Homework 4: Behavior Score to Minimize the CLR

Siqin Yang

Tackle the Problem

To design a procedure to further minimize the CLR, I would focus on building a better behavior score. Currently they are using "bad", defined as 90 days delinquent anytime over the next 6 months, as the output to train the model, and therefore build a binary classifier. However, they still need to calculate CLR in the end, and the binary classifier is only used to output probability to sort by. Therefore, I'm considering building a regression model instead, targeting directly at CLR.

Model Design

- ✧ Model: A supervised model, specifically, a regression model.
- ✧ Input: Same variables used before, made from internal behavioral data.
- ✧ Output: CLR for each observation.
- ✧ Observations:
    - ■ Since I have many past behavior data, I could calculate CLR for each account on a moving level. Besides, accounts' behavior varies during time, given that situations would have been constantly changing, e.g. get a promotion, develop a disease… In order to track latest updates, it would be helpful to calculate the running CLR.
    - ■ I would still use 6 months as the time window. For each account, I would track back 9 time windows, on a step level of 3 months. In this way, I would get 9 observations for each account.
        - ◆ I would track back 9 time windows for most accounts, about which I have data back to 2 years ago. For those newly applied accounts, I would track back for as long as I could (same time window length and same step level).
    - ■ For example, below are all records for account A.

Time Period	Inputs(of each month in the time period)	Output(CLR for the time period)
May 2020 – Oct. 2020	I11, I12, …	O1
Feb. 2020 - July 2020	I21, I22, …	O2
Nov. 2019 - Apr. 2020	I31, I32, …	O3
……	……	……
May 2018 – Oct. 2018	I91, I92, …	O9

Proposed Framework

- ● First I would prepare all data I need. By data wrangling, I could have my inputs according to the time period. And I would use the formulation to calculate CLR for each time period of each account.

  - I would split the date at a ratio of 1:3 to train on 3/4 and test on 1/4.
  - If the number of observations are too big to build model on, I would randomly select 300,000 observations for training, whose CLRs are normally distributed. Accordingly, I would randomly select 100,000 observations for testing.
    - One thing to pay attention to about the test data is information leakage. I would try to avoid selecting observations that not only belongs to the same account with those training ones, but also are in a time period overlapped by those training ones.
    - For example, if time period Feb. 2020 - July 2020 about account *M* is selected as a training observation, I would not select *M*'s observations within time period May 2020 – Oct. 2020 or Nov. 2019 - Apr. 2020.
- With inputs and outputs, I could build the supervised regression model. By maximizing accuracy, I would have my model well-tuned and trained.
- Since I know behavior data for the future 6 month, I could then use them as inputs to predict CLR for each account for the future 6 month.
  - Now I would have predicted CLR for each account. I would standardize CLRs to make them comparable with old ones.
- Then I would sort standardized CLRs on an ascending basis, and find out the n*10%(*X1, X2, …, X10*) value of standardized CLRs(*C1, C2, …, C10*). *X1, X2, …, X10* would be values on my x-axis.
  - *X1, X2, …, X10* are percentiles, and *C1, C2, …, C10* are CLRs.
- For each *Xi*, I could find out all records *Ri* (a set of records) that with a CLR smaller than *Ci*. Then I would calculate the average of their CLRs as *CLRi* for *Xi* of the population.
- With *X*s and *CLR*s, I could then create the plot in the same way as the old one and then compare two lines. Hopefully in this way the cumulative loss ratio at 70% would be smaller.

Alternative Framework

Instead of directly predicting CLR, I would build two regression models, one to predict SL and one to predict SP, based on which I could calculate predicted CLR. Then for each *Xi*, I could find out all records *Ri* (a set of records) that with a CLR smaller than *Ci*. Then I would calculate the sum of their SLs and the sum of their SPs. Sum of *Ri*'s SLs divided by sum of *Ri's* SPs would be *CLRi* for *Xi* of the population.

Other implementation details would be same as the framework above.