Executive Report

Nicholas Lewis, Sam Sheth, and Fareena Imamat

5/7/2020

Overview

There is always a risk for credit card companies that a particular client defaults on his or her payments. In this study, we examine past bills, past payments, and select demographic characteristics for 24,518 clients. Our goal is to determine the best model that uses these variables to predict the likelihood that clients in a separate dataset will default on payments in the upcoming month, October 2005. We begin by processing and cleaning the data and performing exploratory data analysis in order to get a better sense of the data and examine relationships between variables. Then, we use our cleaned dataset to begin building models. We consider a variety of models with varying degrees of accuracy in predicting defaults, and we ultimately present the following four within this report: logistic regression, linear discriminant analysis (LDA), principal components analysis in LDA, and random forests. Our best performing model is the random forests model, and we are confident that it will be of significant value to credit card companies as they learn more about their clients and work to develop the best possible experience for them.

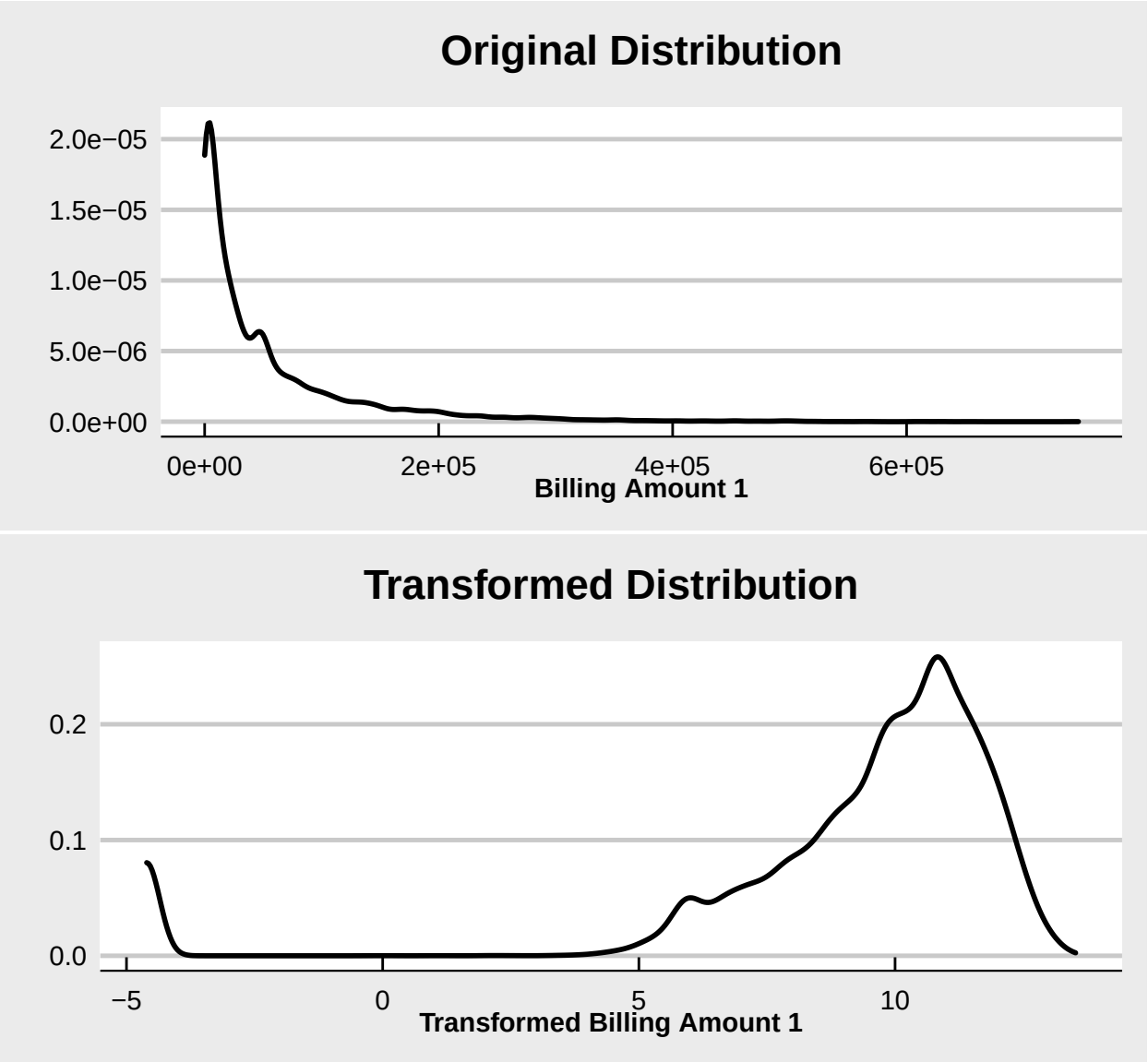
Data Processing

We first partitioned the data into two distinct sets: a training set and a testing set. The training data was used for all EDA and model building. Reported performance measures were calculated on the testing data.

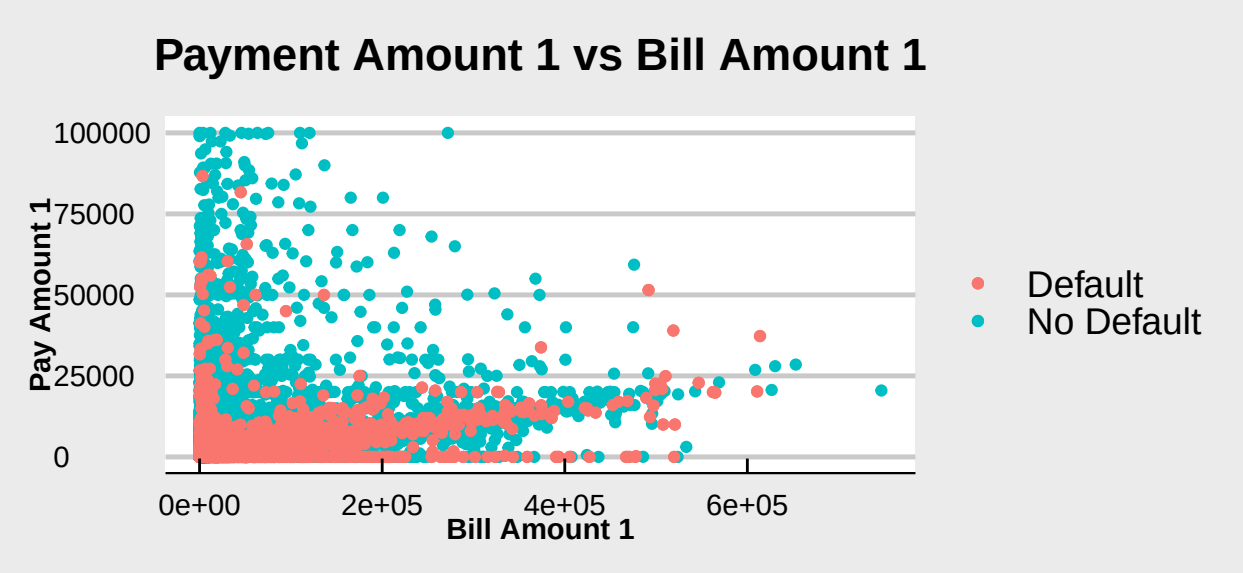
Exploratory Data Analysis (EDA) and initial modelling attempts quickly illustrate the need for new features and the transformation of existing features. First, simple density plots of the features `Billing Amount 1`,..., `Billing Amount 6` and `Payment Amount 1`,..., `Payment Amount 6` as well as `age` and `balance limit` are clearly not normally distributed. Logarithmic transformations are taken in order to remedy this issue. There are several cases in which this transformation is problematic, however, since both the billing features and payment features contain zero values and bill amounts may also be negative. A slightly more complex version of this transformation is detailed below.

$$F(x) = \begin{cases} \log(x) & \text{if } x > 0 \\ \log .01 & \text{if } x = 0 \\ -\log|x| & \text{if } x < 0 \end{cases} \tag{1}$$

The transformation is illustrated graphically below for `Billing Amount 1`. All plots illustrating this transformation are included in Appendix A. The resulting distributions are far from perfectly normal even after the transformation, which may present issues with modelling techniques that assume normality.



Next, we create variables denoting the percent paid for each month. Intuitively, this ratio may be more influential than the billing and payment amount are separately. The plot below, as well as the rest of the family of plots included in Appendix B, illustrates that there is some sort of relationship between these two, but that observations who do not default do not adhere to this relationship very strongly. The noise extends above the top of this graph for those who do not default.

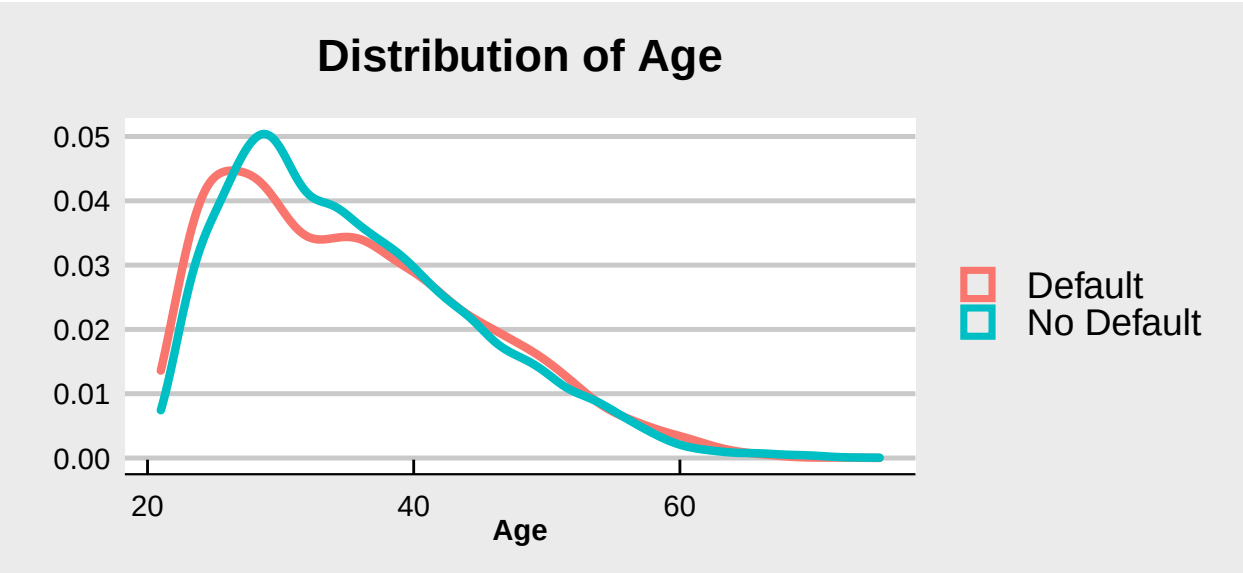


Again, observations which have a transformed billing amount of zero present a challenge. This is dealt with through the logic below.

$$F(x) = \begin{cases} \frac{\log(\text{Amount Paid})}{\log(\text{Amount Billed})} & \text{if } x = 0 \\ \frac{\log(\text{Amount Paid})}{\log(.01)} & \text{otherwise} \end{cases} \tag{2}$$

Extra features are also created for the log transformation of the mean billing amount over the six months and the log transformation of the mean payment amount. This is done to provide a more concise representation of the billing and payment info if needed. The logic for the transformation of negative and zero amounts applies to this transformation as well.

