# CIS 508: Data Mining I

## Project: *Give Me Some Credit*

##### November 30, 2015

### Team 8

### Aditi R Datta

### Navin Krishnakumar

### Graham Landry

### Lin Ma

## Cohort A

Contents

[Introduction 3](#_Toc436665368)

[Problem Definition 3](#_Toc436665369)

[Data Description 3](#_Toc436665370)

[Initial Understanding 3](#_Toc436665371)

[Initial Approach 4](#_Toc436665372)

[Data Pre-processing 4](#_Toc436665373)

[Initial Pre-processing 4](#_Toc436665374)

[Under Sampling of Data 4](#_Toc436665375)

[Over Sampling of Dataset 5](#_Toc436665376)

[Final Approach 5](#_Toc436665377)

[Azure ML 5](#_Toc436665378)

[R Code for GBM Model 6](#_Toc436665379)

[Weighted Averages of Decision Jungle and GBM 6](#_Toc436665380)

[Kaggle Rank 6](#_Toc436665381)

[Conclusion 7](#_Toc436665382)

## Introduction

### Problem Definition

The banking system plays a crucial role in the growth of an economy by providing businesses and households with access to credit used for investment decisions or for large purchases. In order to minimize the risk associated with a loan, banks often times quantify the borrower’s probability of delinquency- or the probability that the borrower incurs financial distress within the next two years. The task at hand for this project is to develop a model that accurately predicts the probability that a potential borrower will experience financial distress in the next two years. With this information, a bank can better manage financial risk/exposure and optimize loan practices for profitability.

### Data Description

The data for this project includes 150,000 records, each with 11 attributes (excluding unique record identifier) pertaining to borrower information, past measurements of delinquency, and other contributing factors). A summary table of each attribute as well as discussion of data quality issues follows below:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Description | Type | Input/Target |
| **SeriousDlqin2yrs** | Borrower experienced 90 days or more delinquency in past 2 years | Binary Flag (Y = 1, N = 0) | Target |
| RevolvingUtilizationOfUnsecuredLines | Percentage of total credit limit used (excluding real estate and installment debt) | Continuous, Percentage | Input |
| Age | Borrower’s age | Discrete, Numerical | Input |
| NumberOfTime30-59DaysPastDueNotWorse | Count of times in past 2 years borrower has been delinquent by 30 days or over but less than 60 days | Discrete, Count | Input |
| DebtRatio | Monthly debt payments, alimony payments, and living costs as percentage of monthly gross income | Continuous, Percentage | Input |
| MonthlyIncome | Borrower’s monthly income | Continuous, Real Number | Input |
| NumberOfOpenCreditLinesAndLoans | Count of borrower’s existing credit lines. This includes installments (e.g. home/car) and traditional lines of credit (e.g. credit cards) | Discrete, Count | Input |
| NumberOfTimes90DaysLate | Count of times borrower has been 90 days or more past due | Discrete, Count | Input |
| NumberRealEstateLoansOrLines | Count of mortgage and other real estate loans. Includes home equity lines of credit | Discrete, Count | Input |
| NumberOfTime60-89DaysPastDueNotWorse | Count of times in past 2 years borrower has been delinquent by 60 days or over but less than 89 days | Discrete, Count | Input |
| NumberOfDependents | Count of borrower’s household dependents | Discrete, Count | Input |

## Initial Understanding

It was initially understood that the task at hand for this project would be classification. The lender (bank) was looking for a predictive model to help determine the likelihood of a potential customer defaulting on their credit card payments within the next two years. Ideally, this model would be used during the application process as an input in determining who to extend credit offers to (and likewise, who to not extend offers to) based upon the attributes listed in the *Data Description* section and what this information said about the applicant’s chance of default.

## Initial Approach

At first several individual classification models were tried such as Decision Trees (all models in SPSS), SVM, KNN, RBF, Decision Jungles (in Azure ML), and Two Class Decision Forest (in Azure ML). We alsoThe results of these individual models were not promising, but we hit a high ground with the Decision Jungle model which ranked 167 on Kaggle. The rankings for a few other models are provided below:



Despite ensembling Decision Trees and Neural Networks in SPSS, there was not a significant improvement in our Kaggle Rank. At this point, we were lead to believe that improvements in rank would come as a result of better pre-processing of the data rather than a difference in modeling techniques. We decided to pursue methods of preprocessing the data, namely “massaging” data of null values, smoothing outliers, and trying under-sampling and oversampling of our dataset. We were dealing with a predominantly unbalanced dataset, with data largely supporting conditions where the individual was not bankrupt in two years. The original data was characterized by a 1:14 ratio of minority to majority class.

