Springboard Foundations of Data Science Capstone Project

Eitan Boral

Predicting the Accuracy of Credit Card Default

Introduction

Credit cards are a large part of our financial lives in the US. What started out as a niche product mostly for business transactions has spread to every segment of the population. In fact, data from the Federal Reserve shows that total revolving debt has exploded from just over $1.3 billion in 1968 to nearly $1 trillion at the end of 2016, an annual growth rate of more than 18%. Outstanding Revolving Debt

The Problem

With such a large market of borrowers, credit card issuers have to find effective but scalable solutions to screen potential cardholders for their ability and likelihood to pay off their debt. Failure to do so can mean outstanding loans are never paid off, and an issuer has to write-off the loan. At the end of 2016, the average credit card write-off rate was 3.32% for the top 100 US banks ranked by assets. Write off Data

In addition, for households that carry an unpaid credit card balance each month, the average debt was just over $8,000 at the end of 2016, a 6% increase from 2015. Household Balance

Considering how quickly the use of revolving debt has grown, and the increasing unpaid balance of households, a predictive model of default can be very valuable to credit card issuers.

Our goal in this analysis is to help financial institutions reduce their credit card loan write-off rates by identifying the traits of borrowers most and least likely to default. We will attempt to build a model that can explain which factors have predictive power.

Data Set

While not focused on the US market, the data set for this project looked at one response variable, default or payment of credit card debt for 30,000 Taiwanese credit card holders in October 2005.

- Client action (0 = no default, 1 = default)

The following 23 independent variables were also included.

- 1: Credit limit available to the borrower (ranging from $10,000 to $1,000,000)
- 2: Gender (1 = male; 2 = female)
- 3: Education level (1 = graduate school; 2 = university; 3 = high school; 0, 4, 5, 6 = other)
- 4: Marriage status (1 = married; 2 = single; 3 = divorce; 0 = other)
- 5: Age in years (ranging from 21 to 79)
- 6-11: History of past payment (six variables, April to September of 2005)
  - -2 = No consumption (card holder has nothing due and didn't access line of credit)
  - -1 = Paid in full (previous balance has been paid off)
  - 0 = Revolving credit (previous balance not fully paid off)
  - 1 = Payment delay for 1 month; 2 = payment delay for 2 months;. . . ; 9 = payment delay for 9 months and above
- 12-17: Monthly balance (six variables, April - September of 2005)

- 18-23: Amount of previous monthly payment (six variables, April - September of 2005)

Data is available at the following link: Data set

Data Limitations

Two variables not included in the data set are the credit score and annual income of the borrower. Both of these variables are considered important data points for financial institutions that issue credit cards. Any conclusions from this study should take into account that including these two variables may have led to different results.

Another limitation is the study's narrow focus on the Taiwanese credit card market. The health of the Taiwanese economy, domestic interest rates at the time of the study, and local political or cultural factors may have had an effect on the default rate that may be less relevant in other populations.

Additionally, interest assessed on outstanding debt is not included in the data set. While this study attempts to assign probabilities of default to different types of loans, a more complete analysis would combine the likelihood of default with potential return. This would give card issuers the ability to predict expected profit for different types of loans.

Finally, 2005 was a period where many Taiwanese financial institutions greatly expanded credit card issuance with little to no regard for the ability of borrowers to repay. In addition, cardholders accumulated higher than average balances during this period. This resulted in abnormally high default rates in the Taiwanese market. Credit card lending in Taiwan during this period was similar in some ways to the mortgage loan market in the US leading up to the 2008 housing crisis. Any conclusions about who fits the profile of a risky cardholder must take into account the lax lending practices of issuers during the period of this study.

Data Wrangling

The data set was comprehensive with no missing values. It required some simple cleanup and reformatting. The steps taken are described below. For a detailed review of code, an R markdown document explaining each step with code is included.

Columns and rows not needed for analysis were removed. For example, an ID column and a row duplicating the name of each variable were removed. In addition, column headings in the data set needed to be transformed from ambiguous names such as X1, X2 etc. to names which give a clear description of each variable such as Limit Amount and Gender.