Then, looking at the density plots of age by the value of the default value shows that there are age domains that are more likely to contain defaulting observations than others. To potentially reinforce these distinctions, we create indicator variables for the inclusion of an observation in the ranges [0, 27), [27, 40), [40, 55), and [55, inf).



We use a similar methodology to create indicator variables for high and low pay (divided at \$2980) and for high and low billing limits (divided at $\$1.25 * 10^5$). Similar plots for these variables are included in Appendix C.

Modeling

Logistic Regression

The default variable is a binary, categorical variable, with the two categories being "defaulting" or "not defaulting." We begin our modeling process by considering the logistic model, which will consider the predictor variables for a client and report the probability that the response variable (default) falls into a particular class ("defaulting" or "not defaulting"). The probability outputted is the probability of "defaulting" from which we can determine the probability of "not defaulting" since the probabilities must add up to 1. If the provided probability of defaulting is greater than 0.5, we predict that the corresponding client will default ("yes"). Otherwise, we predict that the client will not default ("no").

From our initial attempt at logistic regression, we learn that only 19 of our 36 predictor variables are significant in predicting default in this model (Appendix D). An important observation is that of the "percentage paid" variables, only perc_paid1, the percentage of the bill paid in September (the month right before the one we are interested in), is significant. This supports our initial view that since many people live paycheck to paycheck, there is uncertainty from month to month when it comes to defaults, so only the month closest to our focus (October) could be significant in predicting the probability of default. However, the amount paid in each month prior to October is significant. However, since the amount paid relative to the bill amount is not significant in all but September, we suspect that there is a confounding variable that is unaccounted for: expenses paid in cash to lower the bill amount and avoid worsening limit_bal (credit).

The "age" variable is also not significant. This makes sense because financial insecurity affects people of all ages. There are plenty of younger people who are richer than older people, and vice versa, so age is not helpful in predicting defaults. This initial model results in a log-loss of 1.83967, so we run a new logistic regression model with only the significant variables included, but we only improve the log-loss to 1.617169. The logistic regression models we developed provided useful information on the significance of our predictor variables. However, they did not perform well, likely because they are still considering too many predictor variables, so we now move on to linear discriminant analysis of our data.

Linear Discriminant Analysis (LDA)

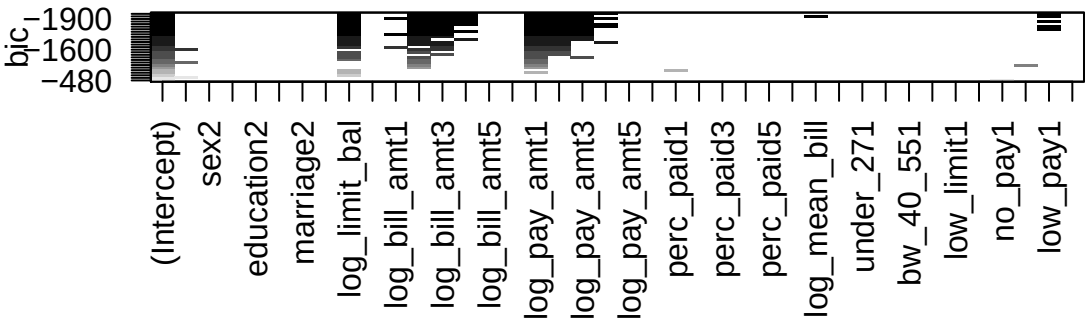
Since we suspect that we are utilizing too many variables, we begin by implementing variable selection procedures (Appendix 2). We attempt to perform forward, backward, and mixed selection using the stepAIC function, but stepAIC would not work because the AIC is negative infinity for our model. This means that stepAIC would lead to overfitted models, which is why the function would not run properly. We try the all-subsets selection procedure, which is successful in returning the following variables as significant: log_limit_bal, log_bill_amt2, log_bill_amt3, log_pay_amt1, and log_pay_amt2. This selection provides further evidence that months closest to October are important in predicting if a client will default. The 1, 2, and 3, in these variables refers to September, August, and July. Now that we have narrowed our significant variables from 19 to 5, we move forward with linear discriminant analysis.

Linear discriminant analysis (LDA) is apt in pattern recognition. It finds a linear combination of variables that separates our two classes ("defaulting" and "not defaulting") in order to make the best predictions. This method of developing a predictive model resulted in a log-loss of 0.4656489, which is much better than the log-loss of 1.617169 that we got from logistic regression.

All-subsets variable selection

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 3 linear dependencies found

