

Capstone Individual Reflection

Brandon McCoy
github.com/McCoyBrandon/AI_Capstone

December 7, 2025

I really enjoyed this project. At work I use a lot of tabular data and have had to create risk factor predictions with linear models as part of my work. But it was fun to utilize the programming skills and machine learning knowledge to build something that I could translate into real world use. For example, instead of machine failures, in the case of academia I could utilize it to predict risk factors for students that may drop a class or program. Or I have seen research into toxic course combinations, and the attention based transformers seems like an interesting prospect in trying to create such a prediction model. Because I did a lot of projects in this program utilizing visual or spatial data, but tabular data is more prominent in the business world. So I'm now more confident in being able to utilize my skills gained in a manner that implements AI in a format relevant to my industry.

Starting out the most difficult thing was trying to figure out how to address the project was the class imbalance issue with this dataset. But it turned out to be an interesting challenge and was happy I was able to find literature out there that has many different techniques to address class imbalance, and experimenting with the different types. I hadn't considered that you may be dealing with large amounts of 'normal state' observations and your object is to find the a small percentage of anomaly states. And how that may affect the training. The other projects that I worked on in this program didn't have this issue, nor had I worked with transformers in the way I did here. But I was able to use my knowledge and skills I have obtained during this program to find ways to address it on my own.

It was also unfortunate that I lost my partner in the project about half-way through the semester and they had little contribution in the project. But this ultimately gave me the chance to approach this from a more holistic approach. I've worked in professional team settings for many years at this point, but often I'm the only technical person on the team. So this allowed me to think about how I would approach an analytical machine learning problem in which I only had myself to rely on for the bulk of the technical work. For example, trying to break out my project into modularize code that works together. As I worked to implement different methods and techniques into my starter transformer to making util files that made exploratory data analysis, evaluation, and checking metric logs from my grid search to be helpful in managing my tuning and preparing for sharing results.

There are a few things I would have loved to do if I had more time on this project. Firstly setting up a continued learning integration with drift detection using the existing TabTransformer. The other literature I had included this type of integration for their models and showed an increase of accuracy for their binary 'Failure' and 'NoFailure' classifications. But I think it could help greatly with addressing the recall accuracy of the machine failure types predictions I was doing. And along with this continued integration model, also finding additional datasets that I could utilize to improve on the pretrained model checkpoint I have to see how that continued learning integration would improve on the baseline model I've created. And lastly, working on some level of a front end interface that a user could interact with the inference stage to see the fully realized intention of this model. During the poster presentation I was asked why it was important to do the failure types vs the binary NoFailure/Failure. And being able to visually represent how I picture the tracking of these different failure types in the maintenance and management of a datacenter would be helpful.