNG100 simulation. The following were the results obtained on this dataset:

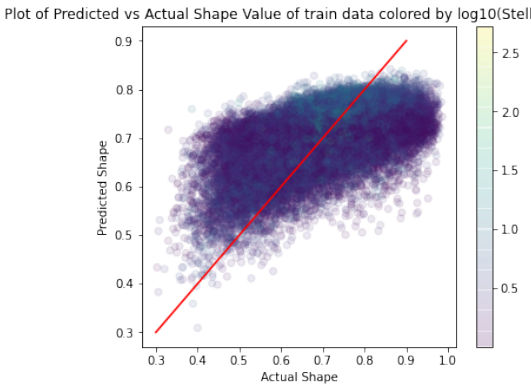


Figure 11A

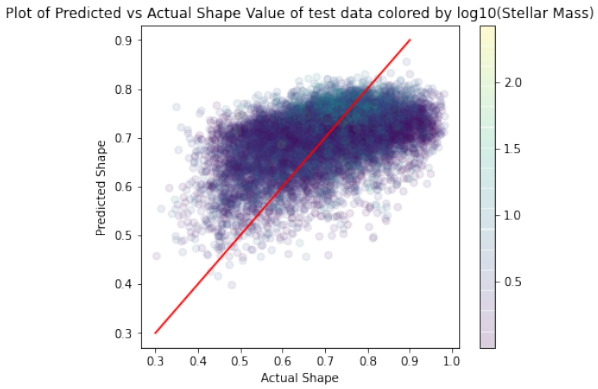


Figure 11B

Train R2 = 0.30, Train RMSE = 0.102, Test R2 = 0.23, Test RMSE = 0.107

Figure 11: Prediction plots on the train and test data for the entire TNG300 dataset

We see that the model performs significantly worse on the TNG300 dataset when compared to the TNG100 dataset, supporting the hypothesis that there is not enough information in lower resolution galaxies to make accurate shape predictions.

Model 8: However, in order to make an accurate comparison between the two models, we create a dataset similar to the mass thresholded TNG100 simulation dataset. In order to achieve this, we consider only those subhalos with a stellar mass $> 10^{0.5}$, and then replicate the size and shape distribution of the TNG100 dataset by considering shape bins of 0.05 (subhalos with shape between 0.50 and 0.55 will be part of the same bin for example), and try to sample the same number of data points as the TNG100 dataset for each bin. We then build a gradient boosting model on this dataset

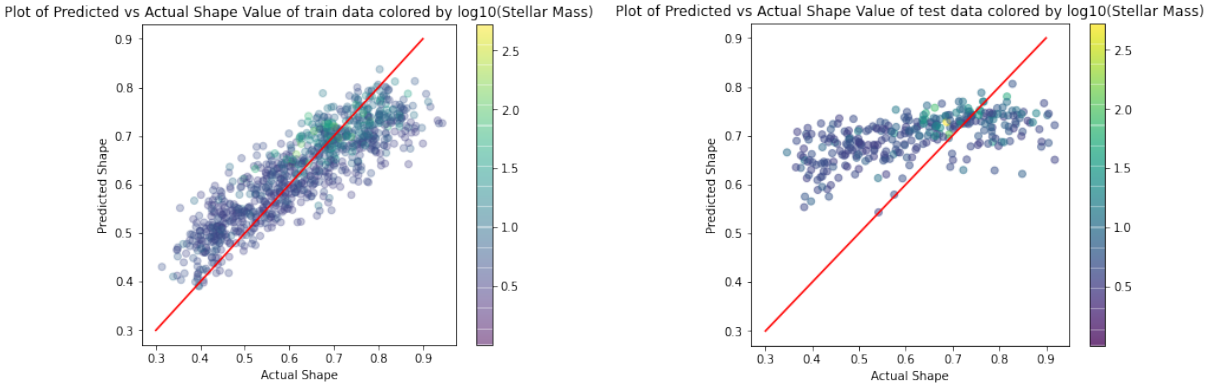


Figure 12A

Figure 12B

Train R2 = 0.73, Train RMSE = 0.072, Test R2 = 0.395, Test RMSE = 0.149

Figure 12: Prediction plots on the train and test data
for the mass thresholded and sampled TNG300 dataset

Although from the cross-validation and test R2 scores, it seems like the model performs only slightly worse than the model on the corresponding TNG100 dataset, we see that the RMSE and the prediction plot of the test dataset clearly indicate that the predictions are in fact much worse than the TNG100 model.