## Reordering variables and trying again:
```



Optimal variables: log_limit_bal, log_bill_amt2, log_bill_amt3, log_pay_amt1, log_pay_amt2)

Using optimal variables from all-subsets selection

Quadratic Discriminant Analysis (QDA)

As part of a deeper analysis, we briefly considered quadratic discriminant analysis (QDA), which creates a non-linear division between our two classes rather than the linear division that LDA creates. However, QDA resulted in a worse log-loss of 0.7679433. This tells us that there is a multi-dimensional linear boundary that separates the data into the two classes, "defaulting" and "not defaulting." We now move on to principal components analysis in LDA to see if we can develop a better predictive model.

LDA with Principal Component Analysis (PCA)

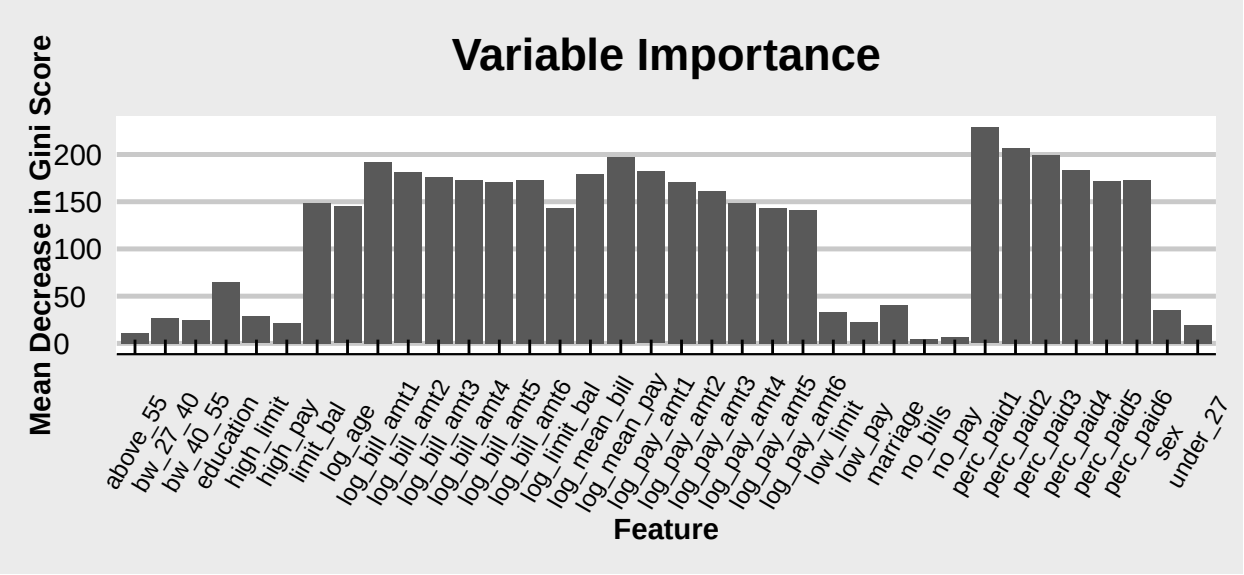
We now attempt to create an LDA model that utilizes Principal Component Analysis (PCA). More information is available on PCA in Appendix -.We perform PCA on the `bill_amt` family and `pay_amt` family, as originally presented in the data. Because the best subset selection process in the previous LDA model did not select any of the indicator variables created during the data processing stage, we build this LDA model on the PCA components and the original qualitative variables `age`, `sex`, and `education`.

This model produces predictions on the test data with a log loss score of 0.4993. However, a closer look at the results reveals that all observations are predicted to not default. This is the same behavior as the trivial model. Therefore, although the log loss is better than the trivial model, it is only because the probabilities are smaller on average. In other words, based off this model we are more confident that all observations will not default. In some sense, that actually makes this model less useful than the trivial model.

