Problem

Manually reviewing an applicant for a credit card can be a long and slow process. Multiple factors are taken into account to determine if the applicant will be approved or rejected for credit. Credit score cards are common to use in the financial industry. It uses the personal information of the applicant to predict the probability of defaulting in the future, and credit borrowings. Build a machine learning model that will determine if an applicant is a ‘good’ or ‘bad’ based on the given data in the dataset. The definition of ‘good’ and ‘bad’ is not predefined.

Approach

From the two datasets provided, credit_record and application_record, I decided to start working on the credit_record first. That table contains the applicant's credit data. Months balance = the amount of days past due. Because the table contains both wording and numbers, i replaced the ‘C’, ‘X’, and ‘0’ statues = ‘Good’. The applicant is either up to date on payments (0), paid off (C), or no loan(X). I created a new credit table with ‘Good’ and ‘Bad’ columns. Because an applicant can have more than one credit_record, I grouped the credit table based on their ID. I then merged the two credit tables together. I created a status column that showed how many good and bad records the applicant has. If an applicant had more good accounts then bad accounts, they will be labeled as ‘Good’ (1). If an applicant has more bad accounts than good, they will be labeled as ‘Bad’ (0). The applicant_record was then uploaded, and merged with the credit table.

Findings

- There were a total of 438,510 applicants
- 67% were female.
- Only 4% of all applicants have a college degree, but the majority have some form of college education
- 68% of applicants are married
- 90% live on their own, with 4% still living with parents
- Age, Income, Employed_Years, and Family_Size were the most important variables
- Logistic Regression was the best model to use for this capstone because of it is heavily using binary classification

Recommendations

1. Based on the data provided, I would offer 3 different card types based on the following:

a. Age: will offer different cost savings and card perks depending on age. For example: an applicant in their 20s can be offered a card that will round up their purchases and put the difference in a reserve account. At the end of 30 days, 90 days, 6 months etc, the reserve account can be converted into crypto.
b. Income: an applicant can be offered cards with annual fees and exclusive membership perks like earning airline miles and hotel points
c. family Size: applicants can be offered cards based off family type. For example, a young growing family can be offered cards with college savings account options

Difficulties

During this capstone, I faced a few bumps along each step. Issues I faced in the beginning was that the credit table contained more than one record per ID. I later realized an applicant can have more than one application when applying for a credit card. After merging the tables, there was a lot of cleaning that needed to be done. Lots of Nan values in the data that would have made the machine difficult to predict. Once I got to the preprocessing step, that's when I hit a roadblock. The data was extremely unbalanced, and I was having difficulty balancing the data. Searching on Google for one question can lead to a million answers. After researching and speaking with my mentor, it was determined that the IDs were causing confusion and that was the main reason the data could not get balanced. The two main takeaways I got from preprocessing and modeling: do not touch the test data to resample or balance, and remove IDs especially if they contain numbers as they will cause confusion with the models.