After importing the data set into R, it was converted from a tibble to a data frame, and all variables were converted from a character to numeric type. Both of these transformations were needed in order to conduct preliminary exploration and data visualization.

Each variable had multiple values. Some of these represented the same value and needed to be merged together. For example, in the Education column, the values 0, 4, 5, and 6 all represented "Other". We merged variables 4, 5, and 6 to 0 in order to simplify the analysis.

Depending on the type of analysis or visualization being created, some variables needed to have the factor type, while other needed the numeric type. Instead of transforming the types back and forth which has a high chance of leading to error, we created duplicate columns of certain existing variables and converted them to factor. For example, Gender and Education existed as numeric. Two new columns called Gender1 and Education1 were created and then converted to factor.

Values for each variable were renamed to give a better description of the possibilities for each variable. For example, 1 and 2 were renamed to Male and Female for the Gender variable.

Finally, we created bins for variables that had continuous values (monthly balance, Age, Limit Amount etc). We segmented into bins in order to study how each segment effects the default rate differently. The

distribution of data points with the highest concentration was split into smaller bin sizes, while those with fewer data points had wider bin sizes. For example, there were far more borrowers with lower credit limits than those with higher credit limits. As a result, there were more bins representing cardholders with lower limits.

Preliminary Exploration

The goal of preliminary exploration was to identify independent variables that appear to have some predictive power of default. Our baseline for comparison is simply the average default rate across the data set.

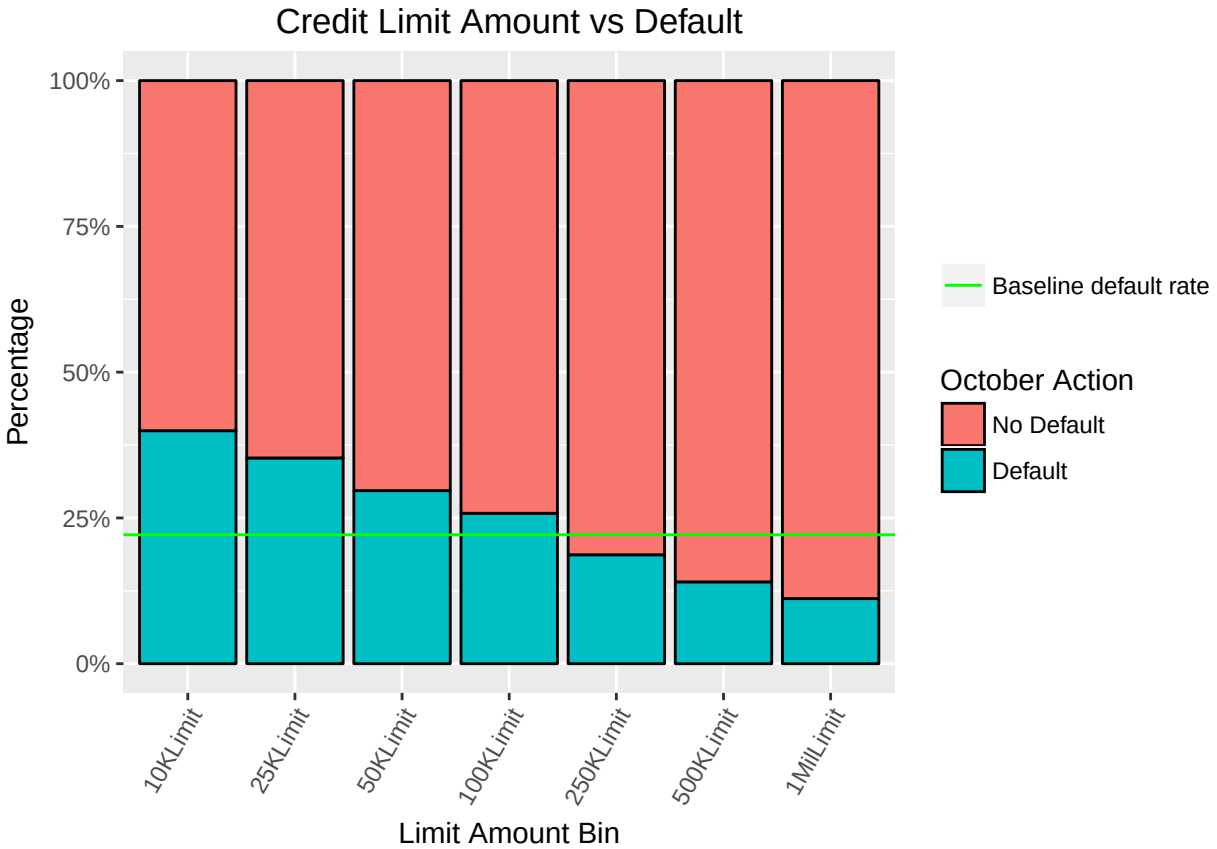
```
table(dcc$DefaultOct05)/length(dcc$DefaultOct05)
```

```
##
##      0      1
## 0.7788 0.2212
```