## Data Pre-processing

### Initial Pre-processing

The training data for *Give Me Some Credit* contains 150,000 records that include a class description (Yes = 1 or No = 0) indicating whether a person was seriously delinquent on a credit line in the past two years along with 10 explanatory variables. After trying out different models we concluded that since the data is messy and unbalanced, we had to pre-process the data to achieve better results

We applied custom substituted value from Azure ML’s clean missing data option and tried out decision boosted Tree as it was our best model at that point. But we were still at Rank 440. Upon further analysis, we found that Azure ML doesn’t treat well for values that has missing value coded as “NA” and hence we had to factor for those. We had also seen that values 96 and 98 in variables such as “NumberOfTime3059DaysPastDueNotWorse”,“NumberOfTimes90DaysLate”,”NumberOfTime6089DaysPastDueNotWorse” were actually skewing our results as they were clear outliers and hence our task was to find out a perfect coded values for “96”,”98” and “NA”. We tried out lot of values such as substituting mean of the values, zeroes etc. but we cracked a near optimal replacement when we substituted the outliers with a coded value of “-1”.

### Under Sampling of Data

Here we tried to under sample with Kohonen method. We ran a Kohonen model with SPSS and then only took clusters where the minority class was fairly represented and eliminated all other records where there was a clear favor towards the majority class. This was the technique we employed but here to under sample we could have just reduced the minority class by deleting the observation with random sampling. We now had a dataset with fair representation of minority class with the ratio of the dataset now being 1:8 (Minority: Majority). The results of Decision Jungle with under sampling were not that satisfactory. We achieved a Kaggle rank of 441 with this technique. The AUC value was 0.8636. With this result we decided to do away with under sampled dataset.

### Over Sampling of Dataset

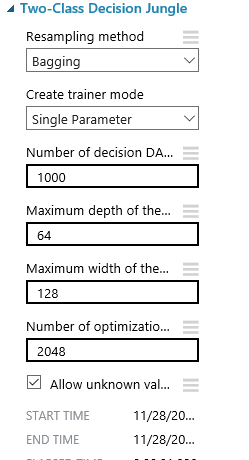
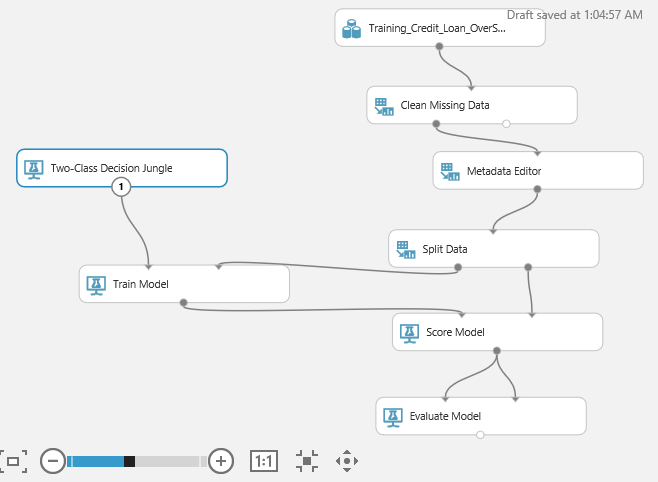
Under sampling of our dataset did not help much with the results and hence the other viable option was to Oversample our dataset with the minority class and hence we increased the size of minority class by a factor of 3. We achieved a best result up until this point (Rank – 95) by over sampling our dataset and the AUC value for the above was 0.8673. We decided to go further in our analysis by again increasing the size of the Minority class by a factor of 2. Hence after this we have in total increased the size of the Minority class by a factor of 5. Since we achieved better results by oversampling our dataset, we decided to try out further over sampling and check if we can better our results and hence the minority class was further increased by a factor of 4, totaling of a factor of 9 for the minority class. With this Oversampled dataset our model’s performance deteriorated and our rankings dropped to 267 with an AUC value of 0.8648. Hence we had no incentive in increasing the majority class by a factor of 9 and with these results we have hit on our best preprocessed dataset. Here’s the summary of the preprocessed dataset

1. Replaced outliers such as “96” and “98” and missing values “NA” with a coded value of “-1”
2. Over sampling of our minority class data by a Factor of 5 so as to negate the effect of unbalanced dataset