From this, we can almost certainly conclude that the prediction model is dependent largely on the resolution of the dataset and a good resolution of the mergers, containing valuable information about the mergers is important for making good shape predictions. However to fully corroborate this hypothesis, we would require a simulation of a higher resolution than the TNG100 simulation, which is currently unavailable.

5.2 Neural Network

1. For Neural Networks, we first try to use mse as our loss function, the model will tend to predict the mean of y_train. This may be due to the structure of our input features containing a considerable amount of missing values and zero values, as we shown above some columns may include over one third of the missing values. Those missing values are replaced with a same number in input features but they have a different target values, they contain no extra information for models to predict the target, so the model may think predicting the mean (mean has nearly the highest occurrence in target distribution) is good enough.

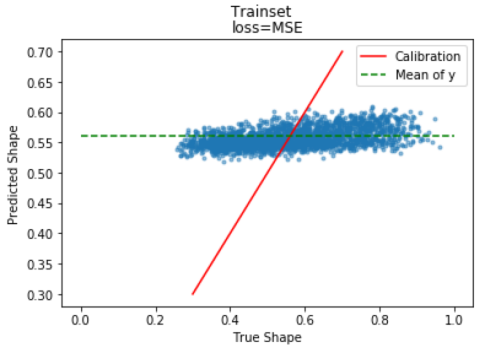


Figure 13A

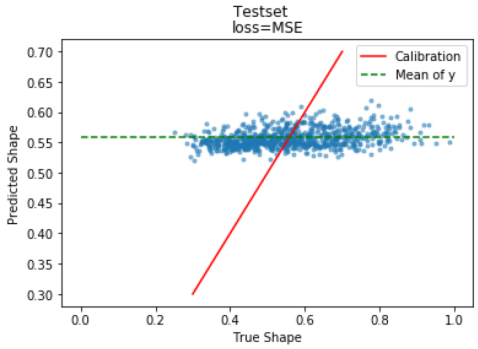


Figure 13B

Figure 13: Prediction plots on the train and test data
for the MSE loss function in Neural Networks

2. To overcome this problem we have defined three customized loss function, custom_loss, custom_loss_better and custom_loss_better2, all of them will penalize the model if it predicts the mean, and the penalty is based on its parameter settings. Those three loss functions differ from their customized levels: custom_loss_better2 is the most customized loss function and we can set two parameters to control its penalty while custom_loss_better can only set one, and custom_loss is fixed. Two parameters are named factor and fac_div, changing them would result in a different loss function, thus a different training process. This is also to say, our model and parameter settings are sensitive to scaling method and dataset. (Note that factor and fac_div can't be any arbitrary positive values, some combinations of them will make our model stuck to local extreme and fail to train)

   custom_loss:

$$\text{Mean}[(y_{pred} - y_{true})^2 * |y_{true} - \overline{y}_{train}| * \text{factor} + \varepsilon]$$

   custom_loss_better:

$$\text{Mean}[(y_{pred} - y_{true})^2 * (|y_{true} - \overline{y}_{train}| - (y_{true} - \overline{y}_{train})/3) * \text{factor} + \varepsilon]$$

   custom_loss_better2:

$$\text{Mean}[(y_{pred} - y_{true})^2 * (|y_{true} - \overline{y}_{train}| - (y_{true} - \overline{y}_{train})/\text{fac\_div}) * \text{factor} + \varepsilon]$$

