I think that machine learning models can greatly improve from feature engineering and make models more accurate. I think that sometimes people will rely to heavily on Neural networks to make predictions, and while sometimes their application could be warranted, the majority of machine learning models don’t really need to be a 5-layer RNN model. When I made a model earlier in the master’s program, I made a model that was attempting to predict car crash severity. Then we chose basic models to do this, but when we would go back and re-train the model, we would put in some time of extracted feature like the day of the weekday, season, and if the time of the crash was in rush hour. The model got 5% more accurate when adding the two extracted features from the data. However, We did drop the timestamp from the model to achieve the accuracy improvement.

I think that sometimes feeding the models vast swaths of information is redundant. Dimensionality reduction paired with feature engineering can also improve various features of the model. If you are able to combine features within initial data transformation and data engineering, you are able to cut down on the amount of features the model is being fed. For example, making ratios, when prepping the data and doing transformations in the table setup can make the model run faster, and cut down on the time of response from the API, even though we could be talking about milliseconds faster. I think that models that use hundreds of parameters can be speed up to perform better when having to do hundreds of computations per second, one good example of this is trading algorithms calculating hundreds of orders per second on one particular stock.

Citations

Krishan. (2023, February 5). *Deep learning and feature engineering*. From Data to Decisions. https://iksinc.wordpress.com/2015/12/18/feature-engineering-and-deep-learning/

*What is Feature Engineering? | Domino Data Science Dictionary*. (n.d.). https://domino.ai/data-science-dictionary/feature-engineering