Across 30,000 loans, 22.12% default and 77.88% do not default. We examined the default rate for each independent variable and compared the results to our baseline.

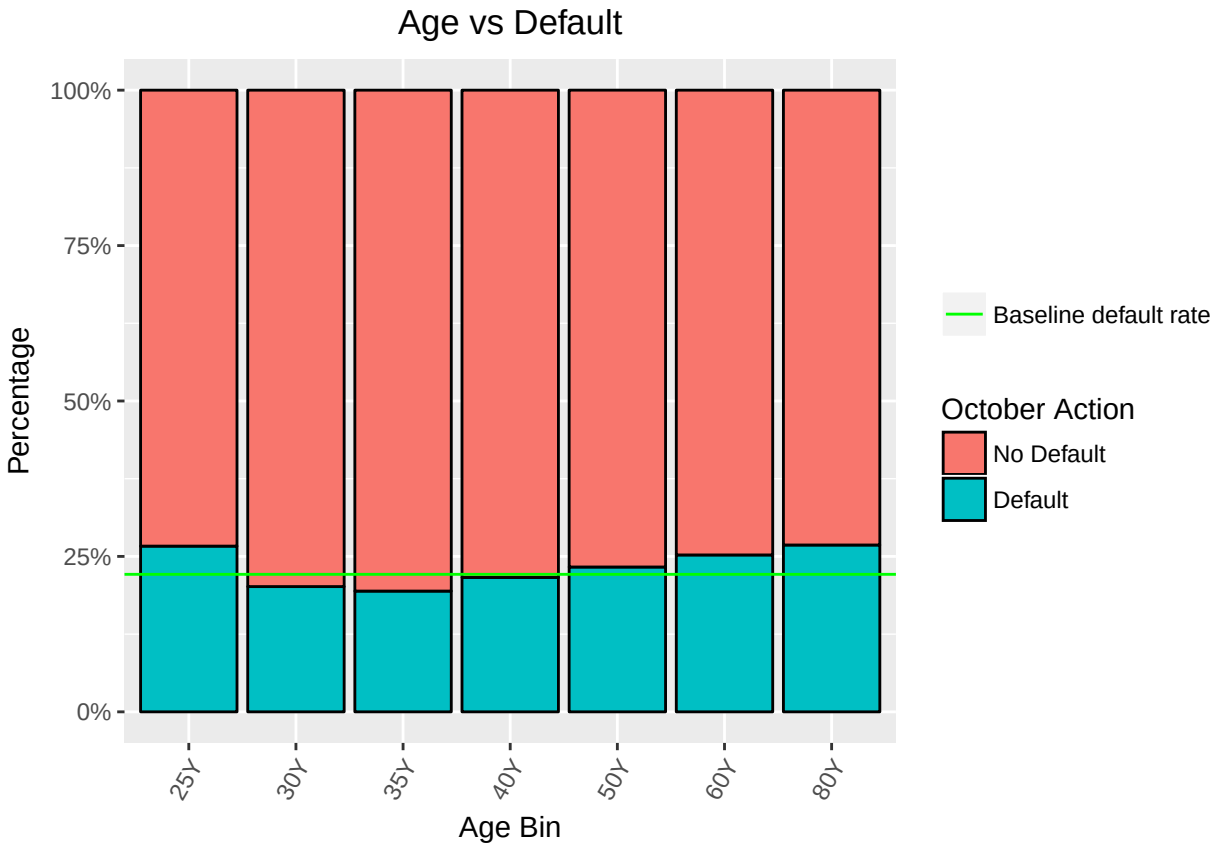
Below are the graphs for those variables that appear to have a strong influence on default.

