

Predicting Payment Default

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 five within this report: logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), 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.

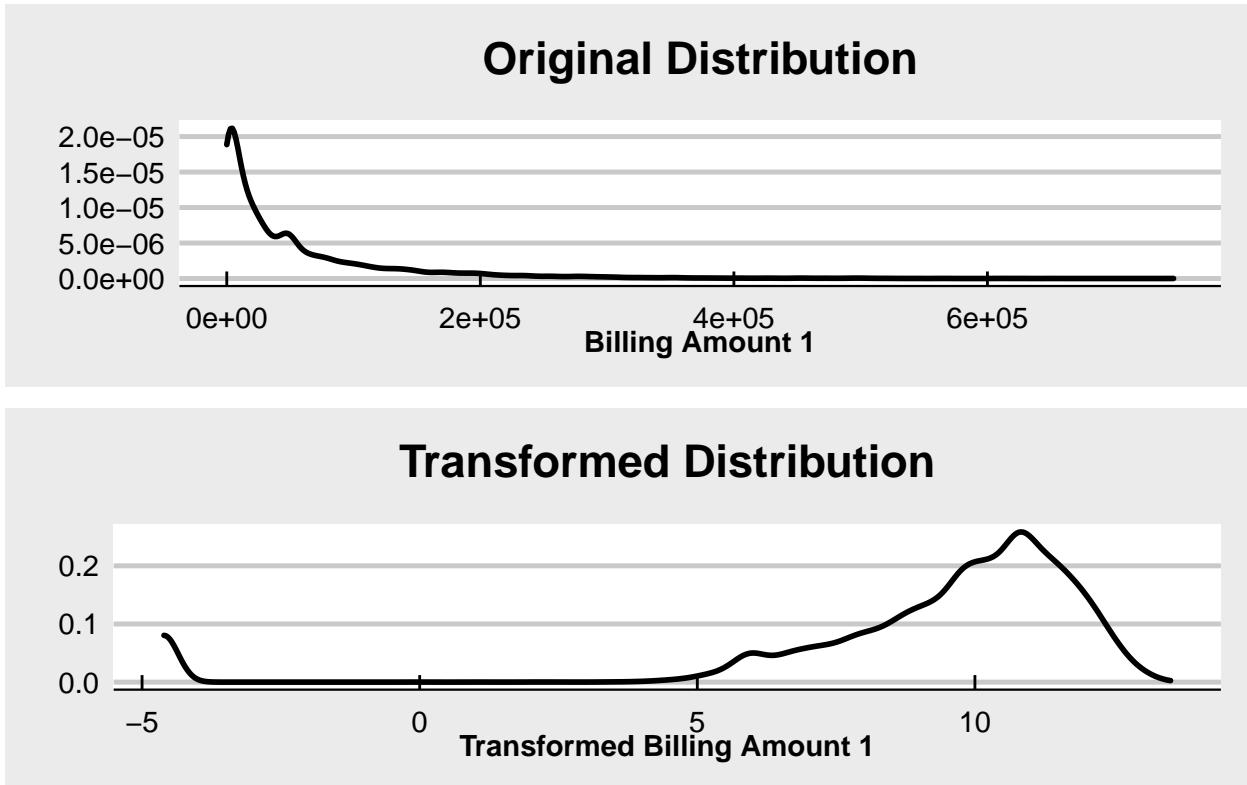
Data Processing

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.

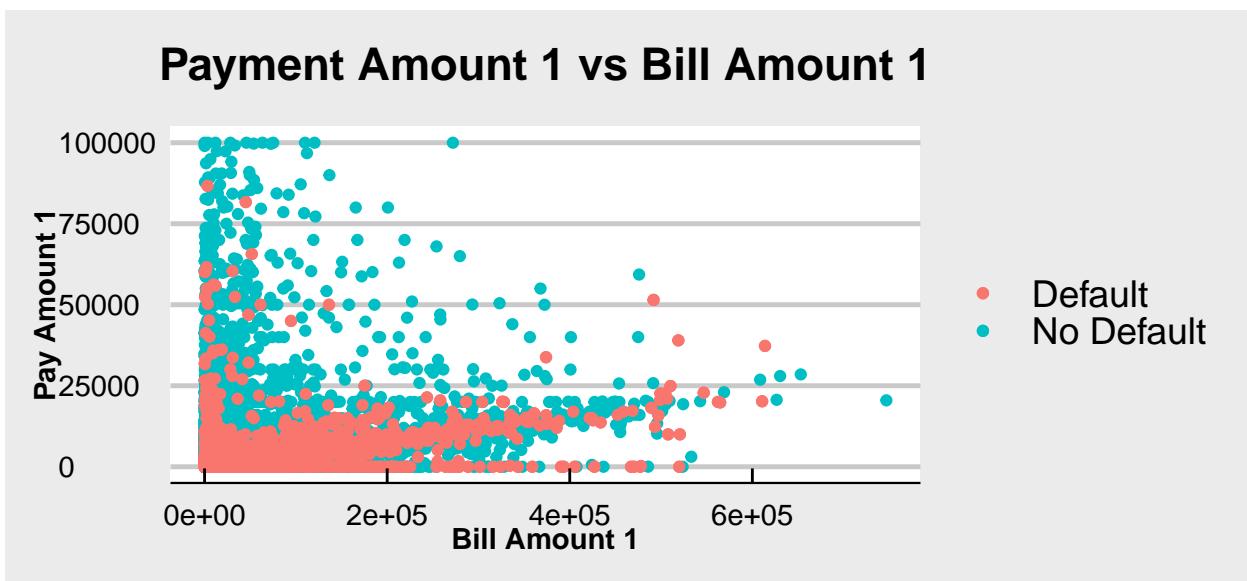
Exploratory Data Analysis (EDA) and initial modelling attempts quickly illustrate the need for new features and the transformation of existing features. First, simple density plots of the features `Billing Amount 1, ..., Billing Amount 6` and `Payment Amount 1, ..., Payment Amount 6` as well as `age` 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.

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

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.

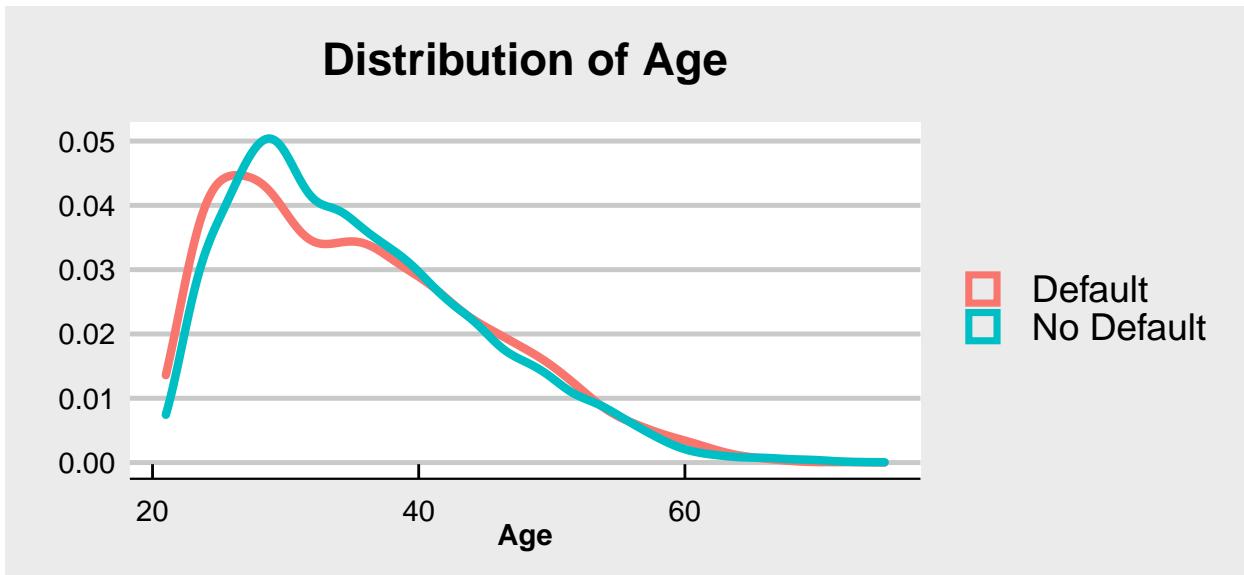


Again, observations which have a transformed billing amount of zero present a challenge. This is dealt with through the logic below.