Random Forest

We first used the `train()` function supplied by the `caret` package to obtain the optimal values for the number of features considered at each split of each tree `m` and the number of trees `B`. This process was fairly computationally expensive, but produced a final model with $m = 2$ and $B = 500$.

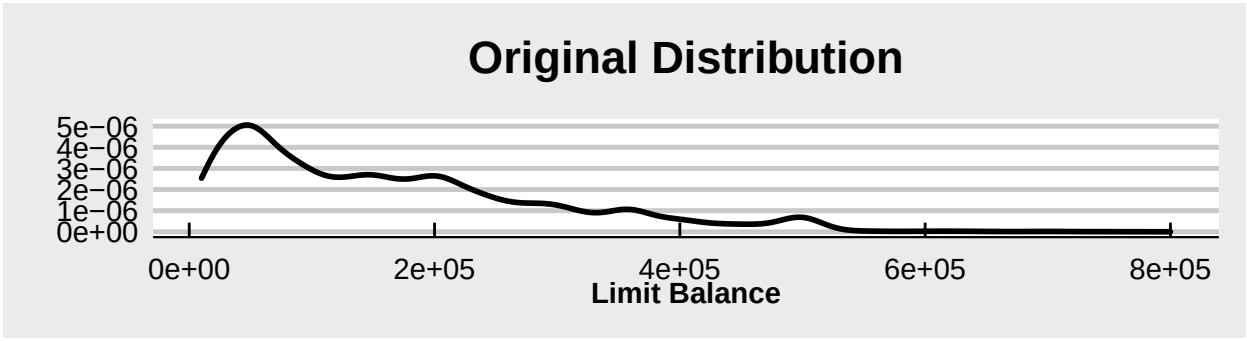
At this point, the log loss is fairly impressive, 0.4607 to be exact. However, we see that many of the variables created during the data processes are not important to the model (as displayed below). A low importance means that the feature was rarely used to make decisions in the forest because it was not a powerful predictor. However, because of the random nature, these low importance features will still be chosen at times. Removing them may allow more predictive features to be used more often, effectively increasing the power of our model.

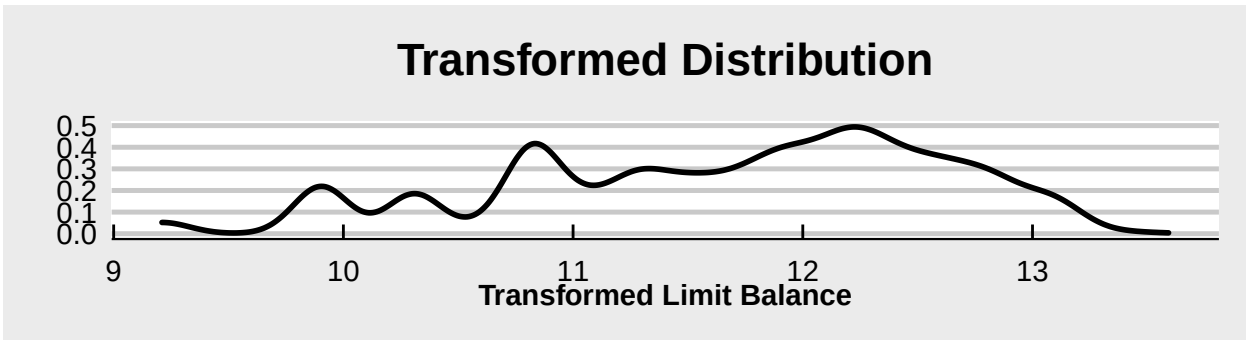


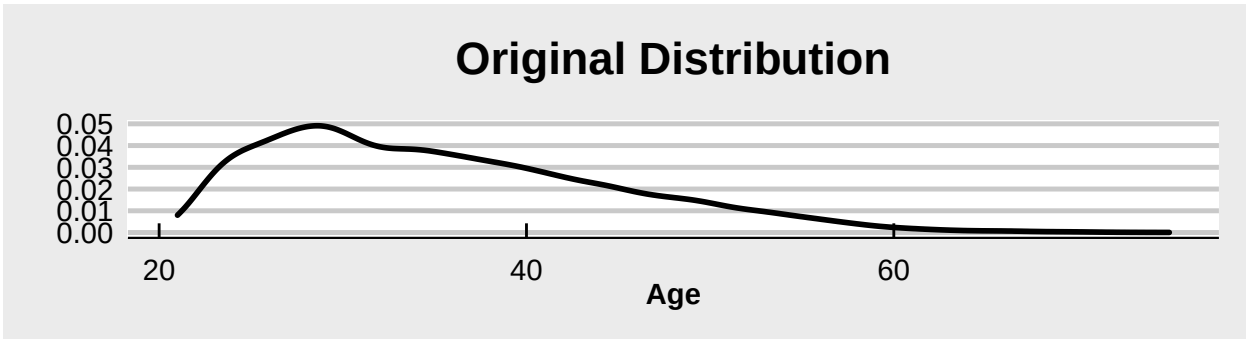
We remove the lowest variables, rebuild the model with $m = 2$ and $B = 500$. By doing this, we now obtain a log loss score of 0.4548.

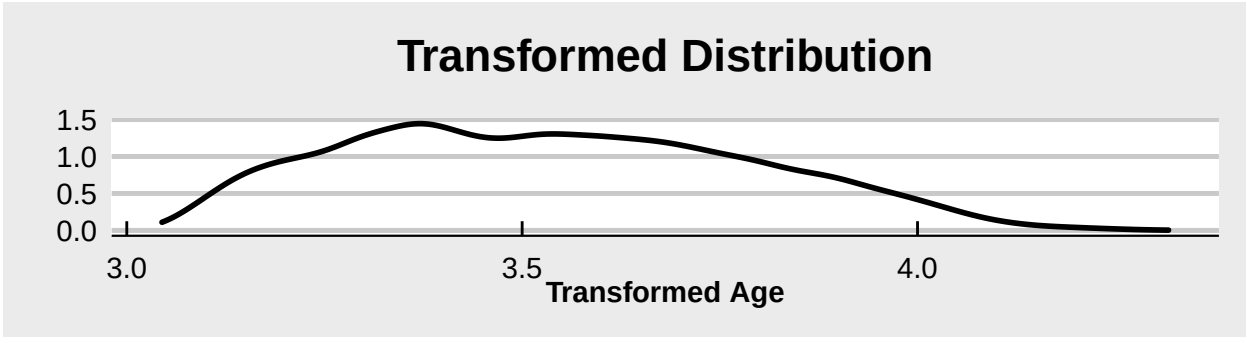
Appendix A

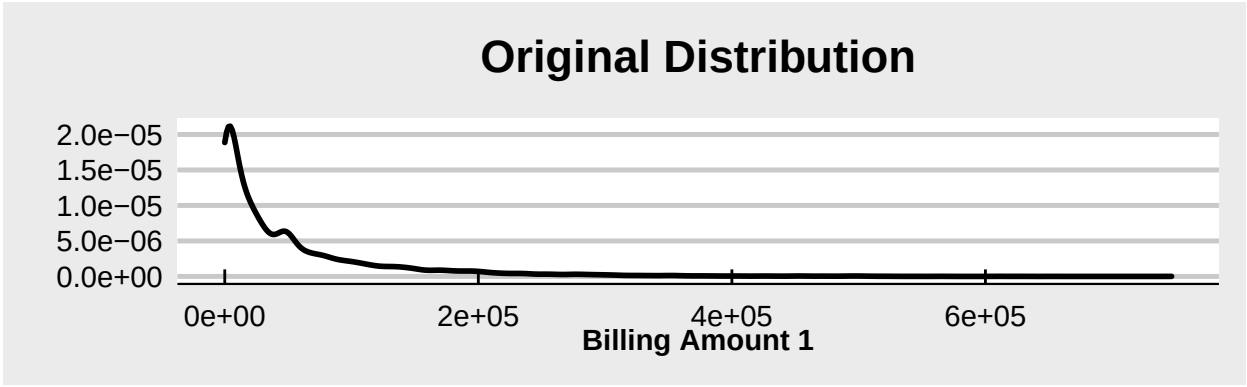
The following are plots of the variables transformed according to equation (1) before and after the transformation.

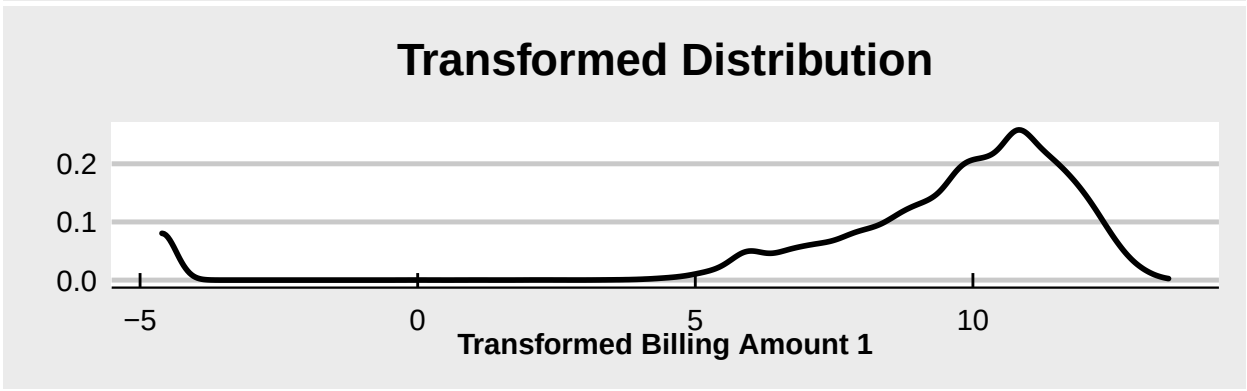


