**Project: PD model development**

**Course:** STA2546

**Term:** Winter 2022

**Due date:** Friday, 25 February 2022, 11:59PM

**Instructions:**

* This is a group project comprising three to four members per group
* The focus of this project is to provide you with the opportunity to apply the model development framework covered in class.
* The datasets compiled here (with the exception of the last one) are based on those from the Kaggle competition HomeCredit dataset found in <https://www.kaggle.com/c/home-credit-default-risk/data> (available as of November 2021) with alternations to meet the purpose of this course.
* The ultimate goal of the assignment is to build a probability of default model and detail the model development process via three deliverables:
  + Jupyter Notebook files using Python 3.7+ code. You should structure your notebooks as follows:
    - Data extraction and integration 🡪 Extracting data from input files. This notebook should produce three distinct datasets: training, backtesting, and impact analysis. Note: for training and backtesting, conduct a random 80%/20% split of historical stratified on default flag.
    - Data treatment 🡪 This includes the following treatments (be sure to show your results in the code file and in the documentation, either in the main section or in an appendix):
      * Feature engineering.
      * Outlier treatment (use training observations; numeric variables only):
        + Flagging outliers 🡪 Use the Z-score approach with 99.9%ile cutoff.
        + Capping and flooring 🡪 Cap and floor using 1%ile and 99%ile.
        + Apply same logic using training dataset’s results to backtesting and impact analysis datasbase.
      * WOE binning for all variables (use training observations):
        + Numeric 🡪

Use quartiles (i.e., cutoffs are 25%ile, 50%ile, 75%ile, 100%ile). Intervals are left-closed, right-open (e.g. [25%ile, 50%ile); note that pandas’s “qcut()” generates left-open, right-closed intervals, so you can either use “cut()” or your own custom made function. For the maximum value, add 0.1. If there are ties, then simply concatenate bins.

For missing values, create a separate category.

* + - * + Categorical 🡪

Create a missing value category if there are missing values.

For the most common three values, create separate bins.

For the fourth most common value and onwards, concatenate into a catch-all bin and label it as “everything\_else”.

The final deliverable should be the training, backtesting, and impact analysis datasets with model-friendly variables (except for the ID variable, which should be kept).

* + - Variable selection 🡪 This includes the following treatments:
      * Univariate: WOE and IV:
        + Numeric 🡪 Keep variables that have a monotonic WOE trend with he target and the direction is justifiable (e.g., income should have a decreasing WOE trend).
        + Numeric and categorical 🡪 Exclude all variables with IVs less than 3%.
      * Multivariate analysis 🡪 applying VIF iteratively:
        + Run VIF all transformed variables and remove the variable that has the highest VIF if VIF > 5.
        + For ties, keep the variable with the highest IV.
    - Model selection, calibration, and backtesting 🡪 This includes the following items:
      1. Split training dataset into two datasets: train\_train and train\_validate (80%/20% split stratified on the target variable)
      2. Starting with the variable with highest IV, run a logistic regression model on the train\_train dataset and calculate AUROC and KS for both the train\_train and train\_validate datasets.
      3. Once the minimum requirements (see scenario below for details) are met in terms of number of variables and AUROC and KS for both datasets, move on to next step; otherwise, add the variable with the next highest IV and repeat the above two steps.
      4. Concatenate train\_train and train\_validate and calibrate the model such that the average PD is 9% (use gradient descent).
      5. Estimate PDs for testing dataset and calculate AUROC and KS and check if it meets the minimum criteria (see scenario below for details). If not, got back to Step 2 and add another variable.
      6. Apply PD mapping in the following rating system (this will demonstrate that the model is well-calibrated):
         * 🡪 A
         * 🡪 B
         * 🡪 C
         * 🡪 D
         * 🡪 E
         * Apply the following checks:

Min concentration test: No category has less than 5% of observations for training, backtesting, and impact analysis datasets.

Max concentration test: No category has more than 40% of observations for training, backtesting, and impact analysis datasets.

Calibration test: Default rate lies in each of the PD ranges (e.g., for A, default rate must be between 0% and 2.5%; for B, default rate must be between 2.5% and 6%).

* + - Impact analysis 🡪 This includes the following items:
      * AUROC and KS of current vs. proposed PD on entire history dataset (combine training and backtesting datasets).
      * Distributional analysis of current vs. proposed PD on impact analysis dataset.

Note: It is very important that before you send me your code files, you check that when rerun, the results are replicable. It is very important that I get the same results when I run your files on the same data. So, be very mindful about setting paths, setting random seeds, and detailing any additional libraries needed.

* + Class presentation (30 minutes allocated for each team) of the methodology used and key results.
  + Model development report used to detail the development process and provide the key results. To be submitted in MS Word using the template provided.
* Total points available: 50 points :
  + Code (15 points):
    - Correctness and soundness of code: 10 points
    - Clarity of commenting and structure: 5 points
  + Model development report (20 points):
    - Structure and flow of process: 5 points
    - Soundness and comprehensiveness of selection and extraction criteria: 10
    - Soundness of analyses and conclusions: 5
  + Presentation (15 points):
    - Structure and content of presentation slides: 10
    - Quality of delivery: 5
* Grade is out of 50 points

For questions of clarification and challenges in meeting the criteria, feel free to reach out to me.