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

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 $[55, \infty)$.



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. This concludes our data processing.

Modeling

We will now look at the processes of building and testing our models. We will use the log loss score as our primary measure of accuracy. The log loss function garners its fame for its use as an evaluation metric in Kaggle. The goal is to effectively minimize the Log Loss for greater success. Log loss quantifies the accuracy of a classifier by penalizing all false classifications. Essentially, the minimization results in a boost in accuracy. The log loss calculation hinges upon the classifier assigning a probability to each class rather than yielding the most likely class. The log loss was adopted into a function for the purposes of repetitive use in the project.

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. An important observation is that of the “percentage paid” variables, only Percentage Paid 1, 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. 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 1.8396, so we run a new logistic regression model with only the significant variables included, but we only improve the log-loss to 1.6171. 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 D). We attempt to perform forward, backward, and mixed selection using the `stepAIC` function, but `stepAIC` will not work because the AIC is negative infinity for our model. This means that `stepAIC` would lead to over-fitted 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: Transformed Balance Limit, Transformed Bill Amount 2, Transformed Bill Amount 3, Transformed Pay Amount 1, and Transformed Pay Amount 2. 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 0.4656, which is much better than the log-loss of 1.6171 that we got from logistic regression.

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 E. 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`.

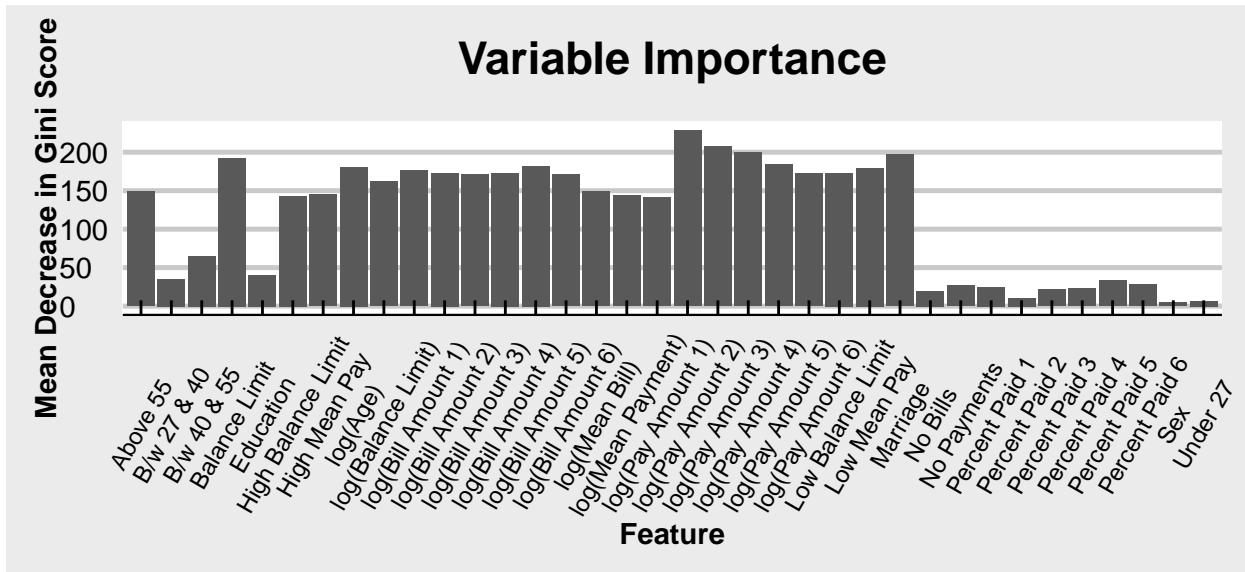
This model produces predictions on the test data with a log loss score of 0.4993. However, a closer look at the results reveals that all observations are predicted to not default. This is the same behavior as the trivial model. Therefore, although the log loss is better than the trivial model, it is 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. Because we appear to have reached the limits the LDA/QDA methodology, we move on to explore a very different technique: random forest.

Random Forest

Random forest is a non-parametric approach, and is therefore very different from the prior methods. More information on the random forest technique can be found in Appendix F.

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.

Conclusion

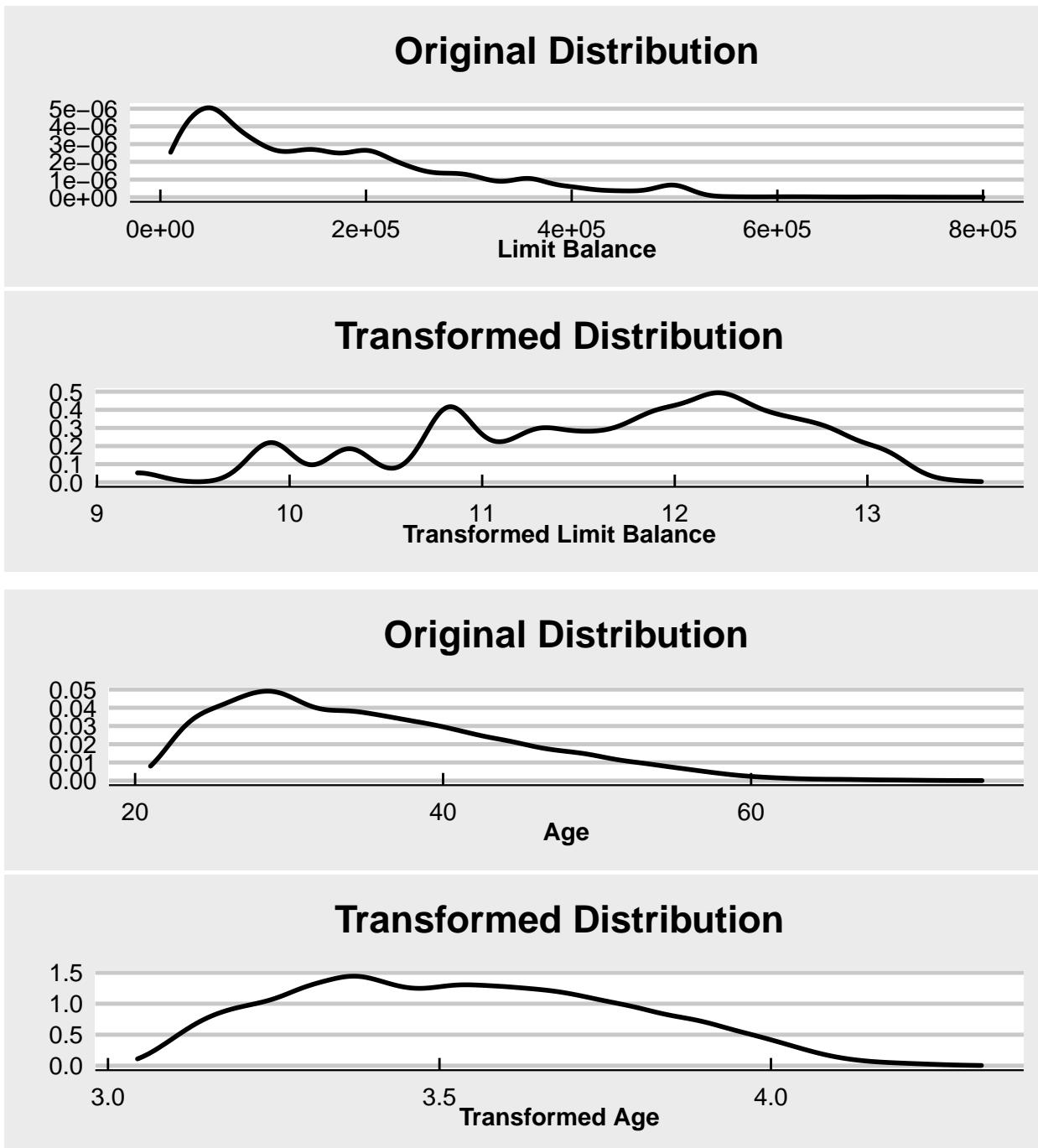
After completing this analysis, many of the modelling techniques did not produce a good fit, however LDA and random forest were able to produce impressive models. Random forest has achieved the best log loss score and is therefore the best model we are able to achieve within the scope of this project. The overall accuracy (ie misclassification rate) for all our models is relatively the same, but we are most confident in the predictions of the random forest model, based on the log loss score. The performance of this model on more data solidifies this conclusion, producing a log loss of .4437.

