ASTR 596 Final Write-Up

Group sqrt(2): Andrew, Lina, Patrick, Tsung-Han

I. Group Meetings:

We had four Zoom meetings, three of which were ~1hr and one which was 30min. Everyone was present and we discussed getting the project set up, who wanted to try which algorithm, and our progress getting these done. Then, our meetings shifted towards creating and editing a presentation, and subsequently doing a few run-throughs before the final class presentation.

The first meeting was on approximately April 24th to decide upon a proposal to send to Gautham. Andrew proposed a Photometric Redshift project that could satisfy the requirements of Machine Learning and time series with the inclusion of some spectral analysis algorithm. In getting the project set-up, we were all open to ideas, and were most interested by Andrew's pitch of doing photometric redshifts. We liked the idea of each trying a different algorithm almost as a friendly competition between one another to achieve the best results. We agreed the best way to begin on the right foot, given Andrew's experience in the problem, was a tutorial notebook introducing snippets of interacting with the data and an introduction to ANN. We agreed to create a git repository and write a proposal to send to Gautham.

We met next on Tuesday, April 28th to work through the tutorial notebook together. We agreed that our project would be in comparing different methods for photo-z on the same dataset. Patrick would write and analyze a random forest. Tsung-Han agreed to work on a MLP ANN. Lina chose a PCA on the features fed into a clustering algorithm, and Andrew would work on the spectral encoder. We worked on these separately, but pushed our progress and final notebooks to Github for everyone to take a look and get feedback. For example, when Patrick posted his in progress RF to Github, he asked Andrew how good/poor of a performance a MAD score of 0.014 was, and whether the plots looked reasonable, etc.

We met next on Friday, May 5th to plan our presentation and finish analysis. We chose to create a google slide presentation, with each section written by the author of that algorithm, and with an introduction to the problem.

On Wednesday, May 13th we met to finish details on the presentation, cut content to fit to 15min, go through last minute questions, and run through the presentation. On Thursday the 14th we met at 12pm to once more to run through the presentation after last minute tweaks.

II. Work Logs:

Andrew Engel, Spectral Encoder: As soon as the proposal had completed I began searching how to efficiently download spectral data from SDSS, then wrote a wget script on the campus cluster to grab co-added spectra for each galaxy. At the same time I began talking with friends with research in DL and ran my model structure by them. A similar sentiment was achieved: "not sure what will happen, might as well try it." I built a keras architecture and let it run on CC, confirmed my suspicions, and focused my attention on other aspects of the project.

Patrick Aleo, Random Forest (RF) Regressor: Once Andrew uploaded the SDSS dataset, I looked to Gautham's RF code from class as reference. The main reason I chose RF was because I had never used it before (outside of that one class exercise). So I used most of the code to get it up and running. Instead of doing a split/train/test from sklearn, I was careful to use the same splitting we decided on in the Tutorial, so that everyone in the group would be using the same input data for a more direct comparison and analysis. Because the deep learning tutorial/ neural net approach is a categorical one (see line y_train = tf.keras.utils.to_categorical(np.round((180-1)*(Y_cut_1.iloc[0:175000].values[:,0]/0.4),0).astype(int))), I tried to use this in RF, but this approach did not work. RF is only a regression, and this categorical way of splitting the redshifts kept throwing an error I could not solve. That's why my equivalent y_train is simply: data_redshifts.iloc[0:175000].values.
Thus I did a standard regression over the same data, but compensated for trying a categorical approach when plotting the predicted and true distributions of my results, as these results are plotted over 180 redshift bins, just as in the Neural net approach. Once I got preliminary results, I posted to Github and asked people to take a look. I also asked Andrew specific questions on the metric scores I was getting (is my MAD score of 0.014 good?), and said I couldn't get the PIT statistic to work (which I don't think you can get in RF). Regardless, overall the results were a good starting point. When our group started making the presentation, I saw their plots and wanted to make mine of similar quality, so I reran my results, doing touch-ups to make the plots better (like not using a rainbow colormap ;)). Like everyone else, we all helped create and edit the slides, and had several run throughs of our presentation, where if I was unsure about something I was saying, made sure to ask the group, such as "is the reason why RF tends to under predict at high redshift (0.4) b/c we don't have enough examples in our training set?" Lastly, because I feel like every presentation is better with a meme, wanted to incorporate something fun at the end, and ended up spending like 30 min making the Gautham Nearest Neighbors pic.

Lina Florez, PCA + KNN: Following our initial group meeting, I followed both video lectures for week 12 to solidify my understanding of PCA and k-means. In our secondary meeting, after discussing with team members, I realized I had misunderstood what data we had to grab. I had the PCA and k-means framework in place to switch out the data and continue with my analysis, so I thought I only had to make minor corrections to my code. That being said, I wanted to be certain that I was taking the correct approach for my end of the analysis, so I contacted Andrew a bit later. After Zooming with him for fifteen minutes and confirming a good analysis procedure, I set out to adjust my code. Although the entire group was great, Andrew specifically was an awesome resource if I had any questions. The few times I reached out to him directly, he always made time, and was really clear and concise in his explanations. We agreed that I use PCA followed by KNearestNeighborsRegressor to complete my analysis. After that, although it took a while, things went really smoothly. As I worked I tried to make my code as clear and concise as I could, so I only made one commit to the group github. Following my commit, we all met to combine our work into our google presentation. From that point, we all worked equally on the presentation. We worked well as a group and I enjoyed the experience!

Tsung-Han Yeh, MLP+ANN: I didn’t have any particular preference for the methods that we were going to try, so I let everyone decide their preferred methods first and then took the last one: multilayer perceptron. It was over a week ahead before we learned artificial neural networks in the final lecture. I also don’t have any relevant prior experience in artificial neural networks, so it took me some time to fully understand what is and how to use a multilayer perceptron for our project. Fortunately, Andrew's great tutorial provided me with a fast track to the topic. After consulting the online Keras tutorial on tensorflow.org, I found that I can use the Sequential class to stack layers and improve Andrew’s multilayer perceptron model with fewer codes. In addition, my modified model has the flexibility (not hard-coded) to adjust the number of hidden layers and neurons, as well as the dropout rate. Once the program was ready, my mission is to optimize the hyper-parameters and get the best performance of our multilayer perceptron for the performance comparison of different methods used in our project. Among all hyper-parameters, I had changed the number of hidden layers, the number of neurons on each layer, dropout rate, learning rate, batch size, and epoch to see what combination will result in smaller validation loss at a given runtime. Due to the limited computing capability of my old and slow laptop, I did grid search only on one dimension at a time. It took me around three days to finalize what hyper-parameters to use for our project. As shown in our presentation, it was pleasing to see that RF, PCA+KNN, and multilayer perceptron all have similar performance according to our assessing metrics. Finally, I want to say I love the feeling and experience of working on the final project as a group. Besides the scheduled zoom meeting, we all had constant and fun conversation in our slack messaging group. It is amazing to see how the project and the presentation were subdivided and done in a parallel manner. I think everyone in this group all did a great job for our final project.

III. How we feel at the end of the day/ final thoughts:

We are pleased with everyone’s singular and collective efforts, and are happy with our presentation slides and how the talk went during class. We wished we could have successfully built a spectral encoder and better answered questions posed during the presentation, but that leaves improvement for future work.