Reflections on this Project and My Data Science Journey

I liked this project. I wish I had more time to work on it and perfect it. However, to truly perfect it, I would need a strong understanding of Python and Machine Learning.

I would like to get to a point where I understand how the algorithms actually work. The course I am taking doesn't leave much time to really explore what happens behind the scenes. Instead, it's a matter of guess and check. It was interesting to see how Guassian Naive Bayes performed the opposite of the others. When time allows, I will need to figure out why this happened. I tried to use a methodological approach, using standard techniques to improve the model like addressing outliers, multicollinearity, and class imbalance, followed by parameter tuning, bagging and stacking. I would have liked these techniques to improve the models more than they actually did. 66% recall with 75% accuracy is not enough. I would like to meet at least 80 % to 95% recall and accuracy rates, which makes me wonder, am I missing something or is this just a reflection of the dataset?

There is also the issue of time. There were many interesting observations in the EDA, with so many features that could have been tweaked. If I come back to this project in the future, I would like to try binning some of the continuous variables. After looking at variable importance, the variables that reflect customer characterstics like education level or sex were the least important whereas payment status, and payment and bill amounts, were some of the most important. I could try creating a new variable that reflects the change in amount of customer payments over time. Perhaps, a decreasing trend indicates that a customer is more likely to default.

Either way, time is a big constraint. Python is not as intuitive as R, especially with visualizations. Matplotlib is not as intuitive as Plotly with R. I spent quite a bit of time becoming familiar with Matplotlib and even trying to install Plotly with Jupyter Notebook. These kind of issues take away from data science time, and a project on developing a model instead becomes a project on how to code in Python. I understand this is part of the learning process. Once I finish this course, my plan is to study algorithms and data structures. I desperately need to become better at coding so that the vast majority of the time I spend on these projects can be on tuning my model instead of figuring out errors. My python code right now is very susceptible to errors. By copying and pasting the same code and renaming objects as needed, I increase the chance of typos. At this point, I wrote functions where I could and made the decision to repeat code simply because it worked, and I would never have finished this project if I had tried to make all my code 'pretty' and 'dynamic'.

Consequently, choosing which algorithms to use was a high level decision and not one with much thought behind it. I got a tip that this problem would best be solved with ensemble methods and I understood that this is a classification problem (default or did not default). After I finish this course, my plan is to also pursue more knowledge in math for data science. I wish I could understand how knn, gradient boosted trees, and the other algorithms work. Instead of guess and see what happens, I want to be able to know which algorithms would work best for

what type of problems, and what techniques would and wouldn’t work best on the types of data. Ideally, this would reduce the amount of work (and time!) for these kinds of problems. An example of this would be how removing highly correlated variables in guassian naive bayes. It essentially made no difference. This has sparked my curiosity but I will have to postpone delving into it as of now, with time constraints.