The team's next step is to delve more into different machine learning techniques. This includes Neural Networks and Generalized Additive Models (GAMs). Neural Networks is a powerful set of algorithms that perform computations similar to neurons in the human brain. In contrast, GAMs are highly effective in instances where the relationship between the features and y is not linear. It would be worthwhile to see the results that we achieve from implementing these algorithms. Secondly, we would put a request in to banks for more data sets. It would be helpful in improving model performance as most of the techniques such as Boosting and Random Forests are very sensitive to the amount of data.

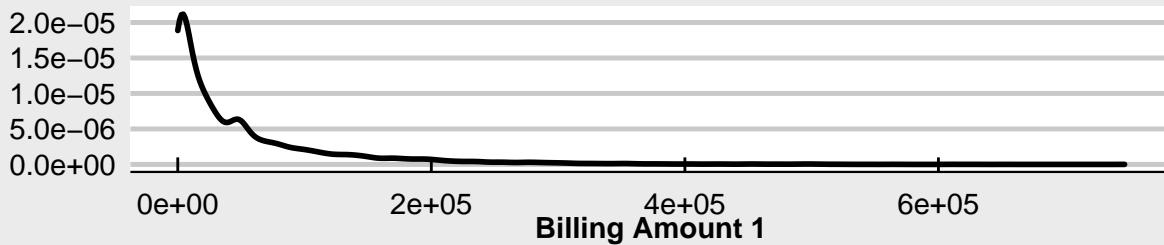
Appendices

Appendix A: Transformations

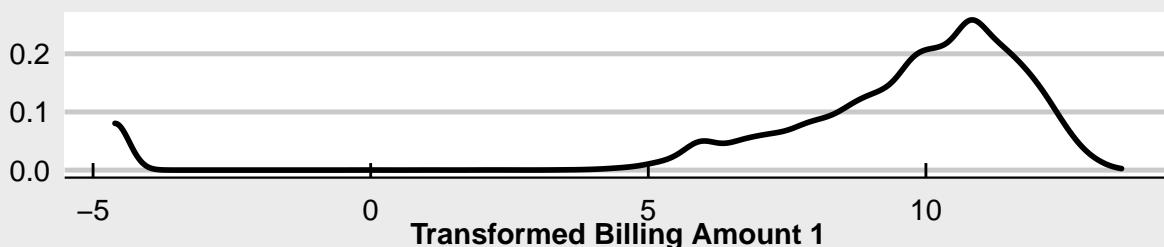
The following are plots of the variables transformed according to equation (1) before and after the transformation.



