Challenge Problem 2

Introduction

For this challenge problem, I took a handful of approaches with various levels of success, including, but not limited to Autogluon models, training a neural net using PyTorch, training a CNN, and attempting classification (spectral clustering). I will focus on two approaches that I developed most thoroughly.

- Autogluon model: tuning hyperparameters
- Neural net using torch. Hyperparameter tuning using GridSearchCV

Implementation

I began working with the Autogluon model. Running it as written in the starter code was fine, but notably did not do well for Advisor 2. I looked into hyperparameter tuning as a means for the autogluon results to train more accurately. I first explored hard-coding specific hyperparameters into the code, for example, the learning rate for NN_TORCH, but then I attempted exploring hyperparameter spaces, as suggested by Copilot, for each model:

```
nn_options = {  # specifies non-default hyperparameter values
      'num_epochs': 100,  # number of training epochs
      'learning_rate': space.Real(1e-4, 1e-2, default=5e-4, log=True),
      'activation': space.Categorical('relu', 'softrelu', 'tanh'),
      'dropout_prob': space.Real(0.0, 0.5, default=0.1)}
```

For my neural net implementation using torch, I created a net with 49 features that pass through three fully connected layers with ReLU activation functions. Then, I apply a sigmoid activation function to the final output. To tune the hyperparameters, I first create the range of values to search over using GridSearchCV. These values were chosen after playing around with the model and seeing how training varied. I did not tune the learning rate because I used a scheduler to automatically scale it down upon plateau.

```
param_grid={
    'batch_size': [10, 50, 100, 500, 1000, 1500],
    'max_epochs': [100, 200, 300]}
```

The following are the optimal hyperparameters for the model for each advisor:

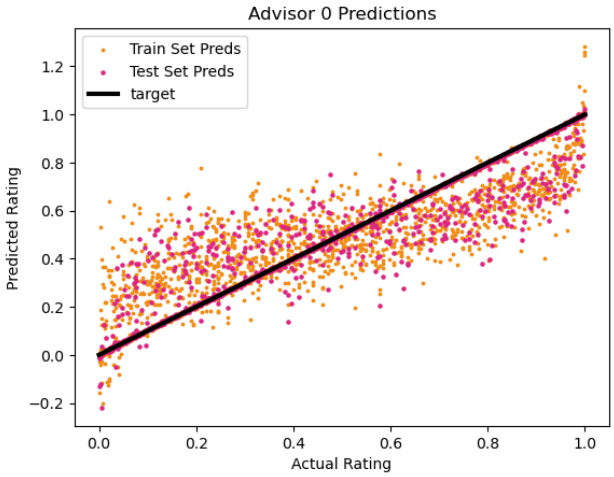
```
#first entry batch size, second entry max epochs
params={0: [50, 200],
        1: [50, 100],
        2: [10, 300],
        3: [10, 300]}
```

Results and Discussion

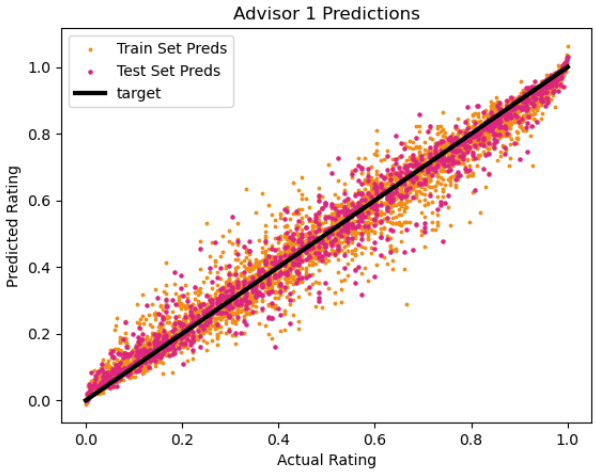
Unfortunately, I do not have the particular code implementation which provided my best results. I looped through all my autogluon models which I ran across many days and found the best model for each advisor. Most of the best models were created about a week prior to writing this report (see comments section for more on this). Once I found the predictions using each autogluon model, I got a score of 0.013, with 84 invalid grids, and a mean advisor score of 0.85. According to my predicted scores, I was supposed to get 100 invalid grids, so I know with a little bit more iteration, I could get that number to 0.

```
Number of valid grids (as predicted): 0
best predicted scores: [0.8073551  0.7908684  0.98004997 0.963675  ]
```

