Capstone 1: In-depth Analysis

Maia Paddock

During this stage of the analysis, the goal was to explore regression models to predict the redshifts of galaxies in the COMBO-17 dataset using the magnitudes, magnitude errors, photon fluxes, and photon flux errors available in the dataset. In addition to simply finding the best method to predict the redshifts, I also investigated the best regression model to predict the redshifts given only the photon fluxes and photon flux errors. This limitation was investigated because photon fluxes and flux errors are direct measurements from telescopes, whereas magnitudes are calculated values, and a model that could predict redshifts directly from photon fluxes could greatly increase the speed of analysis.

The methods that I tried were Random Forest Regression, Support Vector Regression (SVR), Ridge Regression, Lasso, Linear Regression, and then pipelines with Principal Component Analysis (PCA) feeding into Random Forest Regression or SVR. All of the models used were imported from Scikit-Learn. To analyze each method, I used the r2 score for regressions available from sklearn.metrics. Training and testing data was randomly split using train_test_split and best parameters for some of the models were chosen using GridSearchCV, both from sklearn.model_selection. Finally, throughout the investigation whenever I chose to scale the data I used the Standard Scaler from sklearn.preprocessing, which removes the mean and sets the scaling to unit variance. Scaling was always performed on the training and testing data separately. I not scale each type of data, for example, the magnitudes and fluxes, separately because this would not allow me to scale the training and testing data separately. However, doing so could improve the accuracy of the results.

The data was imported from the galaxies.csv file into a pandas dataframe and split into the seven smaller dataframes that contain the different types of variables. The X data for training and testing the data was created by concatenating the magnitudes, magnitude error, photon fluxes, and the photon flux errors. The y data was the MC_z column from the redshifts dataframe, which contains the redshifts as calculated in the 2004 Wolf et al paper. These are the redshift values that can hopefully be predicted using regression models. For all models tested during this investigation, the data was split into 70% for training and 30% for testing.

The first regression model that was tested was a random forest regressor. One hundred trees were used as a first guess for a basic test of how well the model worked. The r2 score was 0.98270 for this first look, which was already very good. Plotting the feature importance showed that magnitudes were of higher importance to the model than the fluxes. Some of the magnitude errors were also surprisingly important to the model.

Optimizing the parameters for the random forest regressor was the next step. Running GridSearchCV on the number of estimators for the random forest was not finished running after three hours for stepping over 10 possible numbers of estimators, therefore I ran a loop to

manually look at r2 scores for a range of estimators from 0 to 400. From this loop, the best r2 occurred when n_estimators = 220 with an r2=0.98333. I was able to use GridSearchCV to find the best max_depth of the tree, to see if it is better to force the forest to have stumps or short trees or just let it run. Running GridSearchCV on this also took a decently long amount of time, and for each list of max_depth parameters GridSearchCV consistently showed that the best parameter was the highest value in the given list. Therefore, I decided not to limit the max_depth parameter. Finally, I looked at the graphs of three random train test splits at alpha=220 to make sure that the performance was consistent over different randomized training and test sets. The performance was fairly consistent, with r2 values of 0.98232, 0.98233, and 0.97962.

The next step was to examine the random forest without parts of the training data to see how that affected the model. Training the forest without any errors decreased the r2 value to 0.96697, which shows how the errors did not have a large effect on the accuracy of the model. Removing the magnitude and flux errors did increase the feature importance of some of the fluxes, which is expected. Adding magnitude error back into the analysis created a series of steps in the true vs. predicted values graph. This could be due to how the magnitude errors are calculated. The r2 for this test was 0.98244. Using only fluxes and flux errors gave an r2 score of 0.73740, showing how important the magnitudes are in the prediction of the redshifts. With magnitudes and magnitude errors removed, feature importance showed that the flux errors do almost nothing. Strangely, not including flux errors increased the r2 score to 0.74647, better than with errors the errors included. Finally, I included the magnitude subtractions that I had previously calculated. These produced an identical r2 score to using all of the variables without the magnitude subtractions, showing that they did not add any meaningful information to the model. The feature importance of this model still showed magnitudes as having the most influence. This was the conclusion of the investigation of random forest regressors for this problem.

Next I investigated using SVR. Using the full set of variables, the SVR produced an r2 score of 0.93469, not quite as good as the random forest regressor. Interestingly, scaling the variables with standard scaler reduced the r2 score to 0.89276. Using the SVR with just fluxes and flux errors, using a standard scaler increased the r2 to 0.51533 from 0.27066. Even with scaling, the SVR performs significantly worse with just the fluxes than the random forest.

There are many variables in this dataset, therefore I tried making a pipeline with PCA to reduce the dimensionality of the problem before feeding the data into a random forest regressor or SVR. The addition of the PCA decreased the accuracy of the random forest regression to r2 = 0.97184. Using only fluxes, the accuracy of the random forest regression was also reduced from the addition of the PCA. PCA to SVR did not affect the r2 score of the basic SVR at all. Therefore, overall using PCA in a pipeline with random forest or SVR did not help the model.

The final part of my investigation was to test ridge regression, lasso, and linear regression. First, I used GridSearchCV to find the best alpha values for both ridge regression and lasso. For ridge regression, the best alpha was determined to be 0.01, and for lasso the best alpha

was determined to be 1e-5. Increasing the alpha for ridge regression did not change the training score but made the test score decrease rapidly, showing how increasing the alpha above the best alpha simply made the model overfit. Ridge regression performed much better than either lasso or linear regression. Lasso performed the worst, with a best test score of only 0.60471. Linear regression performed the second worst, with a score of 0.65331. Finally, the ridge regression showed a marked improvement over either of the other two methods, with a best test score of 0.81111. An interesting drawback of using lasso, linear regression, and ridge regression is that the predicted redshift values were not all positive. Many predicted values were less than zero, approaching -1 for some of the models. This is non-physical, as redshift values cannot be less than zero for galaxies. The accuracy of these models would definitely be improved if the results were constrained to positive values.

After exploring random forest regressors, SVRs, dimension reduction pipelines, ridge regression, lasso, and linear regression, the best results for this dataset were achieved using the random forest regressors. The SVR came close when using the full dataset, but fell short when using only the fluxes and flux errors. Using PCA for dimension reduction did not add any value to the model. Finally, the ridge regression, lasso, and linear regression models suffered from not being able to limit the output values to only positive values and tended to predict many values both well above and below the redshift range of redshifts that occur in the dataset.