Thomas Lamont
CS3270 Project 2 - Phase 0

I have a few ideas for improving the initial execution of Project 1.

One thing I never got to implement in Project 1 that Nic had discussed was using pandas' get_dummies method. Applying this to the dataset before running tests has the potential to improve results for a couple of reasons. One reason is that some categories, such as the destination planet and home planet, are nominal, with no intrinsic order, get_dummies helps to separate these categories by value, which will hopefully avoid misleading the model. I am not certain that using get_dummies will improve the results as it increases dimensionality, but it is a straightforward option that I haven't explored yet. It will be interesting to see if something so simple will increase accuracy.

Another thing I plan on doing is increasing the number of hyperparameters being explored for the more complex models. I was using the vLab for Project 1, so I hope that installing Python on my own machine and utilizing as much of my system's processing power as possible will let me run the neural network and new deep learning models with more than the minimum number of hyperparameters. It would be pretty surprising if running more mlp models did not find better results. Some things I want to test with mlp will be the numbers of hidden layers and the max depth; all the hyperparameters were underexplored by just choosing an arbitrary 5 during Project 1. MLP was already our most successful model, and I'm optimistic that throwing more computational power at the problem will give better results.

The last improvement to Project 1 is taking the preprocessing and feature engineering steps further. For preprocessing, some passengers still have missing information, even after filling in cryosleep and VIP status for all the passengers. Currently, missing information is filled in simply with the mean for numerical data or as missing for categorical data at runtime. By checking for other passengers in the groups with missing data, we can fill in some missing values, giving better information to the models for training and testing. Also, creating a groupsize feature could be useful. I also want to try systematically dropping certain columns like HomePlanet, Destination, and Age to see if these values are noise rather than valuable data for determining whether or not a passenger has been teleported. It is possible that dropping some categories will improve accuracy, and hopefully, combined with improving preprocessing steps, there will be a visible improvement in the predictions.

For deep learning, the data, unfortunately, does not lend itself to multi-modal learning. I am constrained by the data available on Kaggle. There are still ways to implement Deep learning with the given data.

The method I will most likely use is a recurrent neural network, this would be a good fit for the Titanic problem because it can understand complex patterns in sequences and context which are present in the problem. In order to fix the issue of vanishing or exploding gradients, a Long Short-Term Memory network may be a good solution. I will have to try different RNN models to find one that works well. I will then incorporate the deep-learning results into the ensemble and make a final prediction using all the models at my disposal.

There is also the name field, which is data that I have always thrown out without a second thought. I think it is possible, though unlikely, that there are hidden patterns in the passenger name. Maybe certain surnames lead to a higher status? The name feature is the only field that has been dropped indiscriminately. I do not think it is likely to be useful. Still, it is possible that the powerful text analysis provided by a recurrent neural network could squeeze some better accuracy out of the models.

No matter what, the first steps will be feature engineering and preprocessing with get_dummies. Then, I will try to tune hyperparameters using more computational power while systematically dropping features to see if they are just noise. Once these steps are done, I can use deep learning in an ensemble with my past models to hopefully increase the accuracy of the predictions as a whole.