Credit Limit



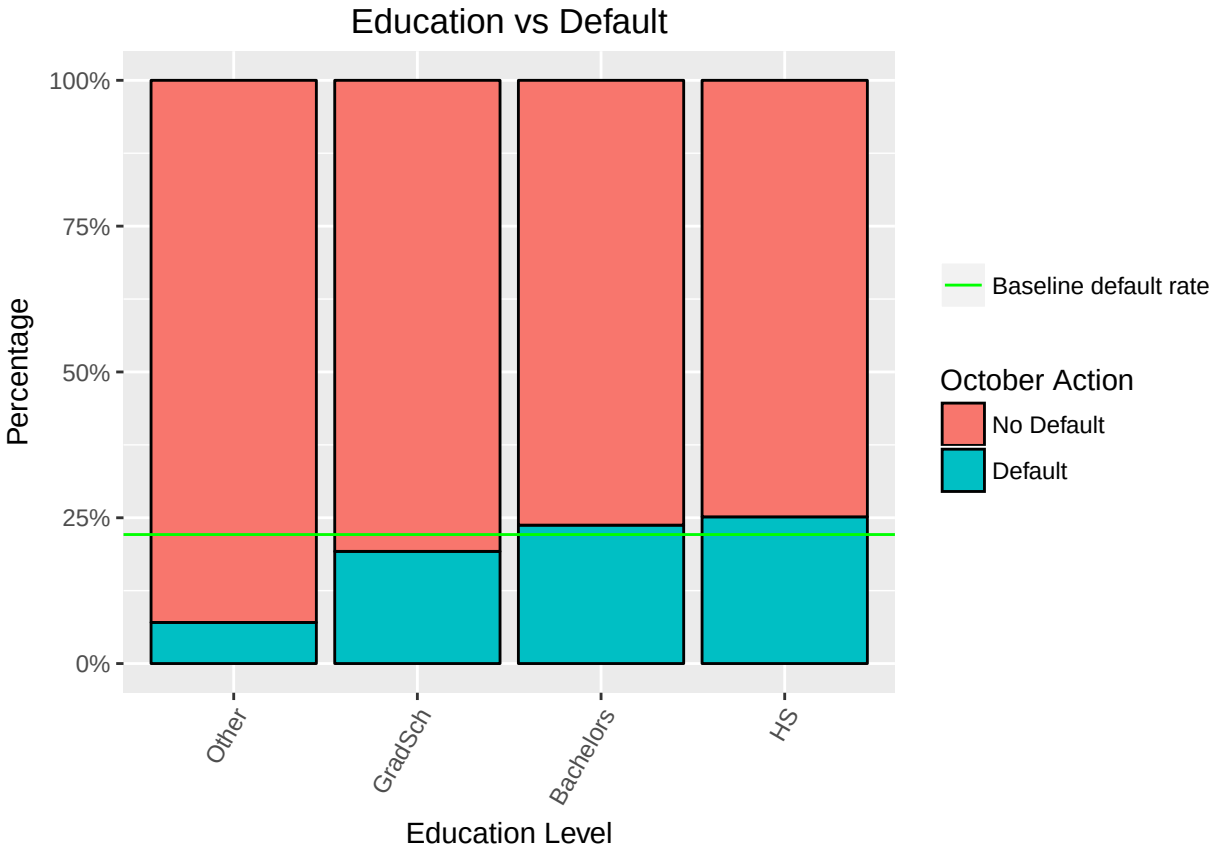
Those with the lowest credit limits have the highest default rates, and those with the highest credit limits have the lowest default rates. Default rates are near 40.0% for the 10K bucket, and decline with each increasing limit bin. The largest credit limit sees a default rate of just over 11.0%.

