Illinois Statistics Datathon

Ishaan Salaskar
Sai Charan Naka
Nishk Patel
Ethan Mathew

synchrony

Sandia National Laboratories

Table of Contents

Abstract Overview

_Our overall approach to determining the number of charge offs in any given month came down to the utilization of probabilities obtained from survival analysis as well as supervised binary classification system to determine whether or not a charge off occurred. A combination of these two successfully outputted a number of monthly charge offs with accuracy of approximately: __%.

1. General Approach

Let's start with the first set of slides

Pearson's Correlation Table

The majority of our initial steps involved the output of a correlation table. From the beginning we decided to remove correlations that were below a certain threshold and didn't contribute to the chargeoff flag.

One Hot Encoding Categorical Data

For categorical data, our team one-hot-encoded values to make them numeric inputs in our future model. This function was used from pandas and was used on some of the following columns:

- Net_payment_behaviour_tripd
- Bank_fico_buckets_20
- industry

This helped us get a better understanding of how these values coordinated with one another

Feature Engineering

In terms of creating new parameters for our models, our group parsed through the mth_code and snapshot parameter in order to make a "time" parameter which gives the difference in months from snapshot and time the event occurred

We also used the macro data given and added a column in percent gdp change from the previous month to the current month and examined its impact on our charge off rate.

1.
Neural Network

Let's start with the first set of slides

Binary Classification

Our group decided on a supervised classification algorithm to determine whether or not chosen accounts would charge_off based on a certain set of curated features.

Regularization

One possibility we explored was regularization within our loss model. Initially, our models accuracy was shaky so we considered making a custom regularization term that punished false positives by propagating their cost by a constant called lamba and added this to the binary cross entropy loss given by keras. However, it seemed that balancing the data differently was more than effective to yield accurate outputs.

Balancing the Data

An important feature to notice about the data is that over 99% of the data is non charged off so if our algorithm predicts 100% hasn't charged off it would be 99% accurate. We needed a different metric which is why we used confusion matrices to determine the true positive(y_pred = 1, charge_off = 1) and true negative(y_pred = 0, charge_off = 0). We also chose to balance the data to manipulate the proportions of charged off to non charged off data to see if the model would perform better.

Balancing Data(cont)

We trained the network with different balanced data sets and then used a test data set from the unaltered data set to prove it's efficacy on the confusion matrix. The goal of this was to find a proportion that maximized our true positive and true negative.

- 1% –> TP: 99.7%/ TN: 32.4%
- 3% –> TP: 99.7%/ TN 26.7%
- 5% –> TP: 99.7%/ TN: 72.6%
- 10% –> TP: 99.7%/ TN: 62.3%