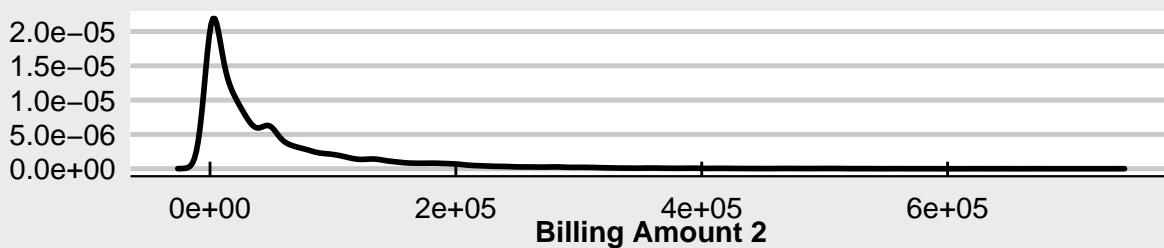
Original Distribution



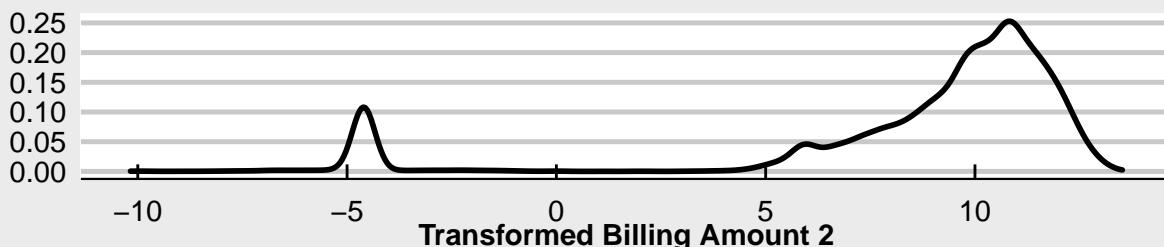
Transformed Distribution



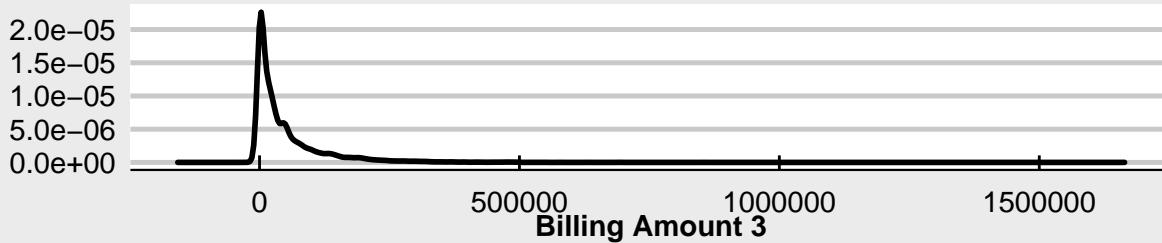
Original Distribution



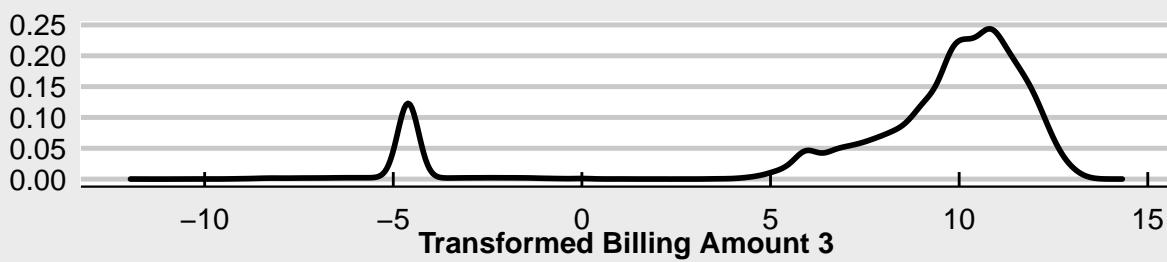
Transformed Distribution



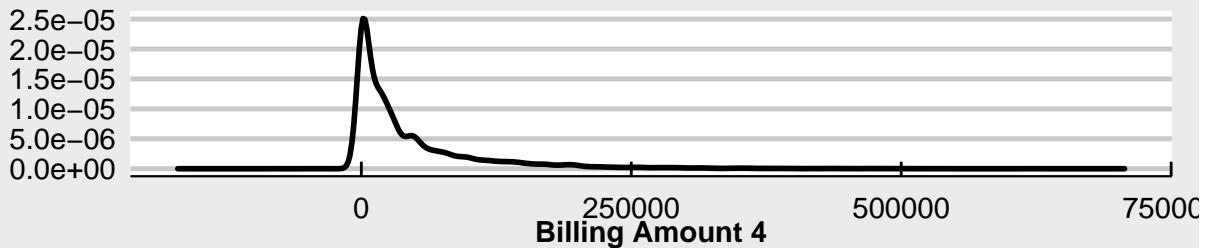
Original Distribution



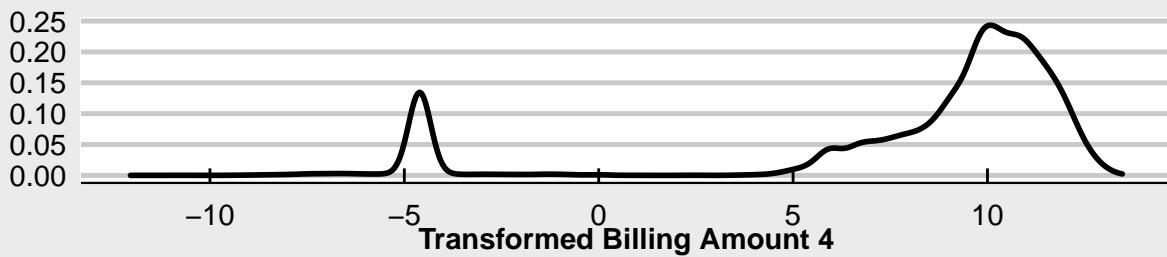
Transformed Distribution



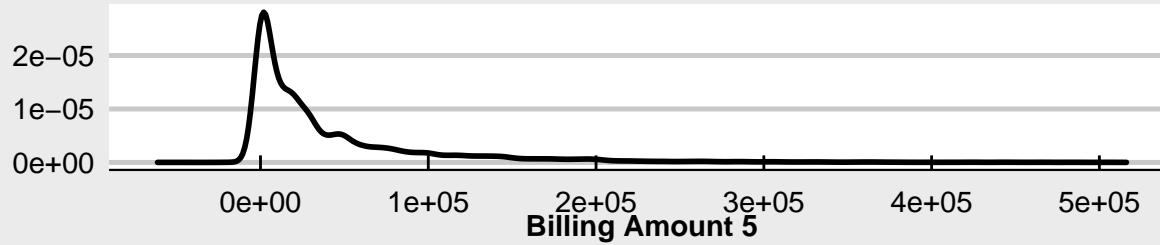