3. Another parameter setting is in optimizer, for this dataset we have both tried SGD and Adam.

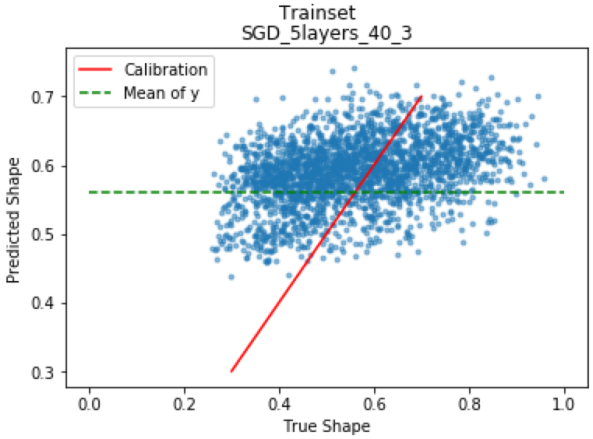


Figure 14A

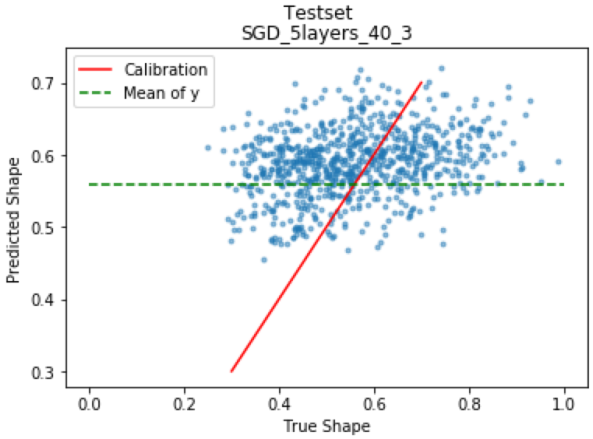


Figure 14B

Train R2 = 0.102, Train RMSE = 0.141,
Test R2 = 0.024, Test RMSE =0.139

Figure 14: Prediction plots on the train and test data for custom_loss_better2 as loss function and SGD as optimizer

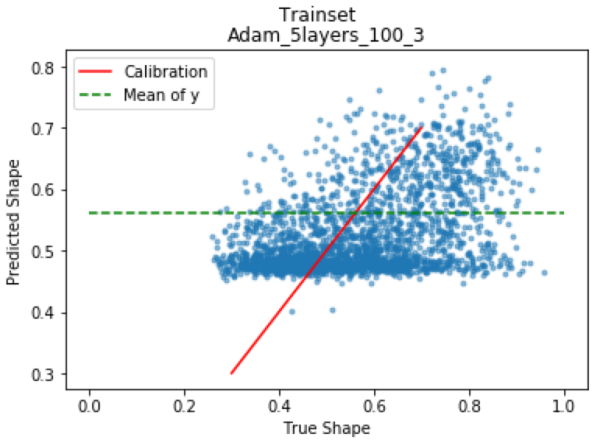


Figure 15A

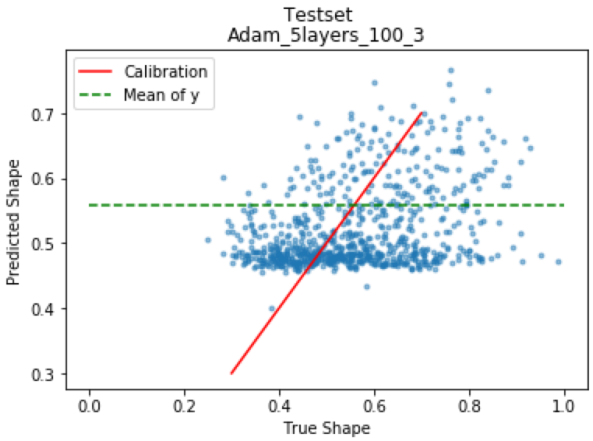


Figure 15B

Train R2 = 0.135, Train RMSE = 0.139,
Test R2 = 0.031, Test RMSE =0.138

Figure 15: Prediction plots on the train and test data for custom_loss_better2 as loss function and ADAM as optimizer

4. However as we try different values of factor and fac_div, even the model becomes different, the performance is hard to get better. From the training loss plot, we see both training error and validation error are hard to overcome a fixed error value, the model reaches its limitations.

5. Why isn't our model performing well even if we use customized loss function? As we can see from the SGD prediction plot, if we penalize the model (too much) to predict the mean of y_train (around 0.56), it indeed avoids predicting that value but fails to find a pattern. From the Adam prediction, if we penalize the model not enough, it continues predicting the a value around the mean. One explanation would be, our model is hard to find a pattern between features and label, it chooses the simplest way to just predict the mean, if not possible, it fails to learn.

5.3 Symbolic Regression

1. Symbolic Regression has the same problem as in Neural Networks, the model will tend to predict the mean of train_y. We try to use the same customized loss function, it indeed improves the prediction performance (otherwise Symbolic Regressor will predict a constant) but the problem still exists, we still observe that many predicted train/test targets are around 0.5.

2. Another problem is the math operations. As we can see from Figure 16 and 17, prediction values always lie above a certain value (around 0.48), this is due to the safety computation rules by gplearn. To avoid complex numbers or other mathematical conflicts, gplearn defines $\sqrt{x}$ as $\sqrt{|x|}$ and also $\log(x)$ as $\log(|x|)$, and any number divided by 0 would result 1. This may result in a prediction that all values are greater than some constant. Constraining the use of some math operations may avoid this problem but reduces the model flexibility. To overcome the math operation problems, customized math operations may be needed.

