# Submission to the DORA Manager Take-Home Assignment

Analyst: Frederick (Fritz) Zuhl

Dec. 16, 2021

**Executive Summary**

**A deep-learning propensity model was created to predict the likelihood of active accounts falling into charge-off status. The major predictors found in the training data include fico\_b, ave\_bal6 and ntriggers. Other active accounts were scored with a probability of future charge-off.**

Hi DORA Team:

Thank you for giving me this opportunity to solve a problem common among all financial institutions-the problem of the rare, yet costly, account charge off.

Discussion of the model

I used Python 3.9 within the IDE of Jupyter notebooks. I saved my code in three forms:

1. The original Jupyter notebook
2. The Python code in text
3. An HTML printout of the Jupyter notebook

For readability, the code is in 2 parts:

**Part1\_IntakePrep:** Code used to review the data, perform common univariate analysis, and feature preparation.

**Part2\_Model:**  Code used to create and fit model, along with one-hot encoding. At its end, it ranks model features in order of importance and score all 'Active' accounts.

The notebooks include all coding used and are narrated with comments and graphs. In addition, I included the active accounts given to me with their probability scores that my model provided. The file is *mrm5\_model\_data\_scored.csv*.

About the modeling approach

I choose to use Python because it has the best algorithm commonly available that I believe to be best for this exercise, Extreme Gradient Boosting, a Deep Learning algorithm, commonly known as XGBoost. The Python implementation has a reputation for having the best features and documentation outside of academia. I know there is a version of XGBoost in R, which your team uses, but I have more experience with Python's implementation.

Notes about the data and some assumptions

**1 Concern about the rim\_age field**

Upon reviewing the data, I noticed something untoward regarding the rim\_age field. When I compare the rim\_age with the number of charge-off per month, I see the rate increasing with the age of the account. This apparently contradicts the behavior mentioned in Melodie's introduction letter that states that new accounts are inherently riskier. I suspected that the rim\_age field may be in 'reverse' in the time window.

When dealing with time durations, a common mistake (which I do it) is to reverse the absolute dates and get the reverse time duration. (Example: in the 42-month window, month 3 in rim\_age would become month 42-3 or month 39 in the field rim\_age\_reverse). The charge-off rates would look more of what I expect then. In anticipation of this error, I built a model with both original rim\_age and the 'reversed' rim\_age. However, in the Jupyter notebook, I only show the output for the model with original rim\_age. I can provide output for rim\_age\_reverse upon request.

**2 Binning Predictors**

In part 1 I 'binned' the some of the continuous values to improve their statistical significance and prevent over-fitting. This is always a tricky yet necessary maneuver with the data. I think if I were to revisit this model, I would like to scrutinize my binning tactics.

**3 Class Imbalance of Target**

This model is a classification model, and its target variable, COS, has a positive rate close to 0.9%, making this dataset highly imbalanced. This is often considered a 'rare-event' problem, and usual modeling approaches fail with so few positives. I attempted to compensate for this imbalance by setting the call to the XGBoost function to give more weight to COS=1 accounts. If I were to revisit this model, I would experiment with other adjustments. It may be that a deep learning model is not the best approach for this problem.

**4 Transaction Data Importance**

If I were to revisit this problem, I would like to take a close look at transaction-level data. As Melodie mentioned in her introductory letter, ACH-related fraud is often preceded by suspicious transactions (online activity, quick withdraws, wild transactions). I understand it would not have been practical to include such data, but transaction-level data is most predictive in these kinds of problems.

**In Closing**

These kinds of models are very tricky and different approaches should be experimented. I am sure you have questions (at least I hope you do). :) So, I look forward to our discussion soon. Again, thank you DORA Team for this opportunity to provide a 'proof-of-concept' approach to predicting charge-offs.

Frederick (Fritz) Zuhl

fzuhl@zuhl.org