Original Distribution



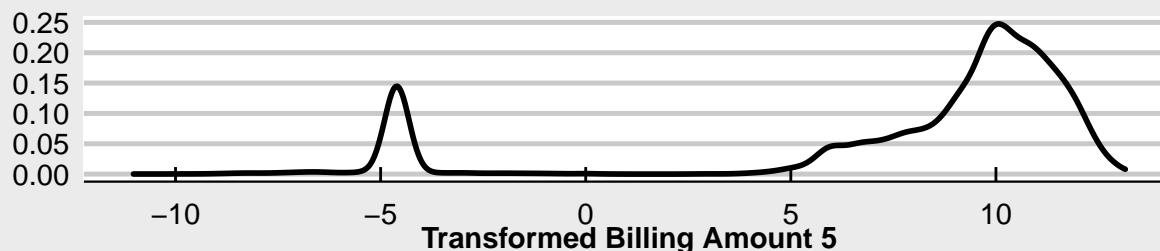
Transformed Distribution



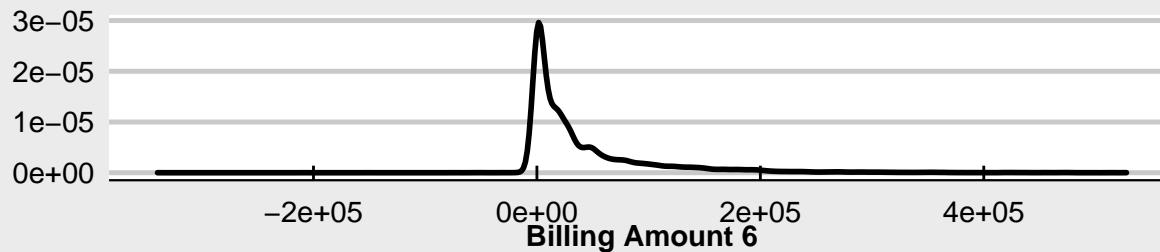
Original Distribution



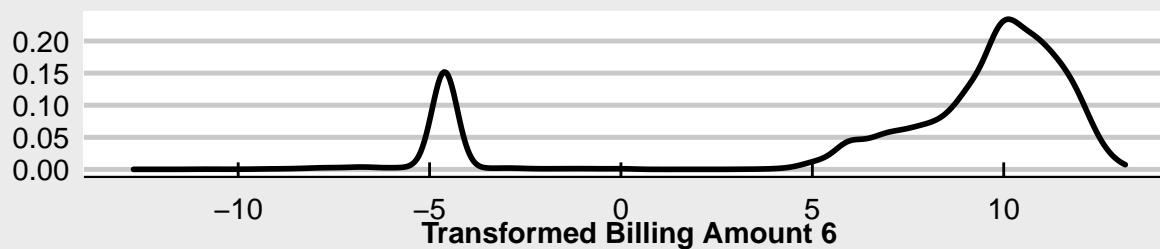
Transformed Distribution



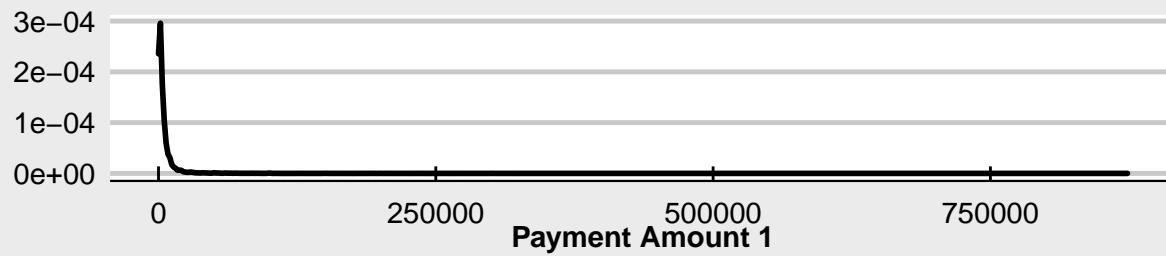
Original Distribution



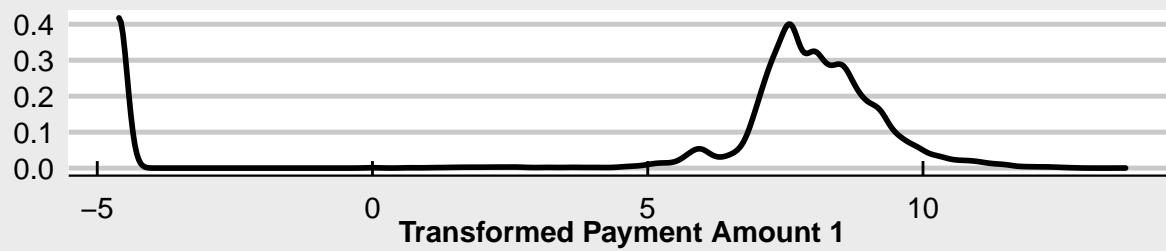
Transformed Distribution



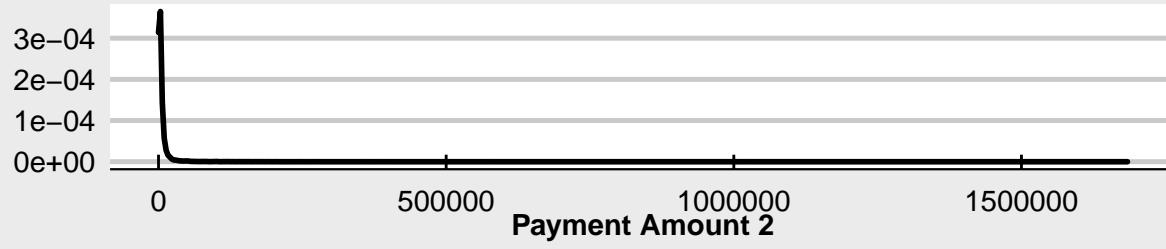
Original Distribution



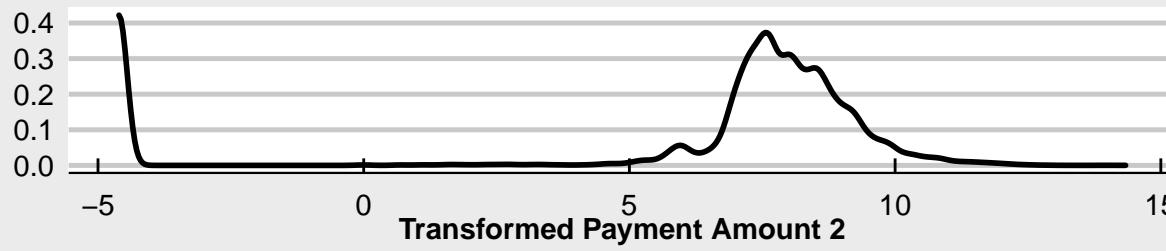