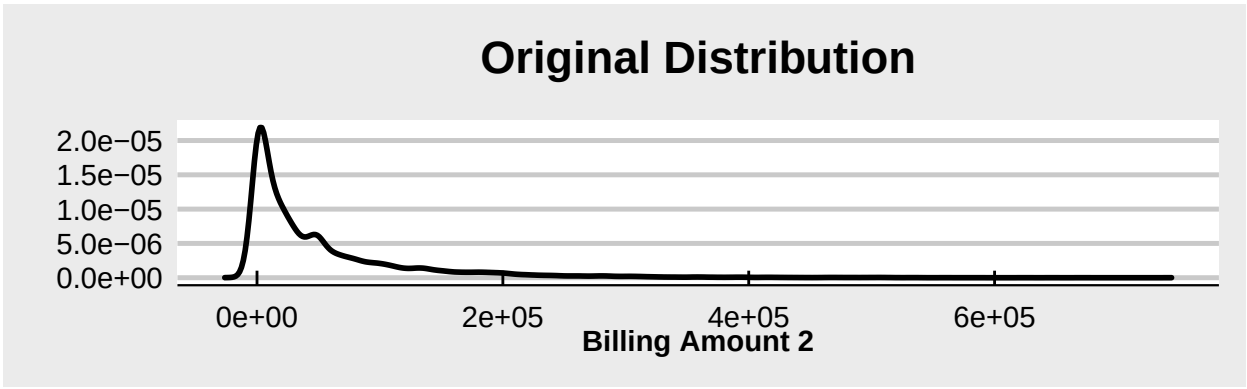


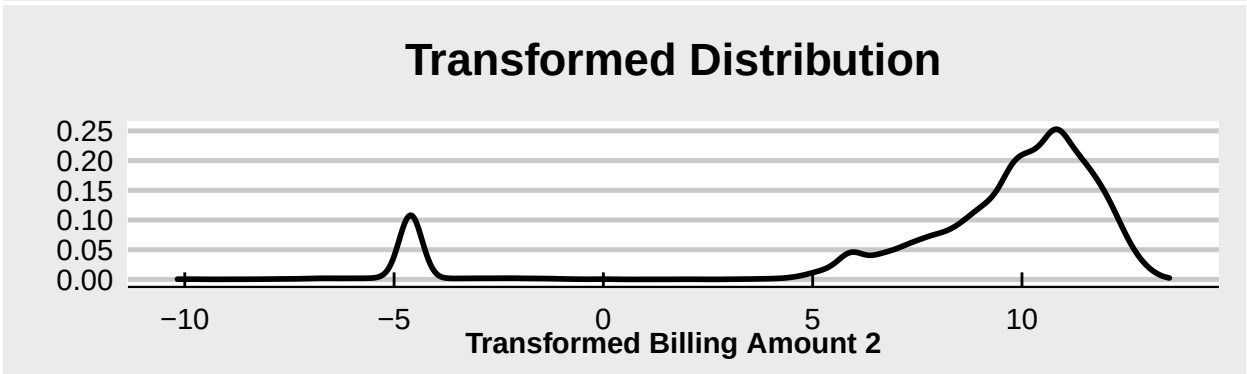


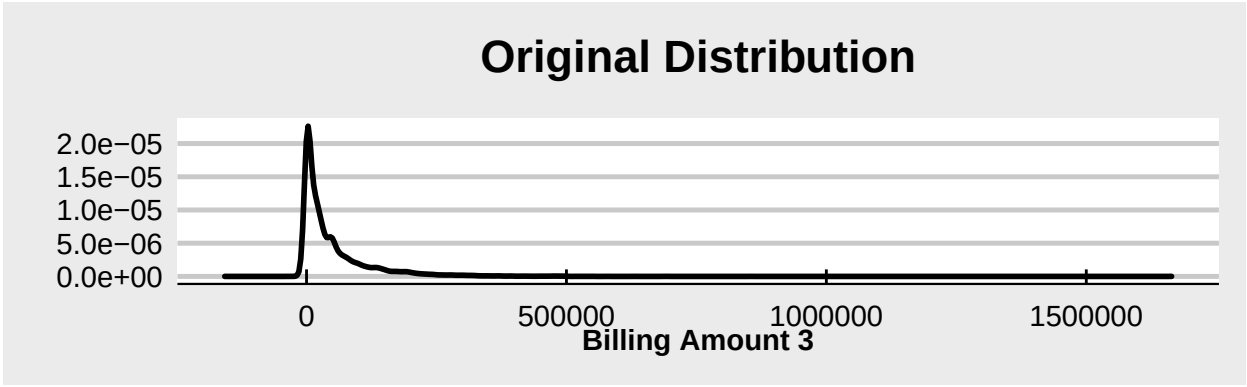


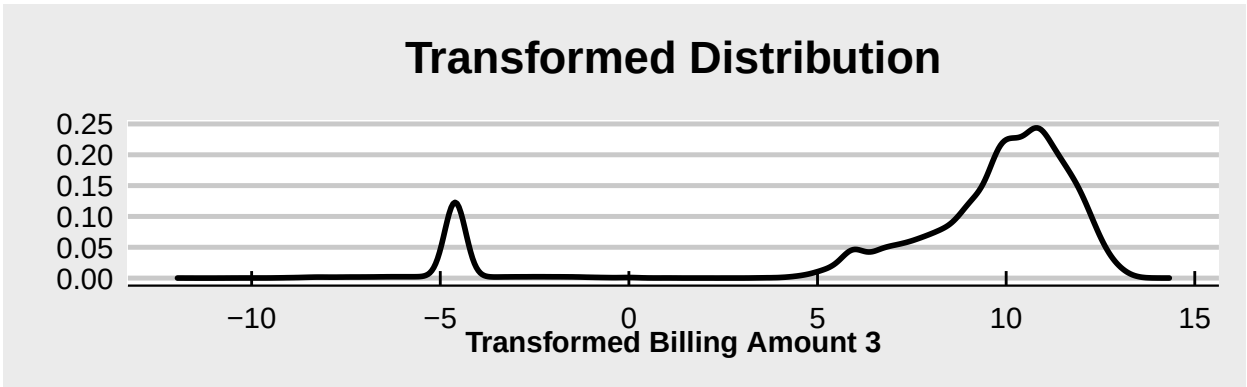

Original Distribution
2.0e−05
1.5e−05
1.0e−05
5.0e−06
0.0e+00
0e+00
2e+05
4e+05
6e+05
Billing Amount 1

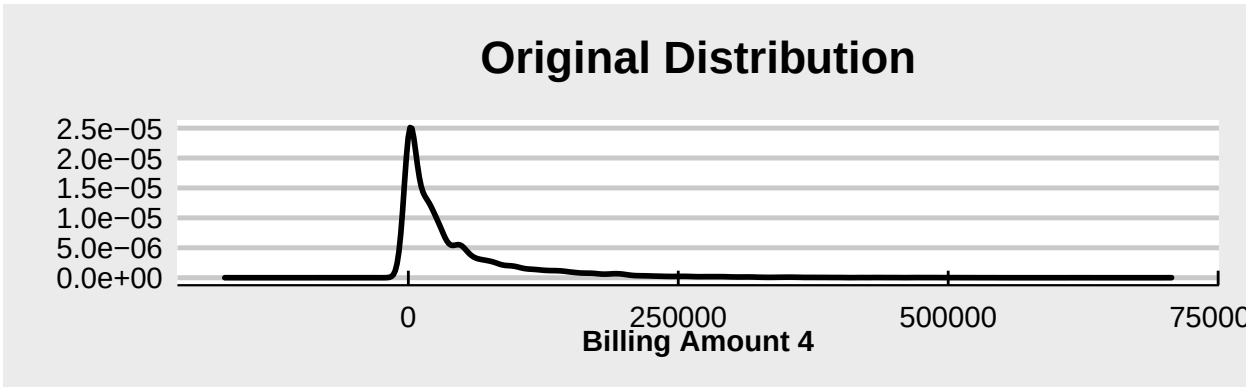

Transformed Distribution
0.2
0.1
0.0
−5
0
5
10
Transformed Billing Amount 1

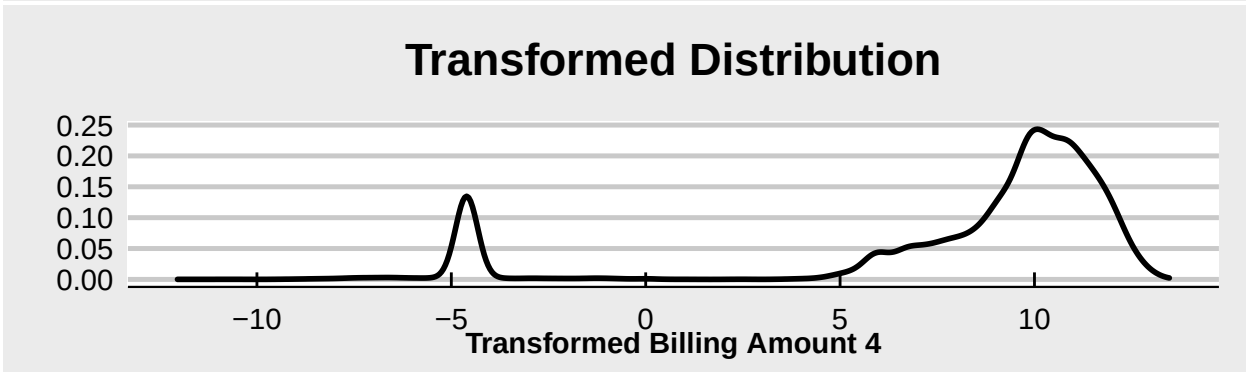

Original Distribution
2.0e−05
1.5e−05
1.0e−05
5.0e−06
0.0e+00
0e+00
2e+05
4e+05
6e+05
Billing Amount 2

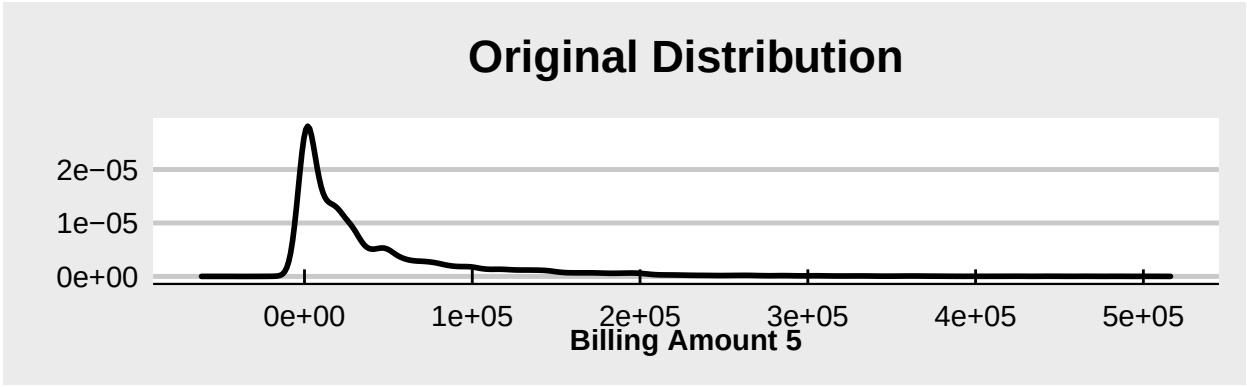

Transformed Distribution
0.25
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
10
Transformed Billing Amount 2

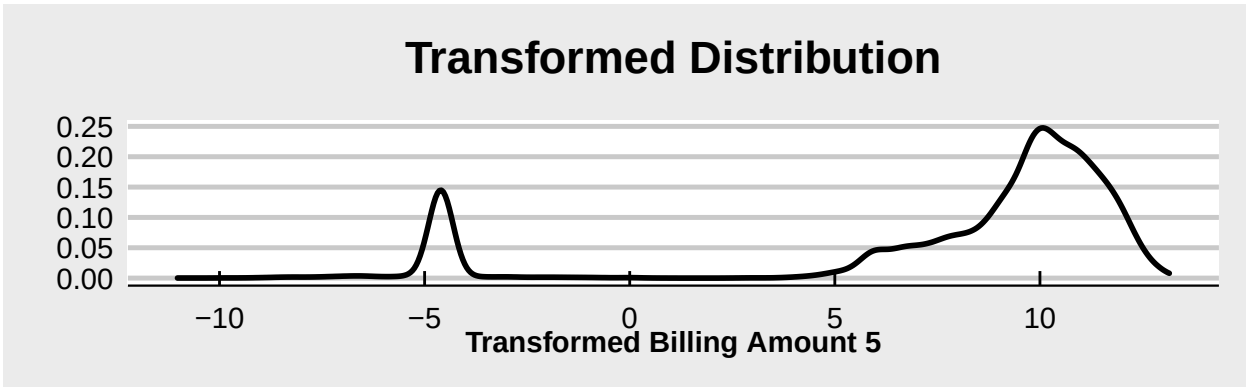

Original Distribution
2.0e−05
1.5e−05
1.0e−05
5.0e−06
0.0e+00
0
500000
1000000
1500000
Billing Amount 3

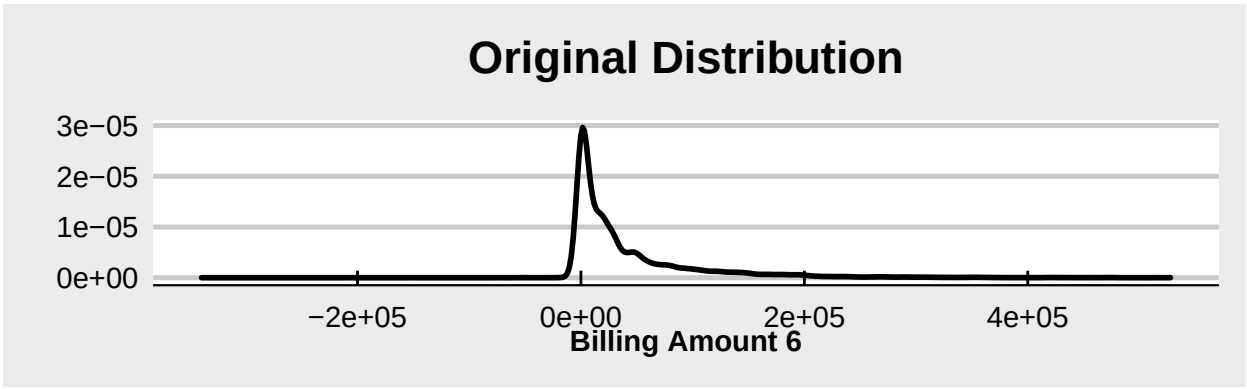

Transformed Distribution
0.25
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
10
15
Transformed Billing Amount 3

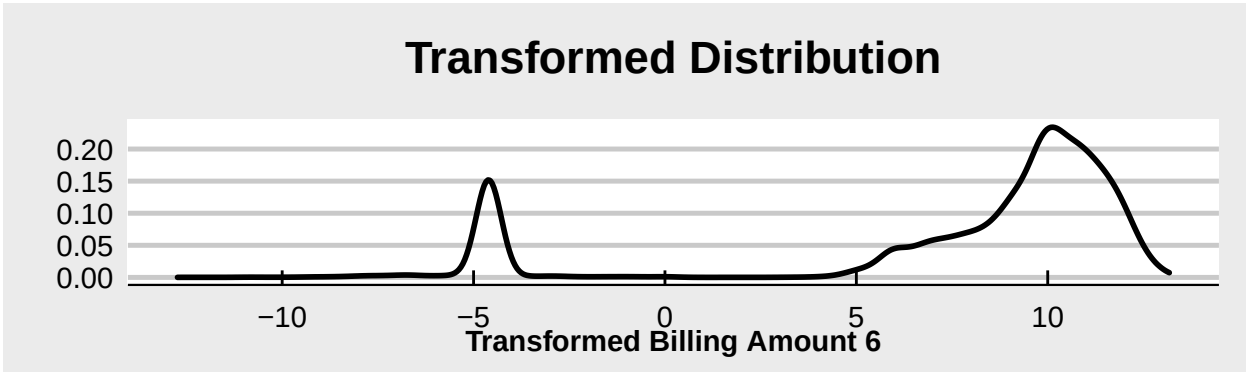

Original Distribution
2.5e−05
2.0e−05
1.5e−05
1.0e−05
5.0e−06
0.0e+00
0
250000
500000
75000
Billing Amount 4

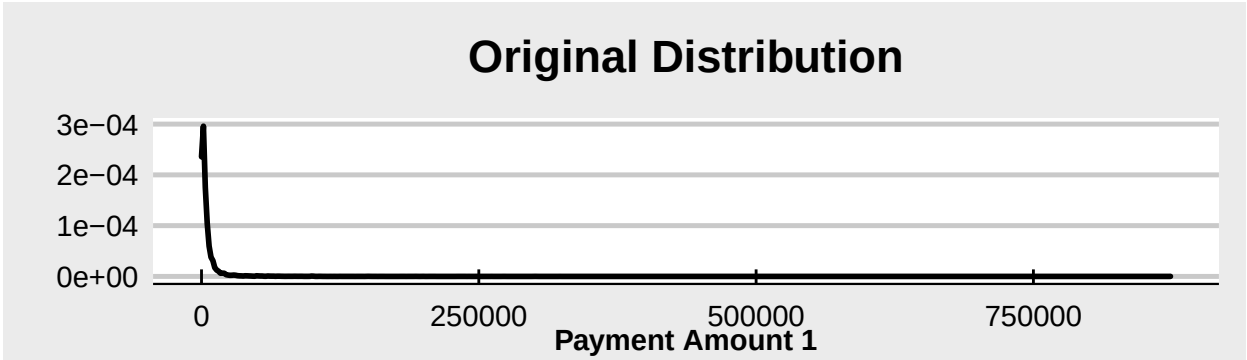

Transformed Distribution
0.25
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
10
Transformed Billing Amount 4

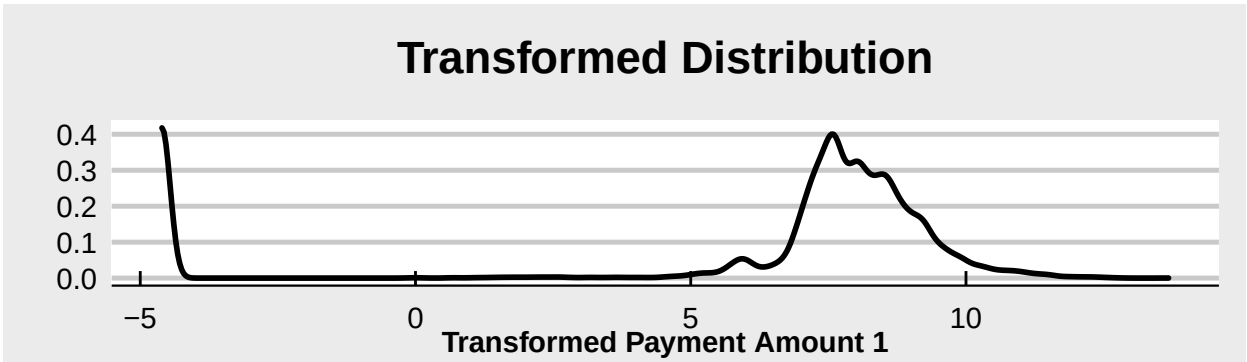

Original Distribution
2e−05
1e−05
0e+00
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Billing Amount 5

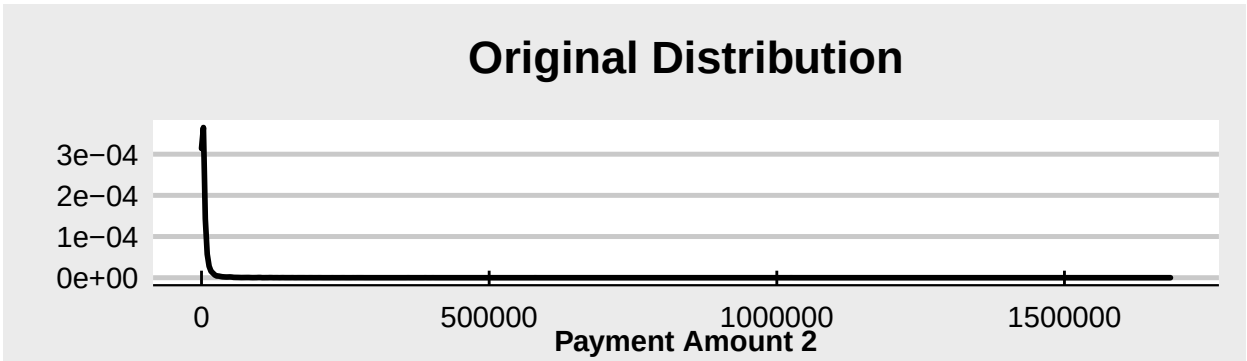

Transformed Distribution
0.25
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
10
Transformed Billing Amount 5