**Scenario:**

Forefront Mortgage Solutions have hired your company to upgrade their PD model. Default is defined as 90 days delinquency. The current model was developed five years ago by external consultants. It was never validated. You’re given a data dictionary and a set of historical and current datasets. After speaking with the Chief Risk Officer, you’ve agreed upon the following requirements:

**Model requirements:**

* Model is on the application (i.e., APP\_ID) level. Datasets provided are:
  + “application\_bureau\_balance.csv” 🡪 Bureau balance dataset.
  + “application\_bureau\_general.csv” 🡪 Bureau dataset.
  + “application\_history.csv” 🡪 Application historical dataset.
  + “application\_current.csv” 🡪 Application dataset for the recent period (to be used for impact analysis).
  + “data\_dictionary.csv” 🡪 Data dictionary dataset explaining variables.
  + “pd\_current\_model.csv” 🡪 PD values of current model in production for all existing applications.
* All assumptions and decisions during the model development process need to be documented. This includes decisions in extracting and merging the original datasets, treating, and cleaning data to make it model friendly, variable selection criteria, model selection criteria, calibration and mapping, backtesting, and impact analysis.
* Random 80%-20% split of training/testing split
* AUROC >= 70% for both training and backtesting
* KS >= 30% for both training and backtesting
* AUROC training – AUROC backtesting < 5%
* KS training – KS backtesting < 5%
* Calibrate model such that average PD = 9%
* Map probabilities to the following rating system, with a minimum concentration of 5% and maximum concentration of 40% for the training, backtesting, and impact analysis datasets:
  + 🡪 A
  + 🡪 B
  + 🡪 C
  + 🡪 D
  + 🡪 E
* For each class, the default rate must lie within the intervals (this ensures that the model is fairly well-calibrated) for both the training and testing datasets.
* No class should have more than 50% of the observations for the training, testing, and impact datasets.
* Conduct an impact analysis of your new proposed model replacing the existing model
* Final model must be such that:
  + There are 4 to 12 features.
  + At least one bureau variable.
  + At least 50 different PD values.
* Although there is no restriction of which model to use, it is important to be able to justify the relationship between each variable and PD used (e.g., if you use a neural network and income is one of your features and has a non-monotonic relationship with PD, you should provide an explanation on why it could make sense that income may not always have an inverse relationship with PD).

**General tips:**

* ***Code:***
  + For each step in the model development process, use a separate code file (e.g., data extraction, data treatment, variable selection, model selection, …).
  + This will make it easier to review and correct.
  + There might be some cases where you may want to have more than one code file in a step (e.g., extraction may be quite lengthy and so it may make sense to have two or three separate files).
  + When going through the different steps and sub-steps using script-type coding (e.g., Jupyter, SAS, R-Studio) for validation purposes, it is important to strike a balance between efficiently writing code and making it clear for another reader what you are doing (e.g., when doing multiple filtering, this can either be done all at once or can be done in different blocks for readability enhancement).
  + Code commenting and sectioning is important for readability purposes but be careful not to get too carried away and comment every single line or write paragraphs.
  + A good rule-of thumb is to write one-liner comments for each type of operation (e.g., adding a feature) and avoid in-line commenting unless an unusual treatment is done
  + More details can be explained in the model development report document.
  + When feature engineering, it is important to be mindful on how variables are named, especially when naming aggregated variables from multiple sources.
  + Names should be clear enough for a reader to understand what the feature means and potentially which data source it comes from (e.g., BUREAU\_TRANSACTION\_SUM vs. BANK\_TRANSACTION\_SUM), but not too long to hinder practical use (e.g., no more than 32 characters long as imposed in Base SAS).
* ***Presentation:***
  + Aim to have between 15 and 25 slides, with the presentation ranging between 15 and 25 minutes allowing for an additional 5 to 10 minutes of questions.
  + Assume your audience to be a senior model executive.
  + It is a good idea to start with the ultimate results early and then walk the audience through the process. Provide a checklist of all of the key items that need to be accomplished. For any items that cannot be achieved, provide some detail and explanation.
  + Thus, you can include some technical details, but the focus should be on model performance KPIs. You are also encouraged to include visualizations.
  + Try not to overcrowd each slide with points (you can verbally present details if you wish)
  + The emphasis should be on summarizing the model development process from start to finish with key results and conclusions.
* ***Report:***
  + Include your company name and logo.
  + Begin with an executive summary that summarizes the business context, the model used with variables used, and performance results. Provide a checklist of all of the needed criteria given and for any criterion not achieved, provide some detail and justification.
  + For subsequent sections, aim to organize them in a linear way (e.g., extraction, then treatment, then variable selection).
  + Aim to keep extensive lists in appendices.

**Mentorship opportunity:**

The best way to learn in practice (other than self-teaching) is under an apprenticeship model. To get the best benefit of learning how to write a successful model development report and present the results, I encourage you to send me partial drafts of code and the report, as well as ask questions (keep in mind that the goal is not to solve code and modelling problems for you but to explain concepts and provide sanity checks of the presentation and structure of your work). You’ll have a much better chance in both nailing the requirements and getting a more permanent understanding of the process.