Transformed Distribution



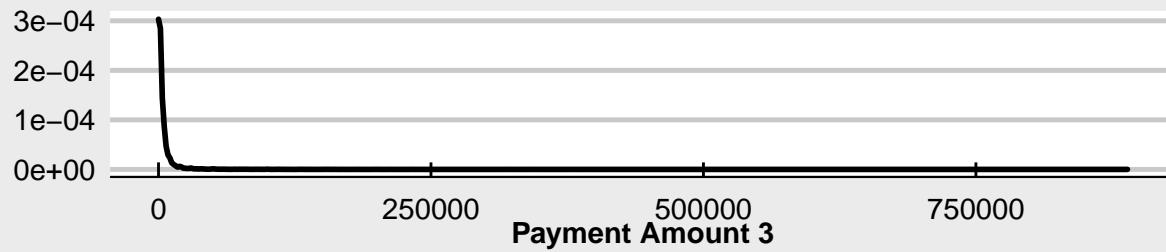
Original Distribution



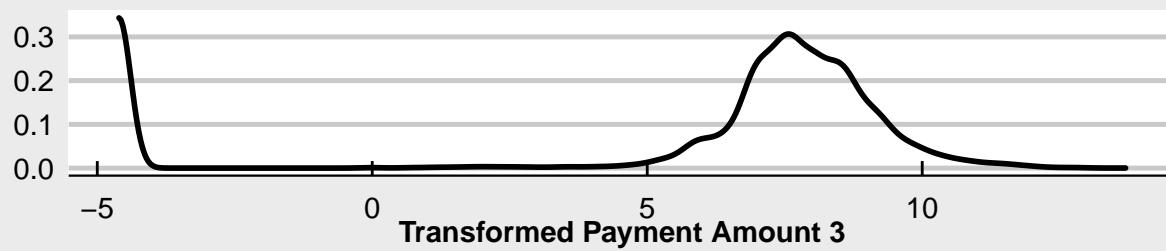
Transformed Distribution



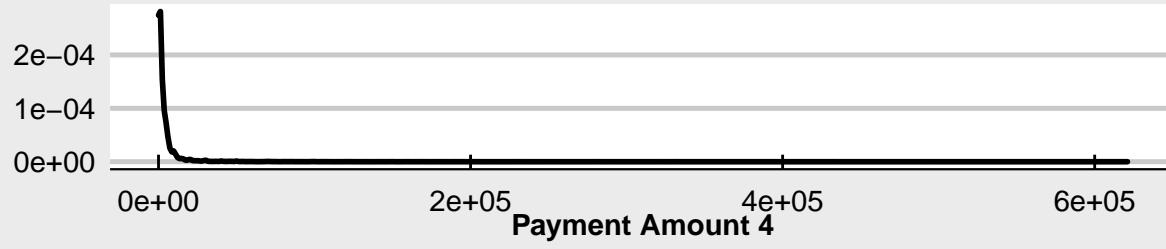
Original Distribution



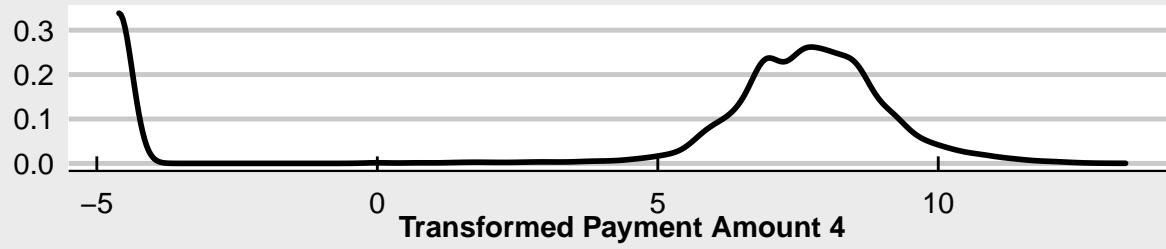
Transformed Distribution



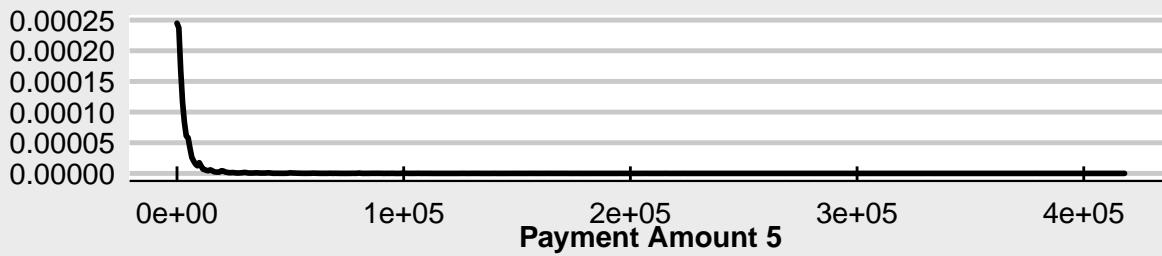
Original Distribution



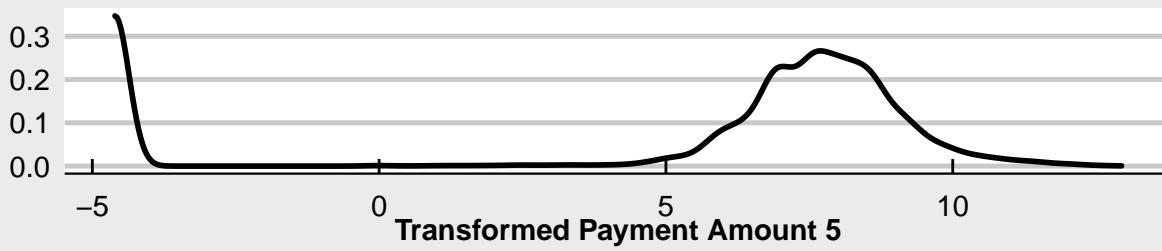
Transformed Distribution