Age



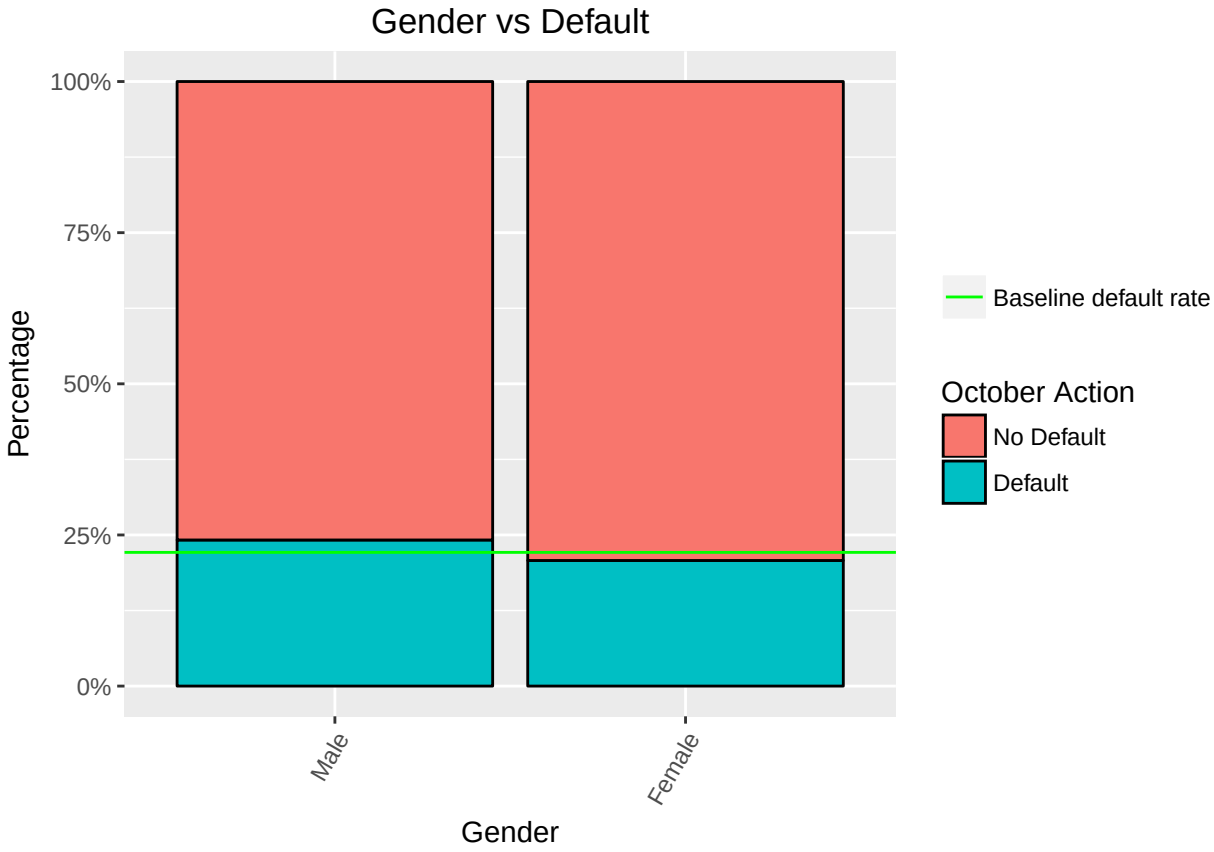
The age breakdown shows a different trend. The youngest and oldest bins (21 to 25 and those between 61 to 80) have the highest default rates, both approaching 27.0%. However, middle-age borrowers (31-35) have the lowest default rate of 19.42%. Default rates based on age have a V shape. At the extremes, loans appear to be riskier. Loans to the middle age crowd appear to be safer.