3. To avoid too many input features (which Symbolic Regression may fail to handle), we build two datasets that have fewer columns. The first dataset contains all major merger features with 16 columns; the second contains 4 important features from first 3 major mergers (SubhaloMassType_stellar,SubhaloStarMetallicity, GasFraction,AgeOfUniverse), and 2 mass ratio features from 40th major mergers (38_MassRatio, 39_MassRatio). 14 columns in total. This selection criterion is from the feature importance of Gradient Boosting.

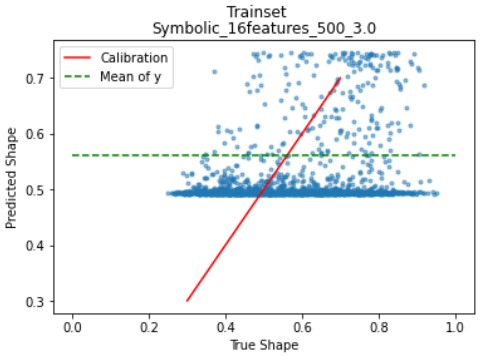


Figure 16A

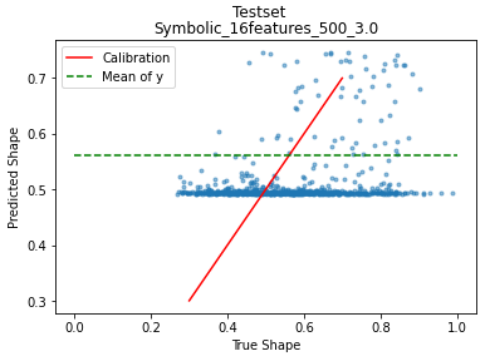


Figure 16B

Train R2 = -0.039, Train RMSE = 0.149,
Test R2 = -0.059, Test RMSE =0.153

Figure 16: Prediction plots on the train and test data for major merger features(16 columns) and custom_loss_better2 as loss function

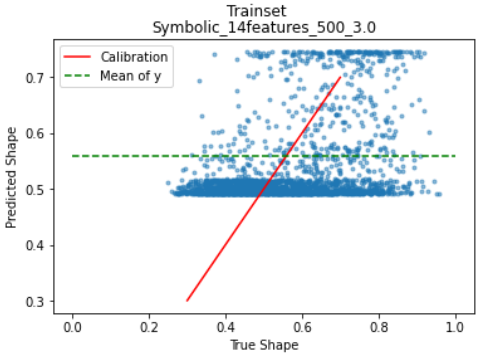


Figure 17A

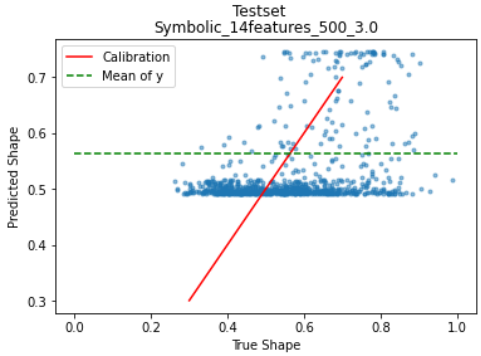


Figure 17B

Train R2 = 0.0849, Train RMSE = 0.14,
Test R2 = 0.0635, Test RMSE =0.143

Figure 17: Prediction plots on the train and test data for selected features(14 columns) and custom_loss_better2 as loss function

An example of formula that the model generates:

$$\sin(\sin(\cos(\cos(\frac{0.77757\sin(\cos(\cos(\frac{0.68}{X_9})))}{X_9}))))$$

4. Though both of approaches perform badly in making predictions on the train/test set, we see a slight improvement of using selected features.

6 Feature Importance

In order to compute the feature importances, we use the $10^{0.5}$ mass cutoff dataset containing only the $z < 4$ mergers (with 60 mergers?) from the IllustrisTNG100 simulation, since this gave us the most accurate predictions and has the subset of galaxies that have the highest resolution and based on our conclusions, is thus the only dataset available to extract the important features for shape prediction.