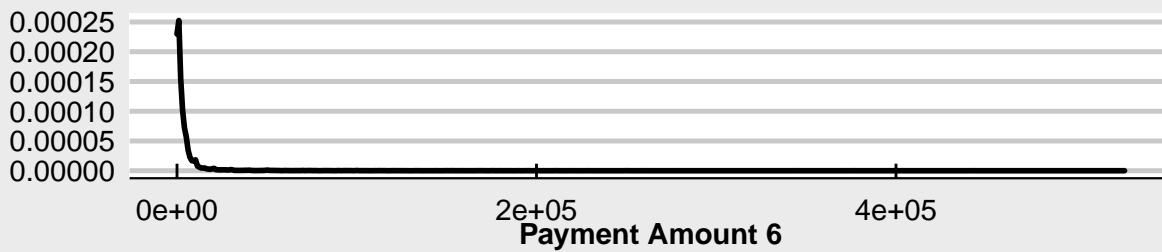
Original Distribution



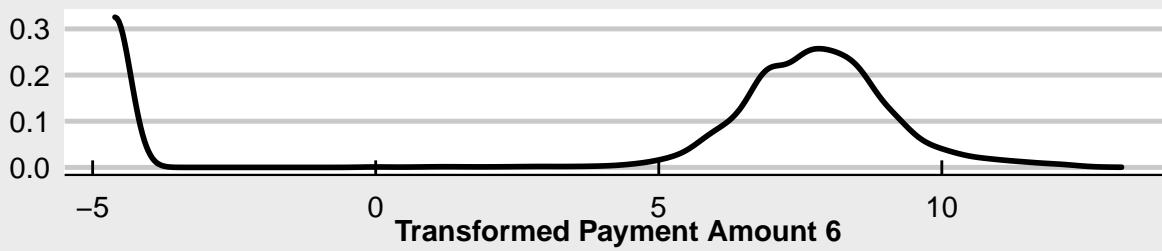
Transformed Distribution



Original Distribution

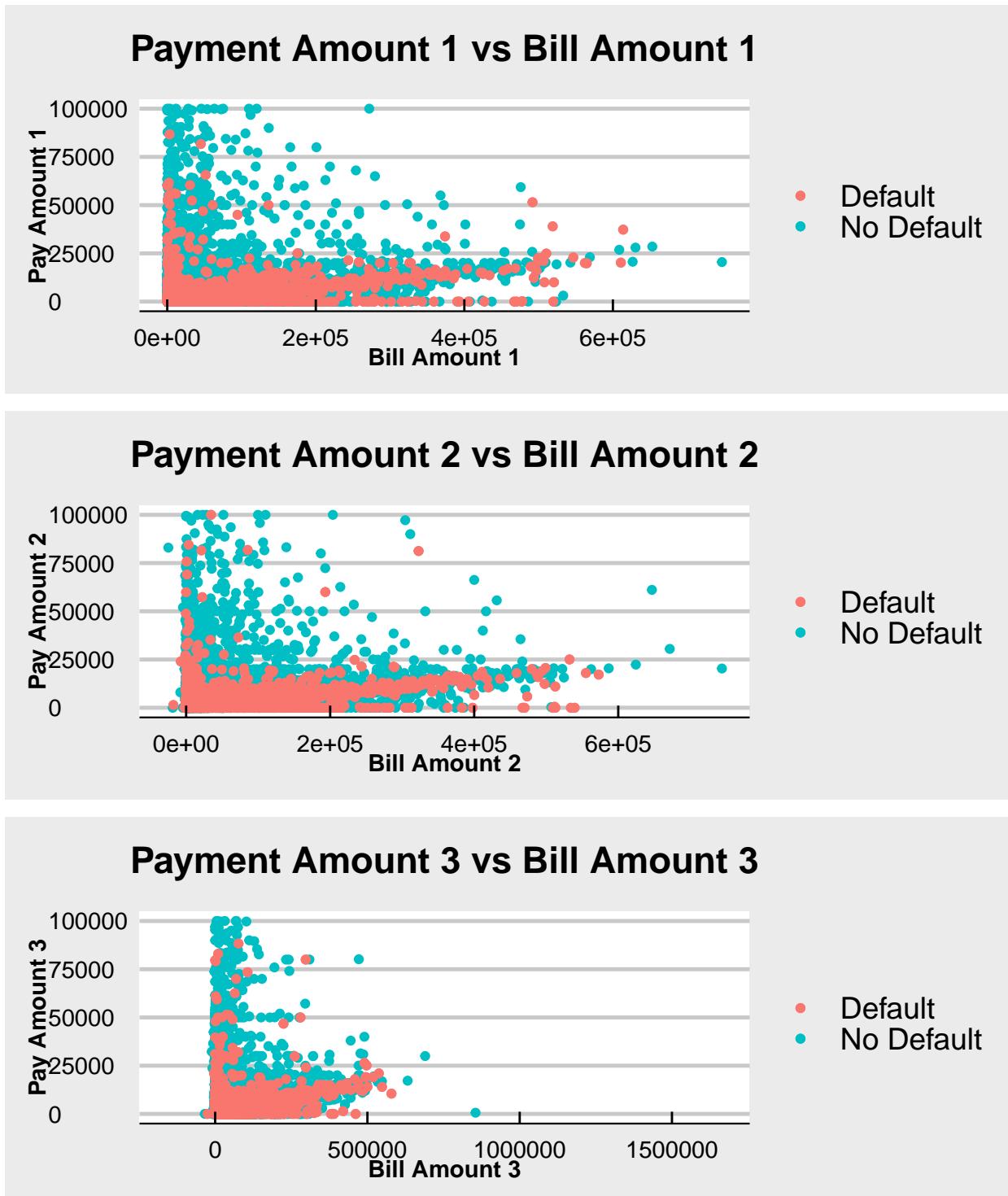


Transformed Distribution

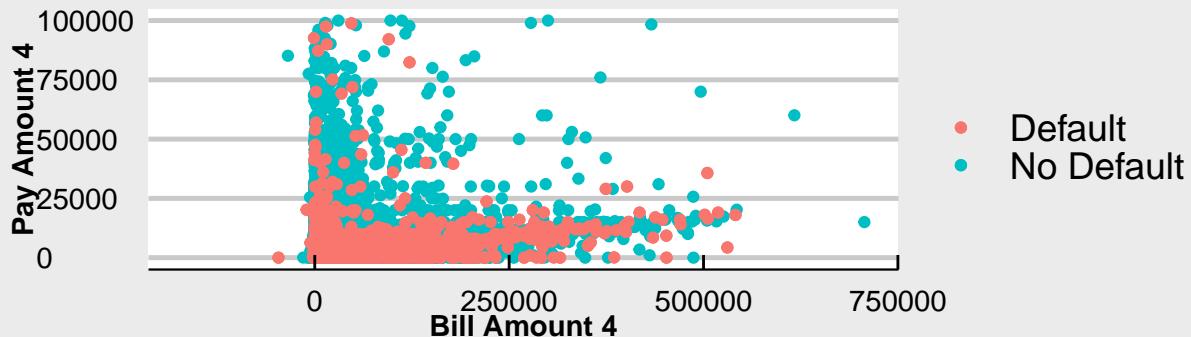


Appendix B: Percent Paid Feature Creation

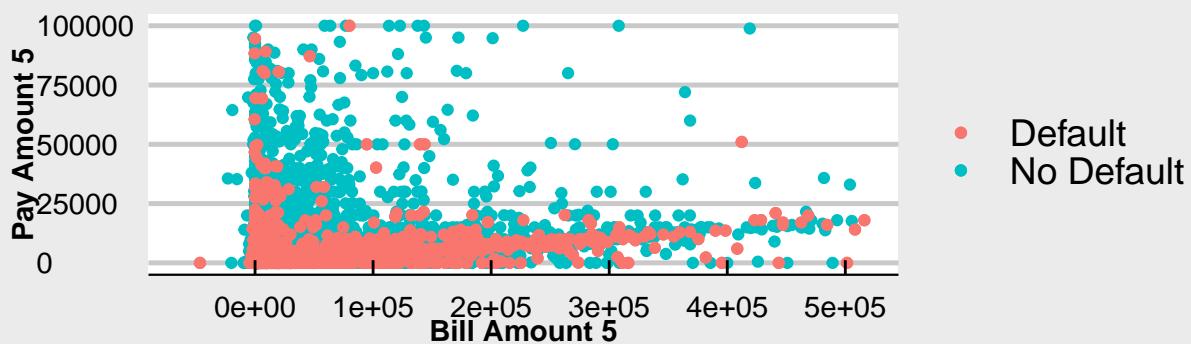
Below are the family of graphs visualizing the relationship between the transformed bill amounts and payment amounts.



