

Team Meeting

5 Aug 2022 / 3:15 PM / Dual delivery

Attendees

Ate, Stefan, Xavier, Jiadong, Zexi, Ni

Agenda

PLS - Ate

Machine Learning Models - Ni

Spartan - Xavier, Stefan

Deep learning - Zexi

To Do

PLS - Ate

PLS-DA is used for classification. For ERisk data, by setting the threshold value to 0.5, we will have

Maximum accuracy= 0.740437

of components=9

test data: 25% data

Data with missing values are omitted (n=1464)

Need to set the threshold value to the value which maximizes the AUC.

Machine Learning Models - Ni

Current Models:

logistic regression, svm, random forest, stacking, voting

Results:

- No much improvements for the first three classical machine learning models after readjusting the hyperparameters
- AUC for stacking: 84.09% (E-Risk dev)
- AUC for voting: 82.87% (E-Risk dev) - better results compared to the nature paper (72.8%)
- AUC for stacking: 77.24% (BSGS)
- AUC for voting: 82.26% (BSGS) - more generalizable, better results compared to the nature paper (79.6%)

Note:

- Only 130 samples left for BSGS after removing family members and missing values (the missing values are in the predictors)
- Computing complexity issue regarding to gradient boosting

Spartan - Xavier, Stefan

Log in: "ssh username@spartan.hpc.unimelb.edu.au"

Project pathway: "/data/gpfs/projects/punim1257/Group31"

Upload file:

"scp [local file path]

username@spartan.hpc.unimelb.edu.au:/data/gpfs/projects/punim1257/Group31"

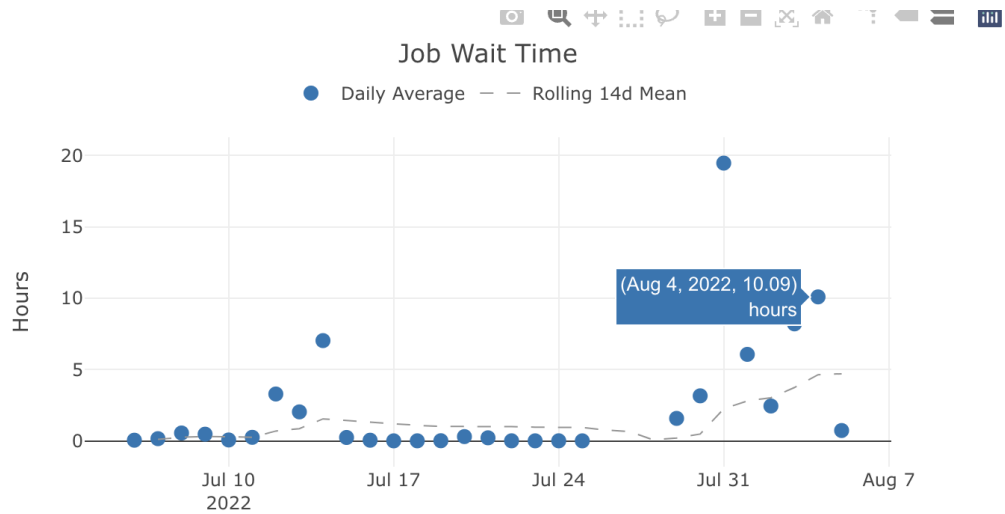
Slurm operation:

Run task: "sbatch 1n1c.slurm"

Display task status: "squeue -u username"

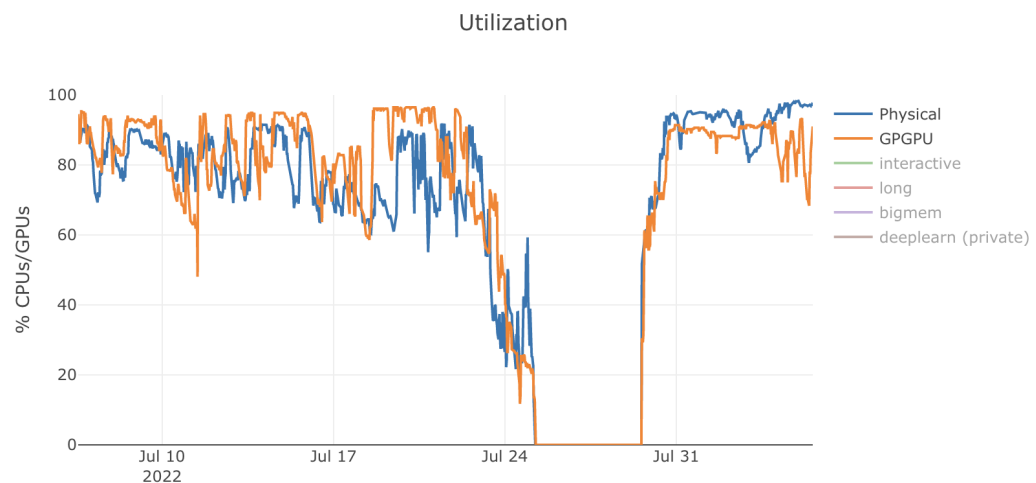
Problem: Long waiting time...

Waiting time for 04/08: 10.09 hours



Current Usage

How busy is Spartan today?



Deep Learning - Zexi

0.89 training accuracy, but 0.5 testing accuracy - overfitting

- Next to do
 - Investigate using AUC criterion



The screenshot shows a Jupyter Notebook interface. The top cell, labeled [14], displays training results for epochs 2400 to 2950. The bottom cell contains Python code for testing the model, and its output shows a correct prediction rate of 118/293.

epoch	loss	accuracy
2400	0.16062527894973755	0.8463114754098361
2450	0.13255243003368378	0.8504098360655737
2500	0.15757770836353302	0.8586065573770492
2550	0.19314362108707428	0.8271857923497268
2600	0.11834997683763504	0.851775956284153
2650	0.11289791017770767	0.8483606557377049
2700	0.125986710190773	0.8483606557377049
2750	0.2254292070865631	0.8647540983606558
2800	0.1278017908334732	0.8818306010928961
2850	0.12691335380077362	0.8592896174863388
2900	0.22744186222553253	0.8490437158469946
2950	0.11420517414808273	0.8907103825136612

```
test_loss = 0.0
correct, total = 0,0
criterion = nn.CrossEntropyLoss()

for data,label in testloader:
    output = model(data)
    for o,l in zip(torch.argmax(output,axis = 1),label):
        if o == l:
            correct += 1
            total += 1

print(f'Correct Predictions: {correct}/{total}')
```

Correct Predictions: 118/293

To Do

1. Check the AUC for other datasets - Ni
 2. Add PLS prior to the training process - Ni, Ate
 3. Ask the client if cares about interpretation? AUC?
 4. Get the Output file of slurm system - Xavier, Stefan
 5. Start writing mpi to implement concurrent processing - Xavier, Stefan
 6. Deep learning (use ray to tune the parameters, decrease number of epochs) - Zexi
- Pytorch lightning for early stopping