We consider multiple methods, which have their caveats:

1. Permutation importance, which permutes the values of a particular feature and then compares the predictions of the resultant dataset with the predictions from the original dataset can be affected by correlated features. We see that this yields no important information about the important features since all the features have some level of multicollinearity and permuting one feature just forces the model to recreate the information of that feature from all the other features using this multicollinearity. It is also not feasible to run this for all the 2800 features in the dataset so we only consider those features with a high feature importance based on the results of the gradient boosting.

2. Just using the feature importances from the model gradient boosting model is also an option. This computes the importances of features based on what features most strongly influenced the model's decision tree creation. This feature importance technique can be affected by the cardinality of the features (not as much of a concern since all features in our dataset are continuous), and randomness of the model - what features are selected depends on the randomness. A model with several iterations, each containing a subset of the dataset for training, however tends to be more robust in what features are important for shape prediction and thus feature importances from the gradient boosting model seems like the best approach for evaluating feature importances of features in a feasible manner in terms of time.

3. Other methods such as sklearn - kbest features and just looking at correlations between model features and final shape may also give us an idea about the important features, and plotting the correlations of the important features for each of the 60 mergers with respect to the shape can lend support to extracting what features of the merger impact the final shape of the galaxy.

Results of the GB feature importance method:

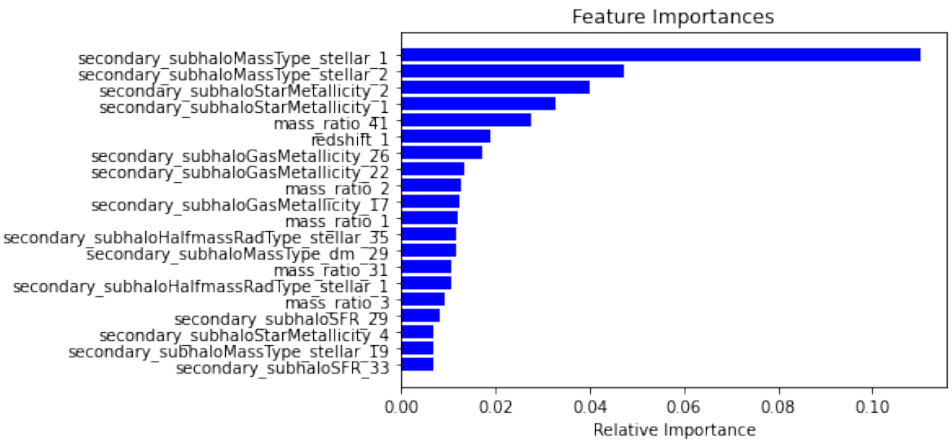


Figure 18

From the Figure 18, it is evident that the stellar mass and stellar metallicity of the 2 most significant mergers clearly have the highest importance for shape prediction. The mass ratio of the merger also seems to be an important feature as we see the mass ratio of the top 3 significant mergers also being among the top 20 most important features. Additionally, the redshift of the most important merger also has an impact on the shape prediction. We see some unexpected features such as the mass ratio of the 41st significant merger, but such things can be expected due to randomness and the fact the feature importances is just what features helped the model make stronger predictions, which is purely based on correlations of the features to the final shape value.

After taking a look at these various feature importance techniques to try and understand what features of the merging galaxy most influence the final shape of the galaxy, it seems like the star metallicity, stellar mass, mass ratio and redshift of the most significant mergers have the strongest impact on the model's prediction of the final shape value of the galaxy.

Correlation plots:

We plot the correlations of the earlier recognized important features for mergers 1-60 with respect to the shape value.

We make two correlation plots - one containing only those mergers for which the merger has occurred and the other containing all mergers, with missing mergers information filled with -1s.