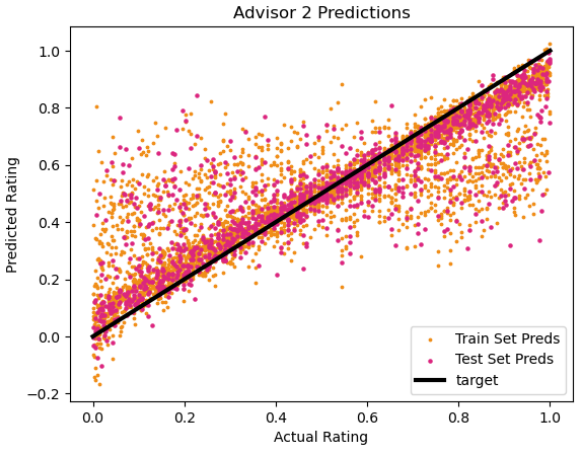
Here are the training and testing plots for each advisor. Advisors 0, 2, and 3 were trained with the same input code, while advisor 1 was different.



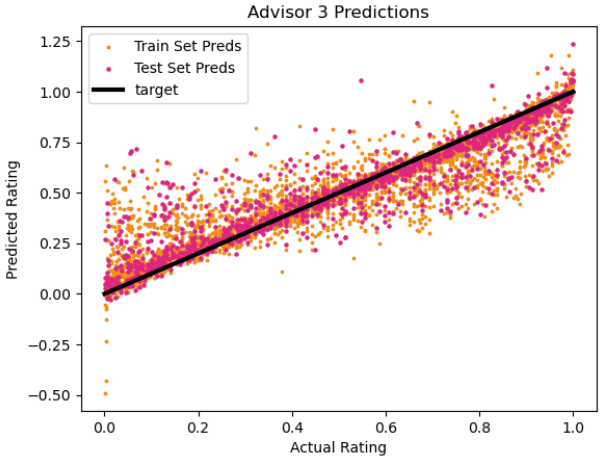

```
Train Set R2 score: 0.865229974020671
Test Set R2 score: 0.8613713310690722
```




```
Train Set R2 score: 0.962542757877789
Test Set R2 score: 0.9618828419234787
```

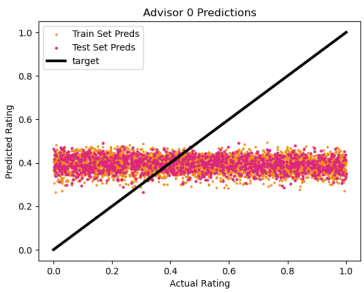



```
Train Set R2 score: 0.720714529943339
Test Set R2 score: 0.7069407517432249
```




```
Train Set R2 score: 0.836547203421984
Test Set R2 score: 0.822807324225539
```

My neural net did work at some point, but unfortunately, it no longer does. When it worked, there were a predicted 10 grids that were valid. I was unable to input my tuned hyperparameters, but I expected that it would have performed better than my best Autogluon results. Here is an image that shows the poor results of training after it stopped working:



Future Work and Comments

One of the biggest lessons I learned from this challenge problem is the importance of documenting iterations and saving versions of my code. It happened a number of times where I had working code, then I would edit it directly. When I introduced new bugs into my code, I was unable to get rid of them, and I was back at step 1. I had some issues with commits in Github, but I brushed them off instead of addressing them. Lesson learned.

Another thing I did which in hindsight I could have avoided is leaning heavily on Copilot and ChatGPT to explain the code to me or write template code. I think it took me about a week or so to process what it is that I was actually doing, but with the AI generated code, I had assumed it would work without understanding the underlying processes.