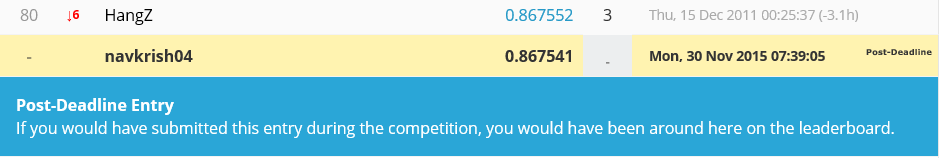
## Final Approach

### Azure ML

Within Azure ML, the Two-Class Decision Jungle was used after accounting for missing data as explained above and using the Metadata Editor to exclude the *Index* column. The data was split 0.75 for training and 0.25 for validation, and the parameters used in the model were defined per the below screenshot:



Here is a screenshot of the Kaggle rank for the Two-Class Decision Jungle by itself:

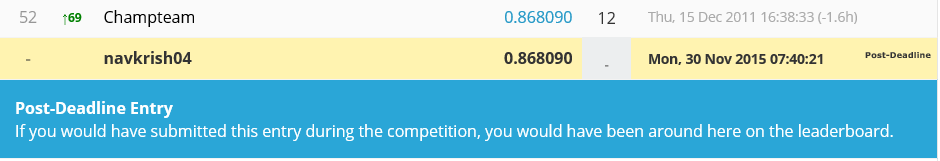


### R Code for GBM Model

In order to try other software’s to achieve better results we tried R. Since we had already tried other general model with SPSS and Azure ML, we wanted to try out other models specifically with R and hence tried Gradient Boosting Machine (GBM) with our Over Sampled dataset. We were able to achieve a rank of 52 with our very simple R code. This became our single best model up until this point.

|  |
| --- |
| GB <- gbm(formula = Train$SeriousDlqin2yrs ~ ., distribution = "adaboost",  data = Train, n.trees = 500, interaction.depth = 10, shrinkage = 0.05,  bag.fraction = 0.5, cv.folds = 3, keep.data = FALSE, Verbose = TRUE)  A gradient boosted model with adaboost loss function.  1000 iterations were performed.  The best cross-validation iteration was 1000.  > best.iter <- gbm.perf(GB,method="cv")  > summary(GB, n.trees = best.iter)  Prediction\_GB <- predict(GB, Testing, type="response")  Using 1000 trees...  > write.csv (Prediction\_GB, file = "predict.csv", row. Names = FALSE) |

The Kaggle rank for the above R Model individually:

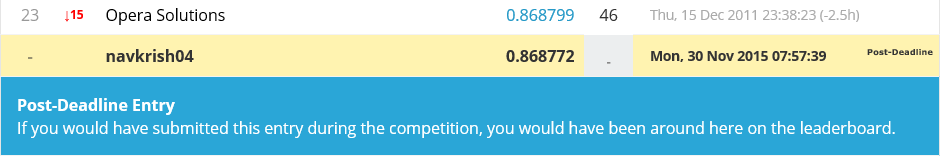


### Weighted Averages of Decision Jungle and GBM

To better our rankings, we decided to use weighted averages for our best model from Azure (Decision Jungle) and R (GBM). We used weights (5 for GBM model and 3 for Decision Jungle) and averaged the probabilities. This was in line with ensemble of ensemble technique in SPSS. The Weighted averages worked perfect for us as we achieved our final best rank of 23 with AUC value of 0.8687.

### Kaggle Rank

After submitting the ensemble model of the Azure ML Two-Class Decision Jungle and the R Gradient Boosting Machine, we were able to achieve a Kaggle rank of 23:



## Conclusion

This project was a valuable experience in learning the end-to-end process of building a predictive model- and not only building a predictive model, but one that is quite effective (as demonstrated by the high Kaggle rank). At first, there was a significant learning curve as it applied to understanding the different types of models and the optimal situations for using each. However, as we spent more time learning in lecture and completing assignments we began to develop insights into the different ways to structure a model- which greatly advanced our submission ranks on Kaggle. One key lesson learned was the importance in both understanding data as well as pre-processing the data so that it can perform best in training the different types of models. We experienced first-hand that 80% of the battle (and likewise 80% of your time) can be consumed with understanding best how to clean the data and apply it in such a way as to optimize the model. The adage “bad input breeds bad output” certainly held its ground during this project.