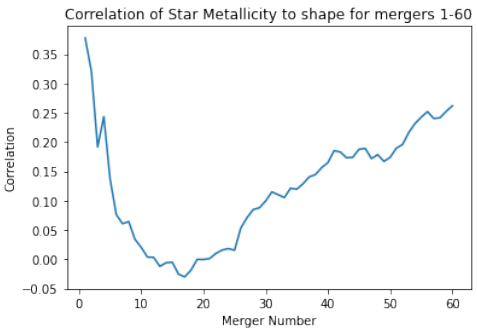
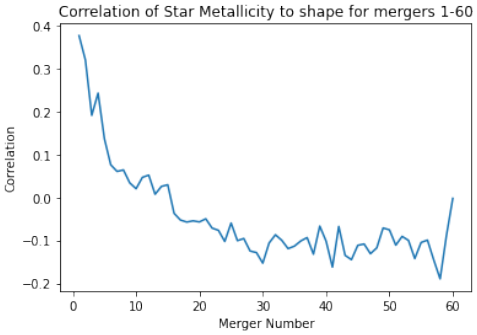
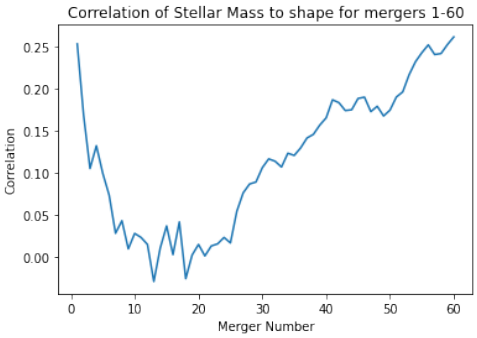
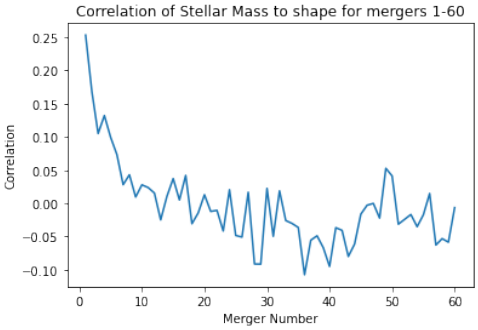
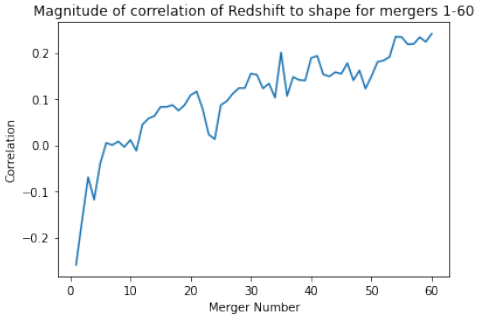
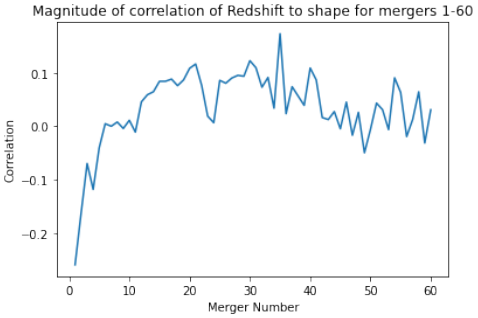
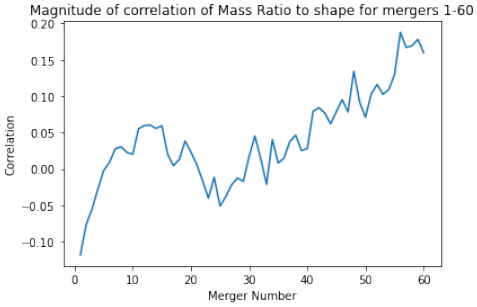
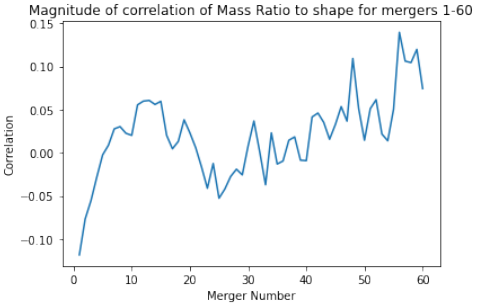
	Missing mergers included and filled with -1	Missing mergers excluded from correlation computation
Star Metallicity		
Star Mass		
Redshift		
Mass Ratio		

Table 3: Correlation plots including (left) and excluding (right)the missing mergers' information for the 60 most major mergers and select important features

These correlation plots also corroborate the findings of the gradient boosting feature importances, with moderately high correlation values for the stellar metallicity and stellar mass of the important mergers and some correlation between the redshift and mass ratio with respect to the shape value.

From the comparison of the two correlation plots for the same feature, we see that for the left column, the correlation values for the least important mergers with respect to the shape are also significant, especially compared to when this information is excluded from correlation computation (right column), where the relative correlations for the same less important mergers with respect to the shape are much weaker. From this, it seems like the fact that merger information is -1 for the less major mergers contains some importance to the shape prediction as the model tries to infer from the fact that there is no merger to make a different shape prediction when compared to galaxies that do contain mergers.