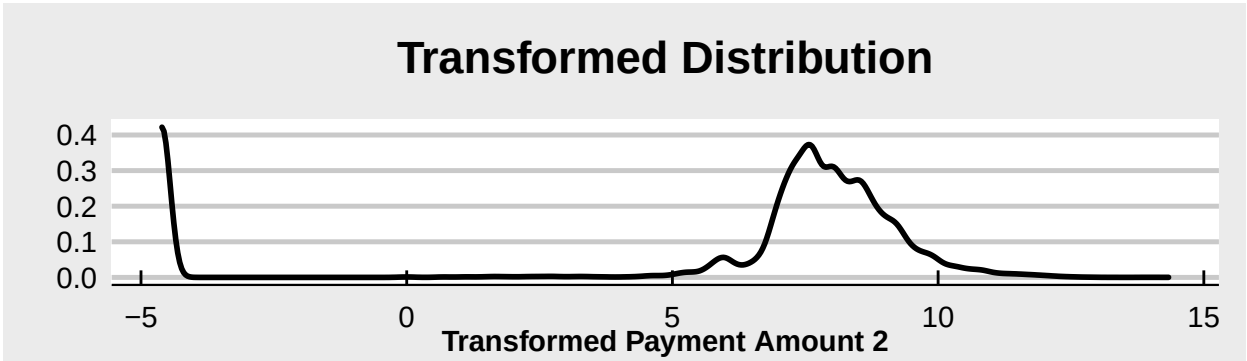

Original Distribution
3e−05
2e−05
1e−05
0e+00
−2e+05
0e+00
2e+05
4e+05
Billing Amount 6

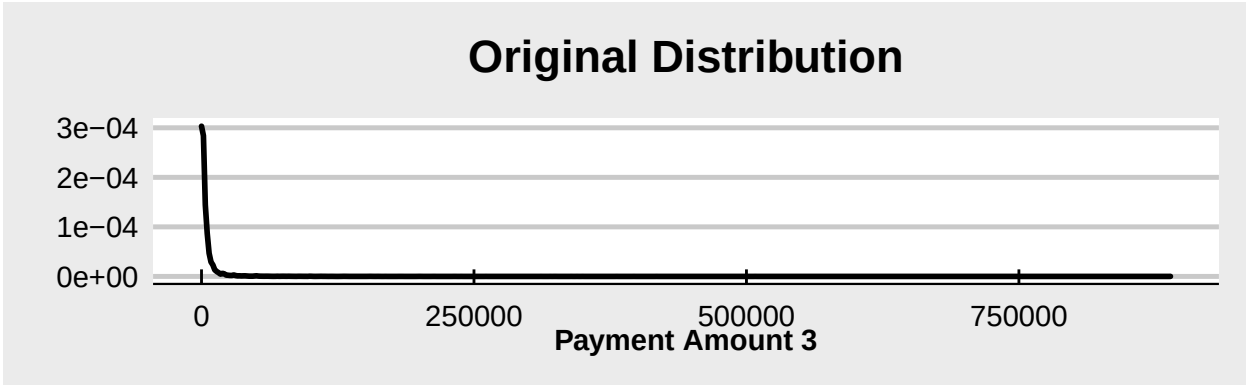

Transformed Distribution
0.20
0.15
0.10
0.05
0.00
−10
−5
0
5
10
Transformed Billing Amount 6

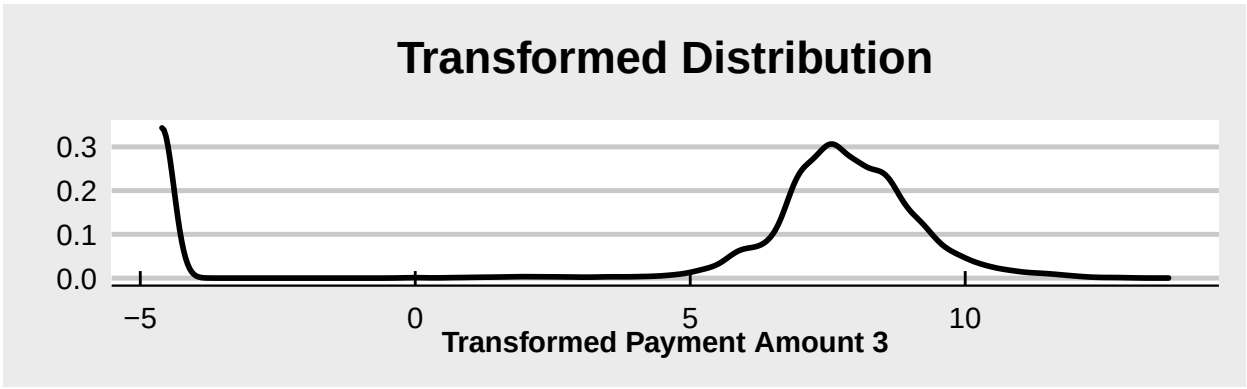

Original Distribution
3e−04
2e−04
1e−04
0e+00
0
250000
500000
750000
Payment Amount 1

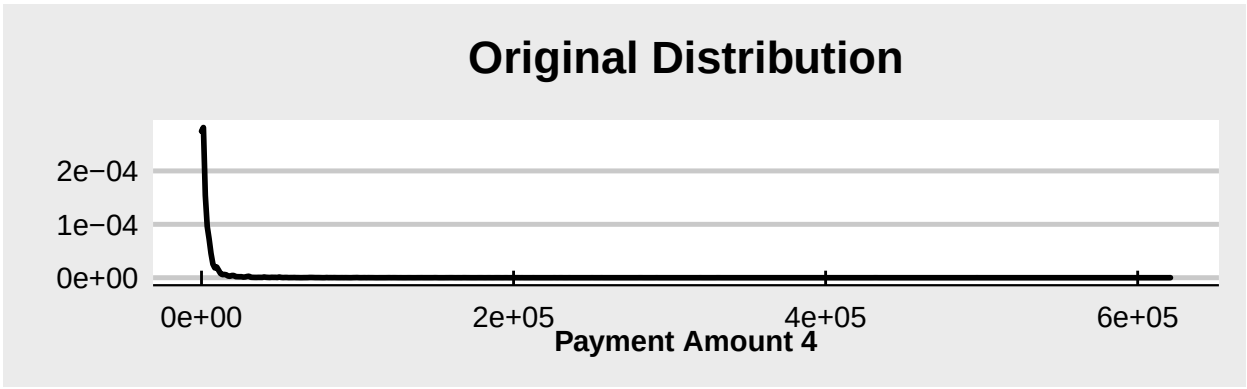

Transformed Distribution
0.4
0.3
0.2
0.1
0.0
−5
0
5
10
Transformed Payment Amount 1

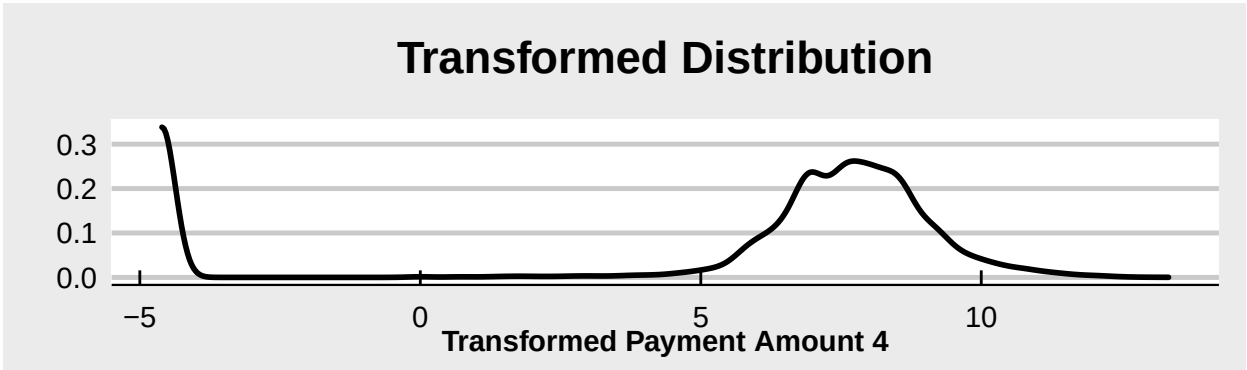

Original Distribution
3e−04
2e−04
1e−04
0e+00
0
500000
1000000
1500000
Payment Amount 2

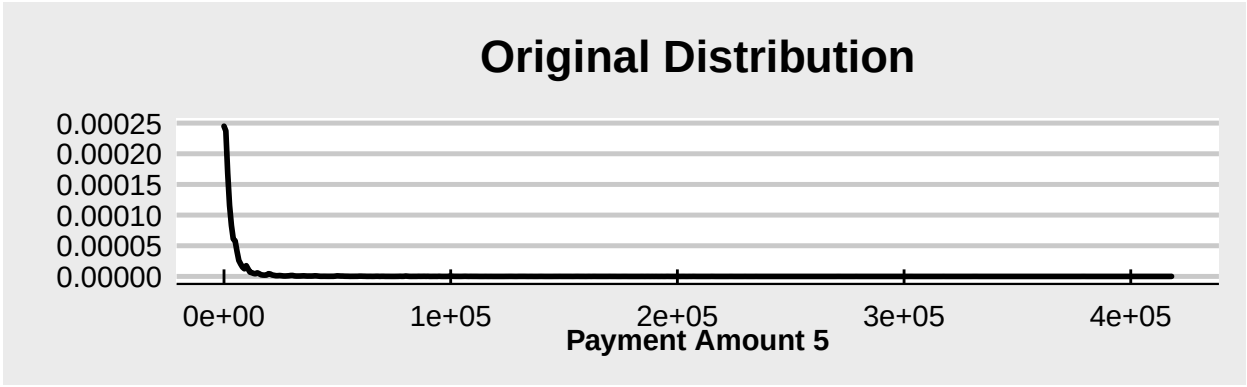

Transformed Distribution
0.4
0.3
0.2
0.1
0.0
−5
0
5
10
15
Transformed Payment Amount 2

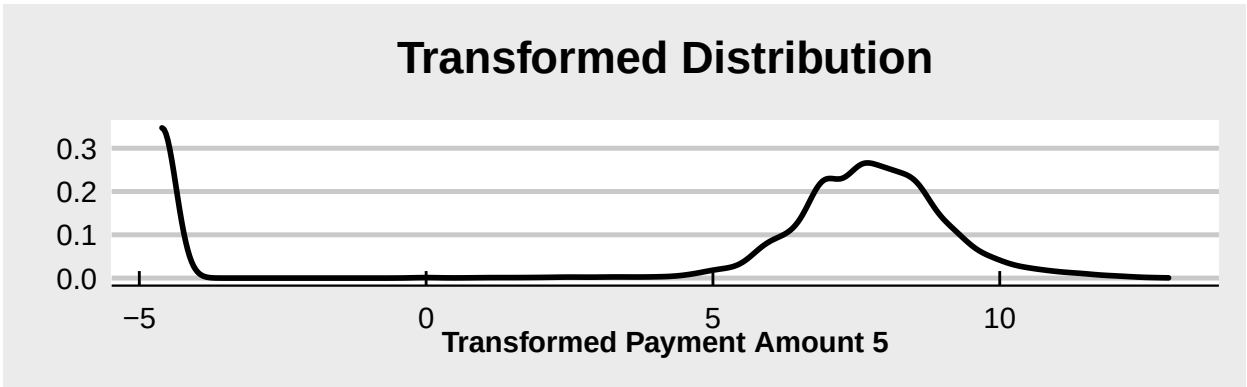

Original Distribution
3e−04
2e−04
1e−04
0e+00
0
250000
500000
750000
Payment Amount 3

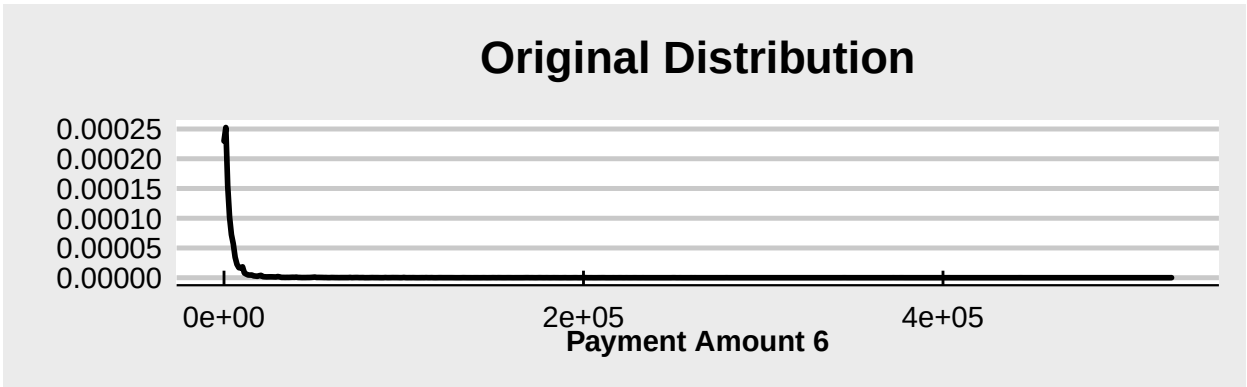

Transformed Distribution
0.3
0.2
0.1
0.0
−5
0
5
10
Transformed Payment Amount 3

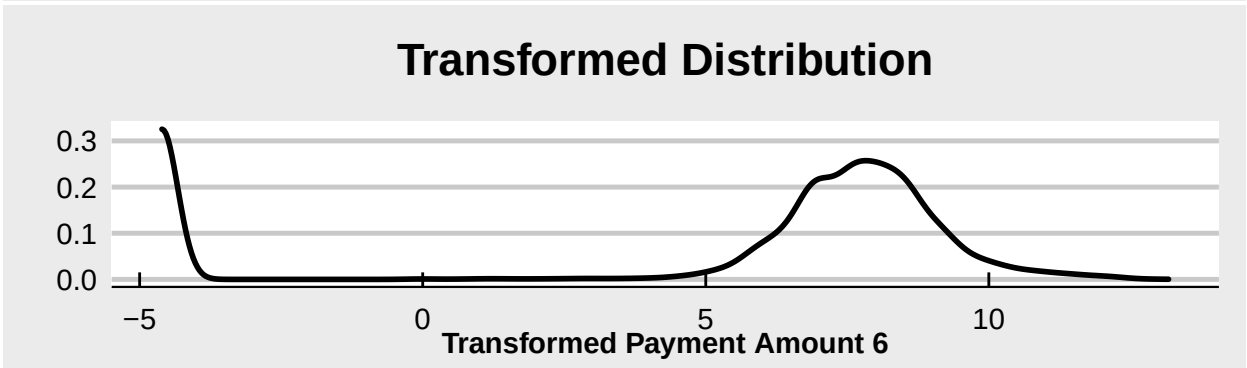

Original Distribution
2e−04
1e−04
0e+00
0e+00
2e+05
4e+05
6e+05
Payment Amount 4


Transformed Distribution
0.3
0.2
0.1
0.0
−5
0
5
10
Transformed Payment Amount 4


Original Distribution
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0e+00
1e+05
2e+05
3e+05
4e+05
Payment Amount 5

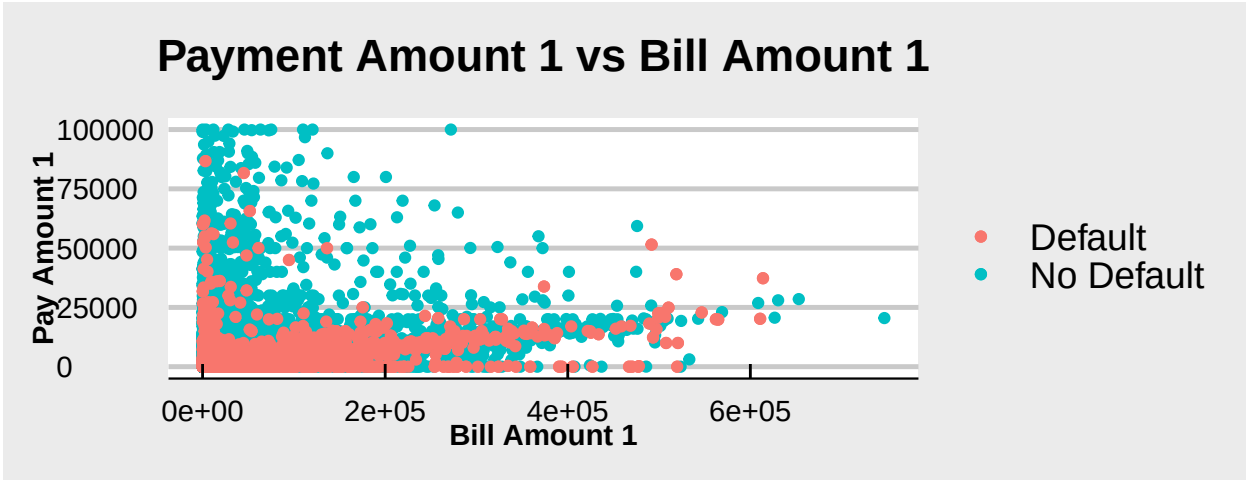

Transformed Distribution
0.3
0.2
0.1
0.0
−5
0
5
10
Transformed Payment Amount 5

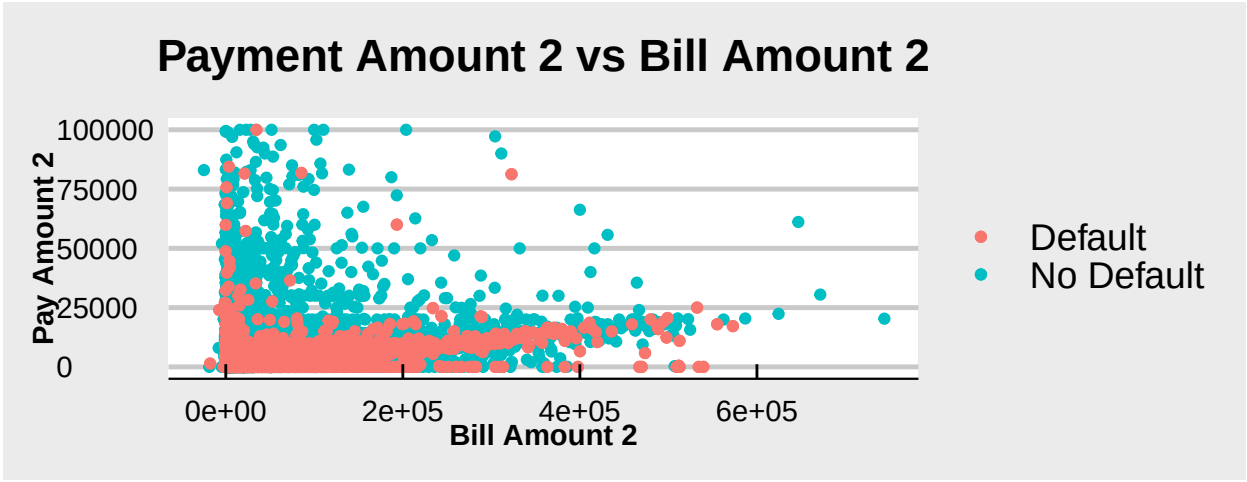

Original Distribution
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0e+00
2e+05
4e+05
Payment Amount 6