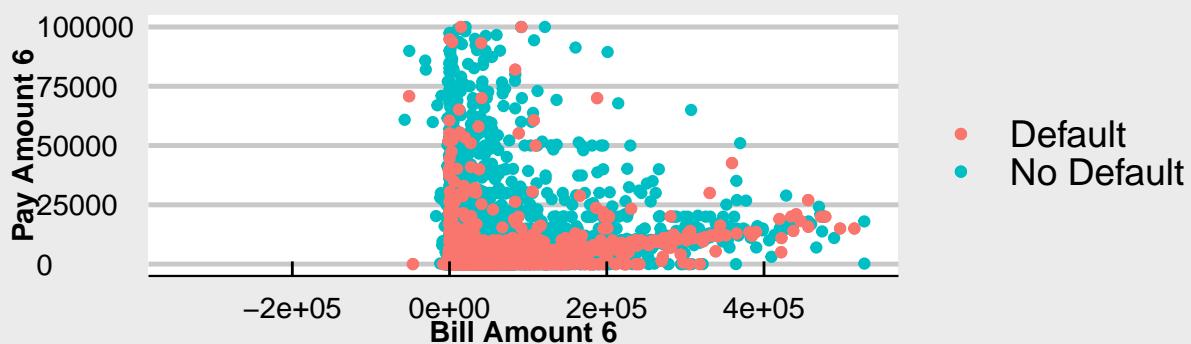
Payment Amount 4 vs Bill Amount 4



Payment Amount 5 vs Bill Amount 5



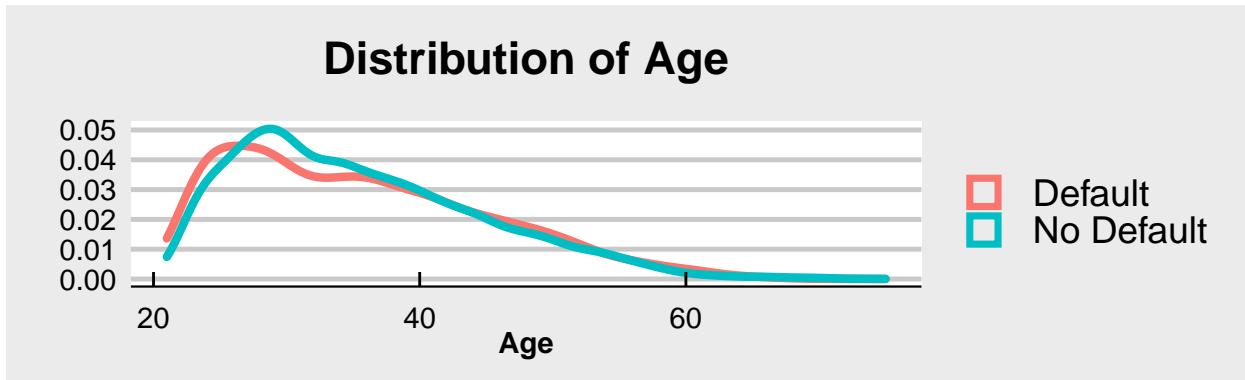
Payment Amount 6 vs Bill Amount 6



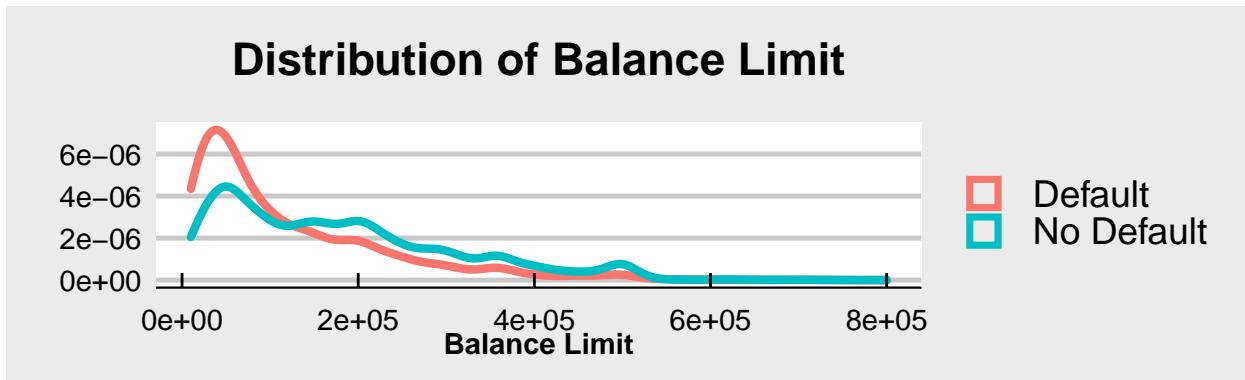
Appendix C: Indicator Variable Creation

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

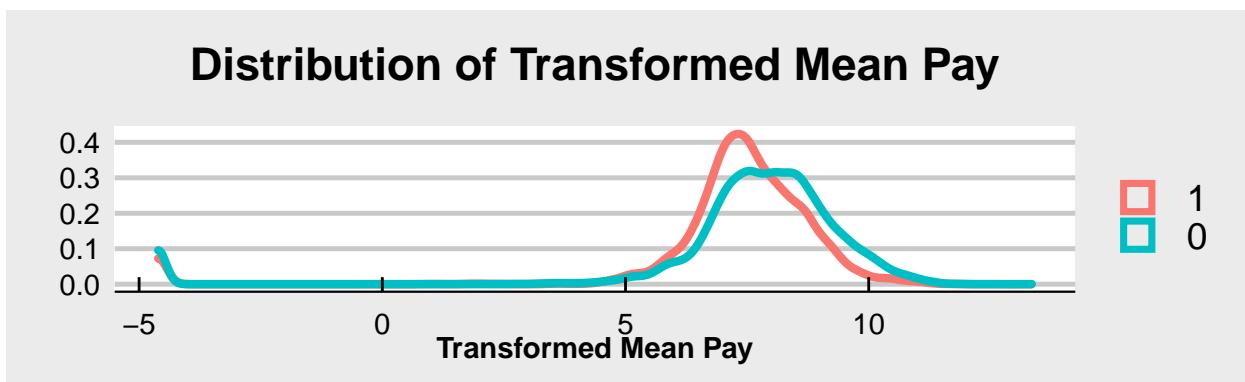
Based on the graph below, we made indicators for the ranges $[0, 27)$, $[27, 40)$, $[40, 55)$, and $[55, \infty)$.



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



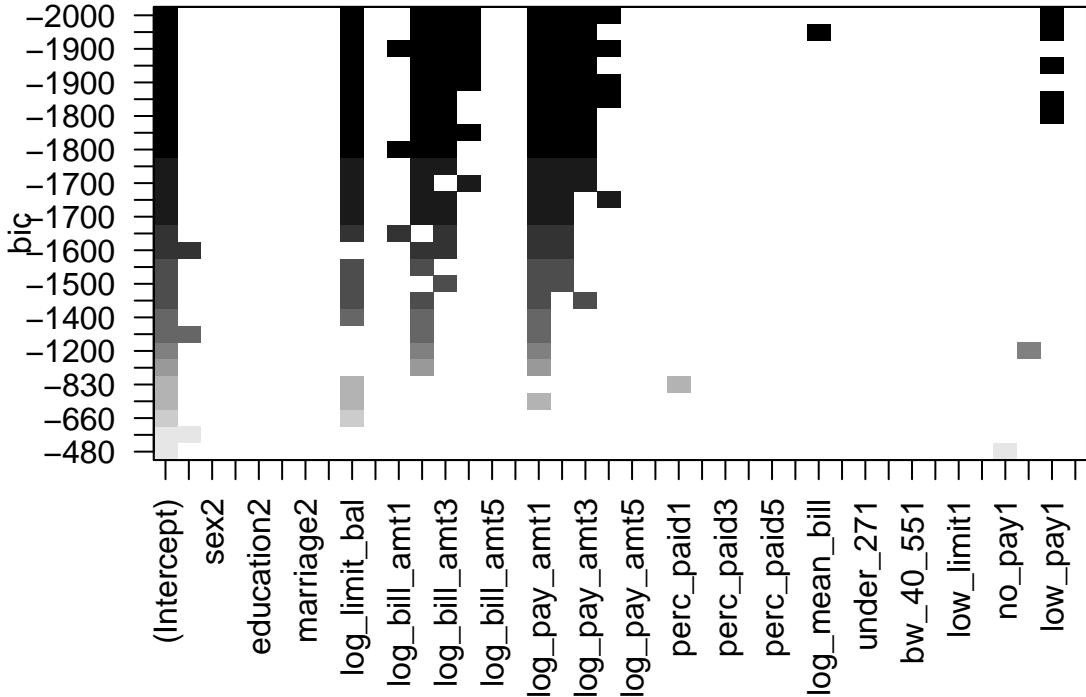
Based on the graph below, we made indicators for the ranges of $[0, 8)$ and $[8, \infty)$



Appendix D: Variable Selection

We can use All-Subset Selection to determine the most influential variables. The 3 best models with each number of variables included ($1 \dots p$) are recorded, and displayed by BIC score below.

```
## Reordering variables and trying again:
```



Appendix E: Principal Component Analysis (PCA)

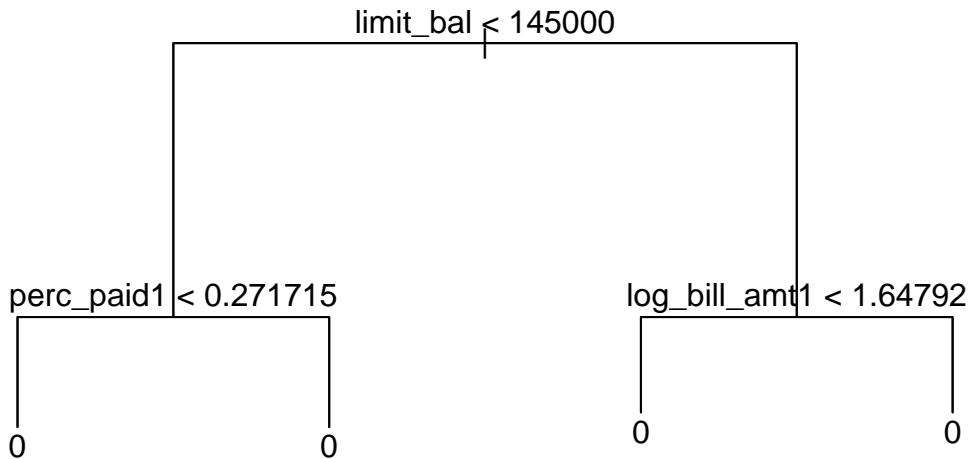
Principle Component Analysis (PCA) is an unsupervised dimension reduction technique. We look at the features in question and seek to create a number of linear combinations of these features. Of course, the number of combinations must be less than the number of original features. These combinations are referred 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 F: 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 referred to as trees because they 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 anecdotal evidence as to why 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 term Random Forest.

Appendix G: Variable List

The following variables were originally given.

- limit_bal: Amount of given credit in dollars, including individual and family/supplementary credit
- sex: Gender (1=male, 2=female)
- education: (1=graduate school, 2=university, 3=high school, 4=others)
- marriage: Marital status (1=married, 2=single, 3=others)
- age: Age in years
- bill_amt1: Amount of bill statement in September, 2005 (dollars)
- bill_amt2: Amount of bill statement in August, 2005 (dollars)
- bill_amt3: Amount of bill statement in July, 2005 (dollars)
- bill_amt4: Amount of bill statement in June, 2005 (dollars)
- bill_amt5: Amount of bill statement in May, 2005 (dollars)
- bill_amt6: Amount of bill statement in April, 2005 (dollars)
- pay_amt1: Amount of previous payment in September, 2005 (dollars)
- pay_amt2: Amount of previous payment in August, 2005 (dollars)

- pay_amt3: Amount of previous payment in July, 2005 (dollars)
- pay_amt4: Amount of previous payment in June, 2005 (dollars)
- pay_amt5: Amount of previous payment in May, 2005 (dollars)
- pay_amt6: Amount of previous payment in April, 2005 (dollars)