7 Summary

7.1 Gradient Boosting

1. Comparing multiple models on datasets using the first k most major mergers, where $k = 10$, 20, 40 and 60, including and excluding the $z > 4$ mergers, we see that the model with the first 60 most major mergers having z¡4 gave the best R2 and RMSE in predicting the shape of the subhalos. We also notice that there is a bimodality in the prediction plots (Table 2), which seems to be explained when each subhalo is colored by the number of mergers in its history.

2. We take the first 30 most major mergers and build two models - one including and one excluding the number of mergers feature in addition to the features from the 30 most major mergers. We notice that both the models do not perform any differently in predicting the shape (see section 5.1.2), thus refuting the hypothesis that the number of mergers in a subhalo's history contains some information about the shape of the subhalo.

3. We build a model to predict the number of mergers of a subhalo using the information from only the first 7 major mergers of the subhalo. We see that these 7 mergers can predict the number of mergers in the subhalo's history very well, with an R2 of 0.66 on the test data.

4. We then build a model to predict the number of mergers of a subhalo using just the stellar mass of the subhalo. We see that the stellar mass can predict the number of mergers in the subhalo's history accurately, with an R2 of 0.75 on the test data.

5. We then color the predictions plot from our best model in predicting the shape of a subhalo in table 2 (first 60 most major $z < 4$ mergers) by the stellar mass of the subhalo, and observe that the bimodality in predictions can also be explained by the stellar mass of the subhalo (see Figure 9). On observing the figure, we notice that the model predictions seem to be centered around the 1:1 prediction line when the subhalo has a stellar mass $> 10^{0.5}$.

6. We filter the above dataset to select only those subhalos having a stellar mass greater than $10^{0.5}$ and build a model to predict the shape only using these data points. We observe that the predictions for these data points perform much better, with an R2 of 0.43 on the test data and more importantly, on inspecting the prediction plots (see Figure 10), the bimodality in predictions is removed.

7. We hypothesize that for subhalos below our stellar mass threshold of $10^{0.5}$, we are missing some other features about the subhalo in our dataset that is important for predicting their shape, or the fact that due to the resolution of the simulation, the mergers for these low mass galaxies simply do not have enough information about

the mergers to accurately make predictions. To test this, we decide to compare our results from this simulation to that of the IllustrisTNG300 simulation, which has a lower resolution but a larger number of subhalos.

8. On building a model using the top 60 $z < 4$ mergers for all the subhalos in the dataset, we see that the model predictions are very poor, with an R2 of 0.23 on the test data, as compared to using all the model built using all the subhalos from the TNG100 simulation, which had an R2 of 0.37.

9. We next build a model on a sample of the IllustrisTNG300 simulation, by considering only those data points having a stellar mass $> 10^{0.5}$, while replicating the dataset size and shape distribution from the corresponding TNG100 mass thresholded dataset. We see that this model not only performs poorer, having an R2 of 0.37 when compared to an R2 of 0.43 on the TNG100 simulation, but also the prediction plots for the test dataset (see figure 12B), are significantly worse to the corresponding TNG100 prediction plot (see figure 10B). From this, we can almost certainly conclude that the performance of the prediction model is dependent largely on the resolution of the dataset and a good resolution of the mergers, which contains valuable information about the merger history and thus the shape of the subhalos.

10. Looking at the feature importances from the Gradient Boosting model, we observe that the best features for model prediction are the stellar metallicity and mass from the top to major mergers. Other important features include the mass ratio of the top 3 major mergers and the redshift of the most major merger (see Figure 18).

11. On building correlation plots for these select important mergers for the correlation between the shape and each of the features for the top 60 most major mergers, both including and excluding the information contained by the missing mergers, we see that these correlation plots (see Table 3) lend support to the findings of the gradient boosting feature importances, with moderately high correlation values for the stellar metallicity and stellar mass of the important mergers and some correlation between the redshift and mass ratio with respect to the shape value.

12. We also observe the fact that including the missing mergers information increases the correlation between the shape and the least major mergers when compared to when this missing information is excluded. This could potentially explain why the less major mergers are also important to the shape prediction as the model tries to infer whether there exists a merger or not to try and make different predictions for the shape accordingly. This also helps supports the conclusion that the number of mergers in a subhalo's merger history themselves do not have any impact on the shape prediction due to the fact that the correlation between the features of the least major mergers and the shape of the subhalo increases when the missing information is included.