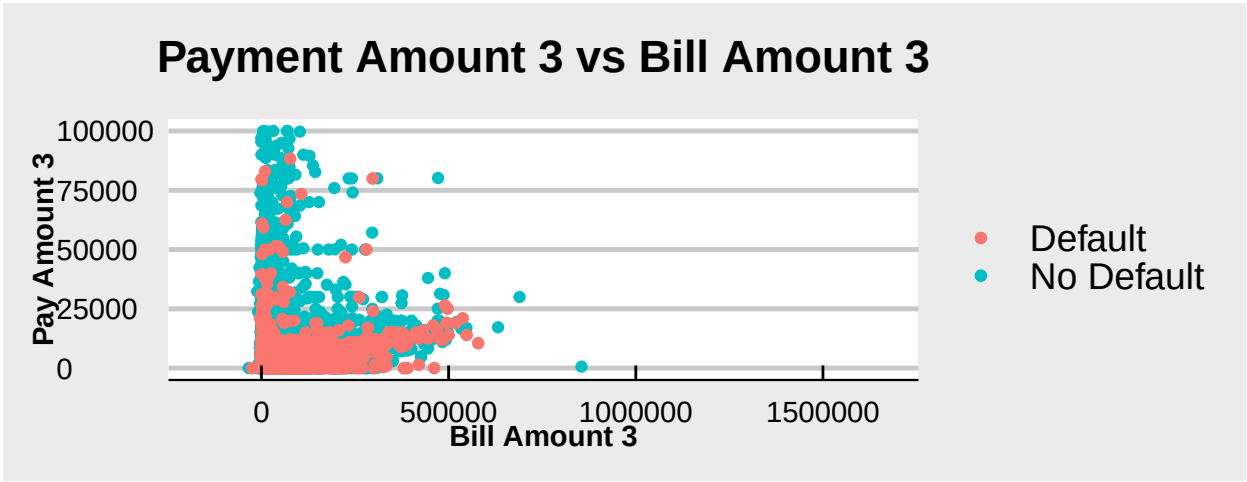

Transformed Distribution
0.3
0.2
0.1
0.0
−5
0
5
10
Transformed Payment Amount 6

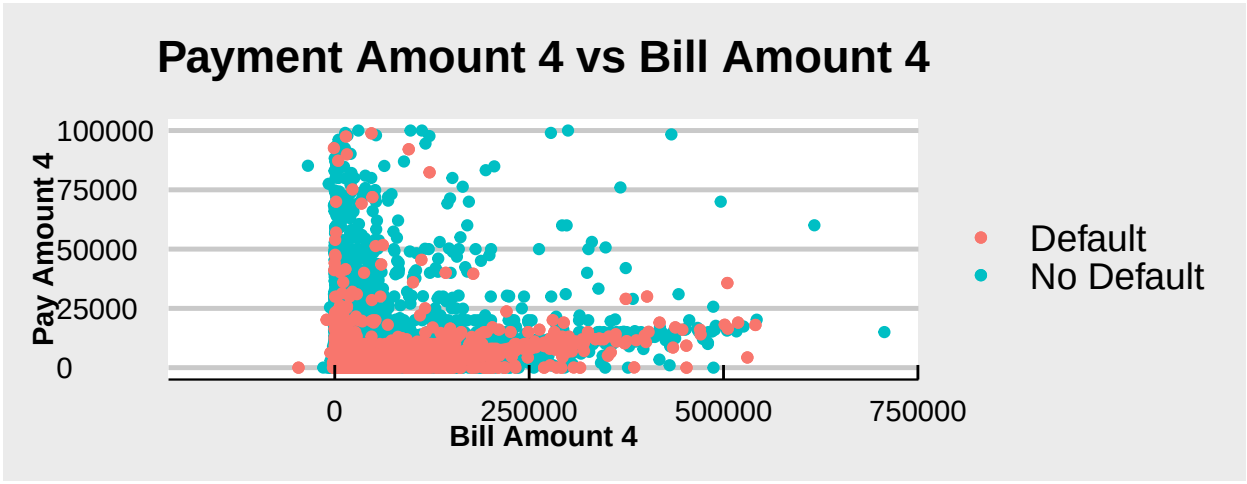
Appendix B

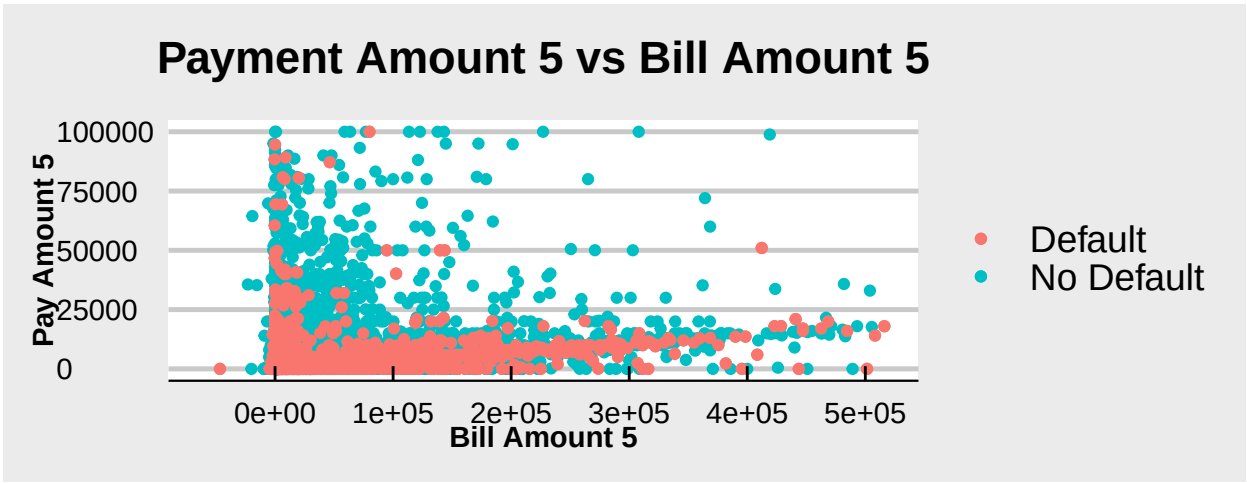
Below are the family of graphs visualizing the relationship between the transformed bill amounts and payment amounts.

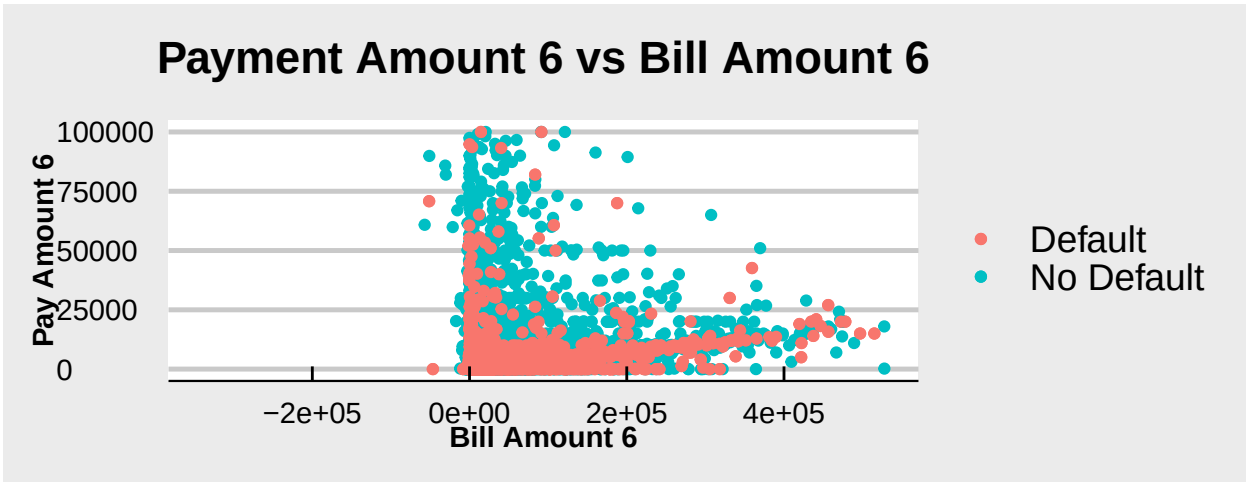







Payment Amount 4 vs Bill Amount 4
Pay Amount 4
100000
75000
50000
25000
0
0
250000
500000
750000
Bill Amount 4
Default
No Default

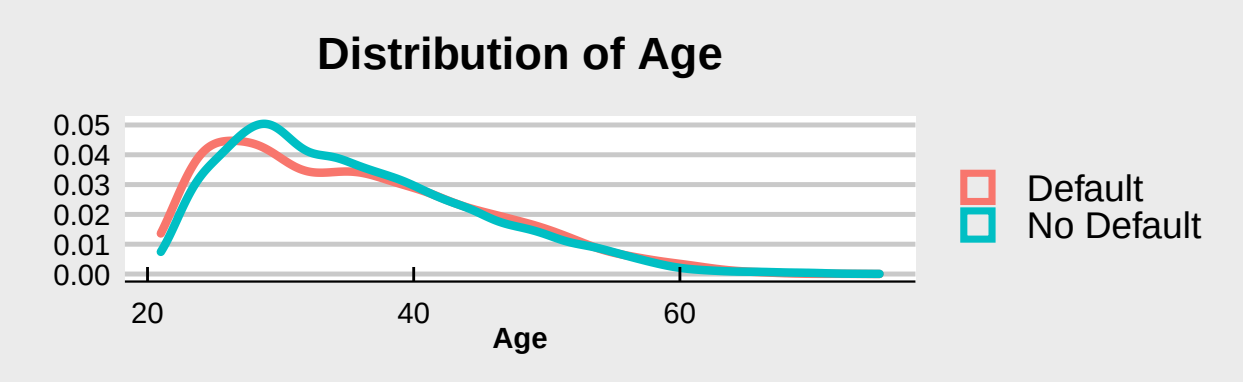

Payment Amount 5 vs Bill Amount 5
Pay Amount 5
100000
75000
50000
25000
0
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Bill Amount 5
Default
No Default


Payment Amount 6 vs Bill Amount 6
Pay Amount 6
100000
75000
50000
25000
0
−2e+05
0e+00
2e+05
4e+05
Bill Amount 6
Default
No Default

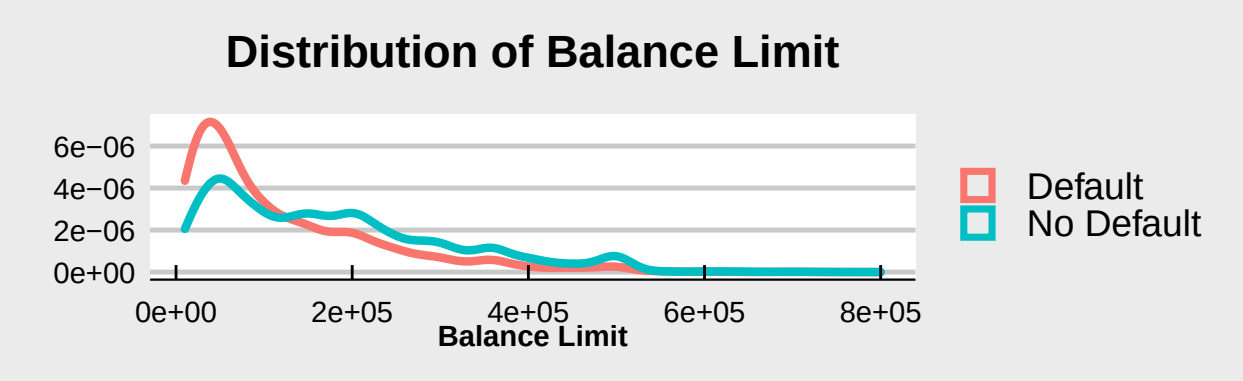
Appendix C

The following plots support our choices to make indicator variables for certain variables.

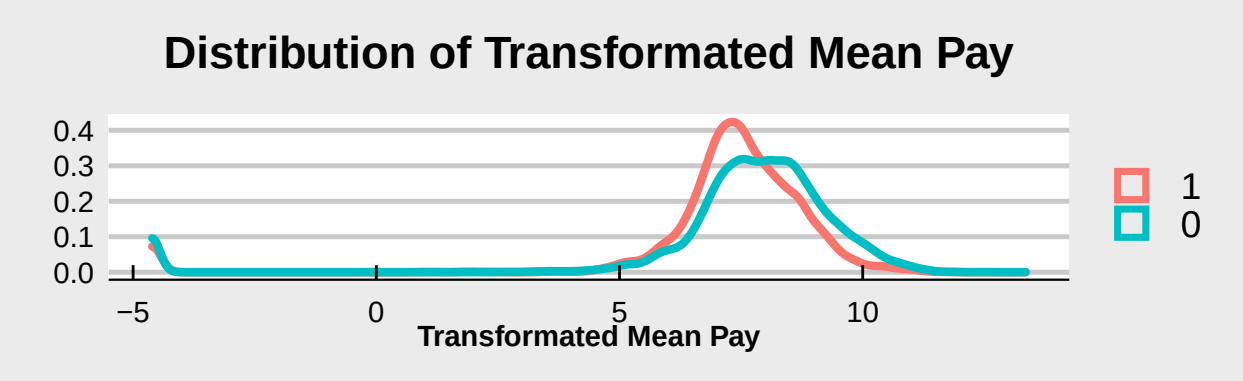
Based on the graph below, we made indicators for the ranges [0, 27), [27, 40), [40, 55), and [55, ∞).



Based on the graph below, we made indicators for the ranges of [0, $\$1.25 * 10^5$) and [$\$1.25 * 10^5$, ∞).



Based on the graph below, we made indicators for the ranges of [0,8) and [8,∞)



Appendix -: Principal Component Analysis (PCA)

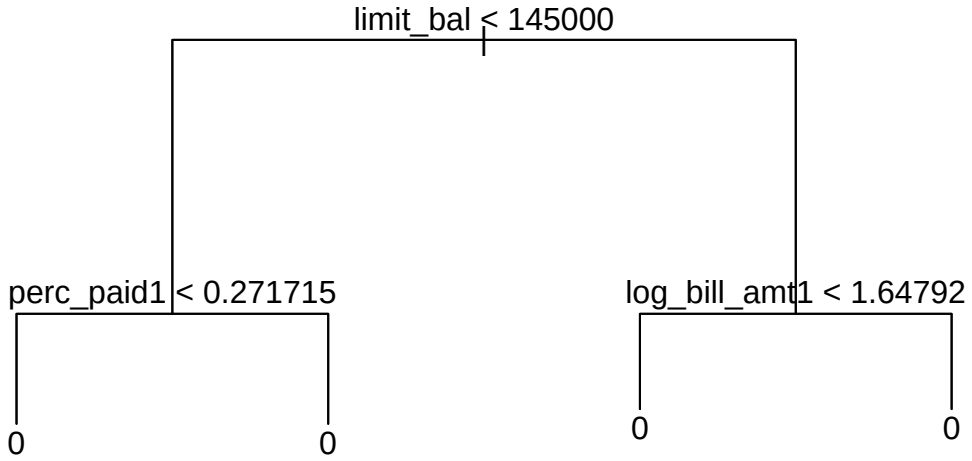
Principle Component Analysis (PCA) is an unsupervised dimension reduction technique. We look at the features in question and seek to create a number of linear combinations of these features. Of course, the number of combinations must be less than the number of original features. These combinations are refered to as components.

In PCA, we use calculus to maximize the variance of these components, placing constraints on the coefficients of each term of the linear combination to ensure that a maximum can be found. In a small example with just two features, we would maximize $\phi_1 V_1 + \phi_2 V_2$ where $\phi_1^2 + \phi_2^2 = 1$.

After completing this dimension reduction, we can implement another modelling technique using the components as features in the model.

Appendix -:Random Forest

To understand the Random Forest modeling technique, we will first explore the idea of a decision tree. A decision tree repeatedly splits the data systematically in an attempt to isolate groups of observations into regions. If a new observation is located within a specific region, it will be classified as the mode response for that region. These are refered to as trees because the can be visualized as such. Below is a small example.



We can follow the path of a new observation through, asking first if it has a balance limit of less than $145,000, then the corresponding question for the correct branch. Notice here that all regions predict observations as 0 (no default). This shows anecdotely the reason why trees are not very accurate.

However, random forest creates a large number of these trees and averages the predictions made by each tree to reach a final prediction value. In order to ensure that each tree is created differently, it only considers `m` random variables at each split in the tree, hence the the term Random Forest.