7.2 Neural Networks

1. Our first approach for Neural Networks is to build a MLP (Multilayer Perceptron) network with MSE as our loss function. As we tried different optimizer and network structure (different number of hidden layers), the model always tend to predict the mean of y_train (Figure 13).

2. Predicting the mean of y_train is the easiest way for the model to perform, due to our scaling method and data distribution (Figure 4), it may be hard to find a hidden pattern between features and target hence using a single value as prediction has a fair performance for our model.

3. To make our model smarter, we apply our customized loss function with different parameter settings to penalize the model. However, in all cases it fails to find a pattern.

4. If we penalize not enough (penalty degree is controlled by loss function parameters), as shown in the ADAM optimizer plot (Figure 15), the model indeed avoids the mean but tends to predict a value around it. This shows that the model wants to find a balance between avoiding penalty and predicting mean. Apparently it fails our object: we wish it to find a more sophisticated pattern but it plays dumb.

5. If we penalize too much, as shown in the SGD optimizer plot (Figure 14), the model just gives up learning and no pattern is found.

6. We then color the prediction plot based on number of missing mergers, to see if this situation is caused by missing information. Unfortunately we do not observe any relation either.

7. A few possible changes that could be made in the future to improve model performance:

   (a) Change scaling method. To deal with a great portion of zero values and missing values, we rashly separate them out and assign another values but the assignment is subjective. This may lead to an unforeseeable consequence in Neural Networks training process. A better scaling method would make model easier to learn.

   (b) Increasing training data. As mentioned above, our scaling method is constrained by the number of training data, we cannot filter out those missing values. Data sufficient is the key for both scaling and training.

   (c) Find a better loss function. Three customized loss functions we used are all inherited from one general formula, it is not surprise that they perform similarly. Apply another loss function may improve our situation.

7.3 Symbolic Regression

1. Following the lesson that is learned from Neural Networks, we also apply customized loss function for symbolic regression. Not surprisingly the model still tends to predict the mean and our penalty only makes the prediction drift to a value below it (Figure 16).

2. Apart from this, symbolic regression predictions also presents a bimodality when training with selected features (Figure 17). We also color the prediction based on number of missing mergers but in this case it is not the cause.

3. A few possible changes that could be made in the future to improve model performance:

   (a) Change scaling method and loss function. Similar to Neural Networks, symbolic regression is also sensitive to scaling, hence scaling method and loss function are the key for performance.

   (b) Reduce features before feeding them to symbolic regression. Python gplearn package gives us permission to control the formula complexity, but when it takes too many features it also takes too many noises (here we think < 10 features are sufficient). Perform a proper feature engineering before symbolic regression is always a good choice.

   (c) Try a different symbolic regression package or use customized math operations. As shown before, for safety concern gplearn has redefined some operations and that could be a potential flaw.

8 Conclusion

The main purpose of this project was to understand how mergers impact the final shape of the galaxy and to understand which features of the mergers are important for predicting the shape of the galaxy. Given this goal in mind, by curating a dataset from the merger history trees in a manner that best supports this cause, the following conclusions can be made based on our research:

1. We can almost certainly conclude that it is possible to make reasonable shape predictions only for those galaxies that are well resolved. We see how for the higher resolution TNG100 simulation, models are able to make accurate predictions only when the stellar mass of the galaxy is above a threshold (of $10^{0.5}$ in our case), indicating that these galaxies are well resolved. We then compared the predictions of the TNG100 and the lower resolution TNG300 simulations, using only galaxies above the same mass threshold, while replicating the dataset size and shape distributions. The superior predictions for the TNG100 dataset compared to its TNG300 counterpart further supports our conclusion.

2. Given these well resolved galaxies for which we can make strong shape predictions, we see that the stellar metallicity and stellar mass of the most major mergers have the strongest influence in the prediction of the final shape of the galaxy. We also see that the redshift and mass ratio of these important ratios also influence the shape. The caveat to the techniques used to deduce these important features is the fact that they are correlation based techniques between these features and the final galaxy shape, and thus we cannot make causal statements about which features influence the shape of the galaxy.

In the future, when higher resolution simulations are available, building models on datasets curated from these simulations can be used to further support our hypothesis regarding the impact that the resolution of a galaxy and its mergers have on making accurate shape predictions. Additionally, causal inference techniques can be explored on the well resolved data to make causal statements regarding the properties of mergers that impact the final shape of a galaxy.