Education



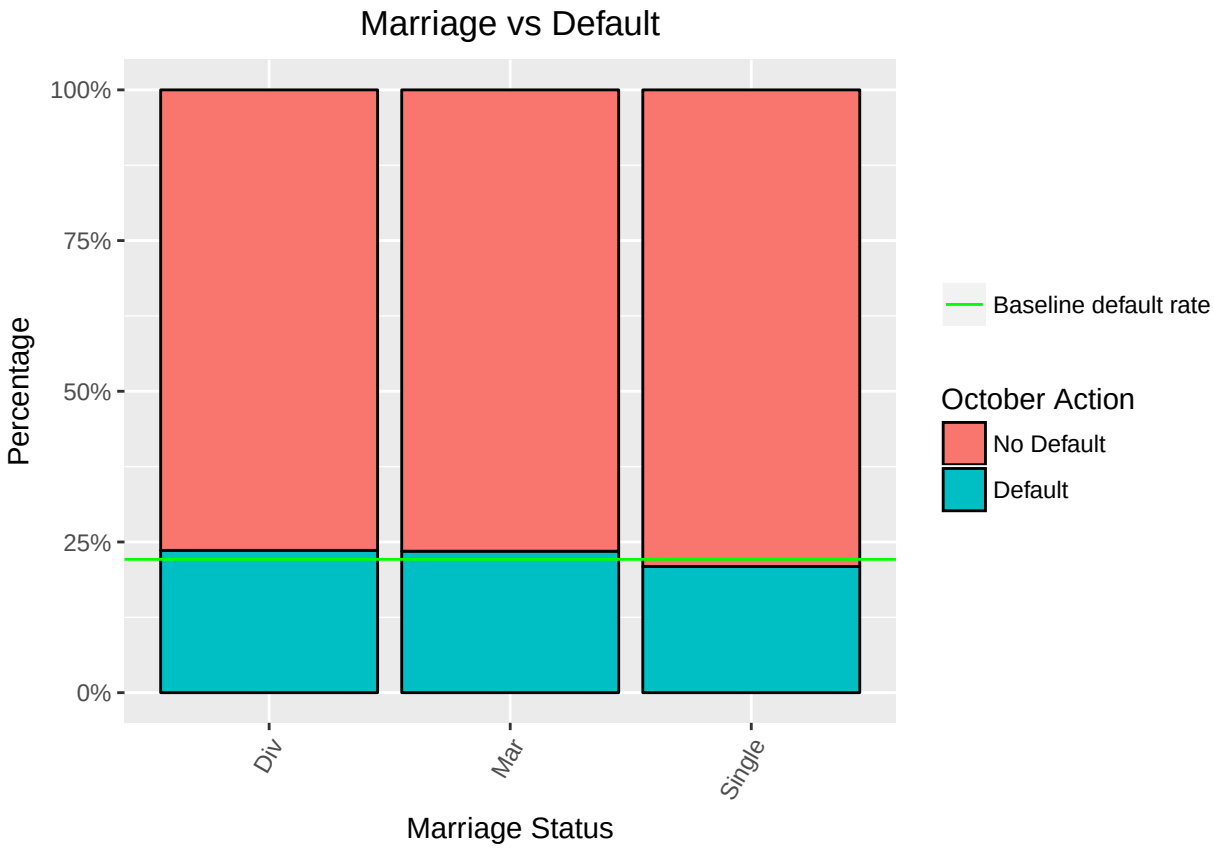
Education shows an interesting trend that intuitively makes sense. The higher the level of education, the lower the default rate. For example, those with only a high school education default over 25% of the time, while those with only a bachelors degree have a default rate of 23.73%, which is higher than our baseline, but still an improvement over high school. Finally, those with a graduate degree have the lowest default rate of 19.23%.

Gender



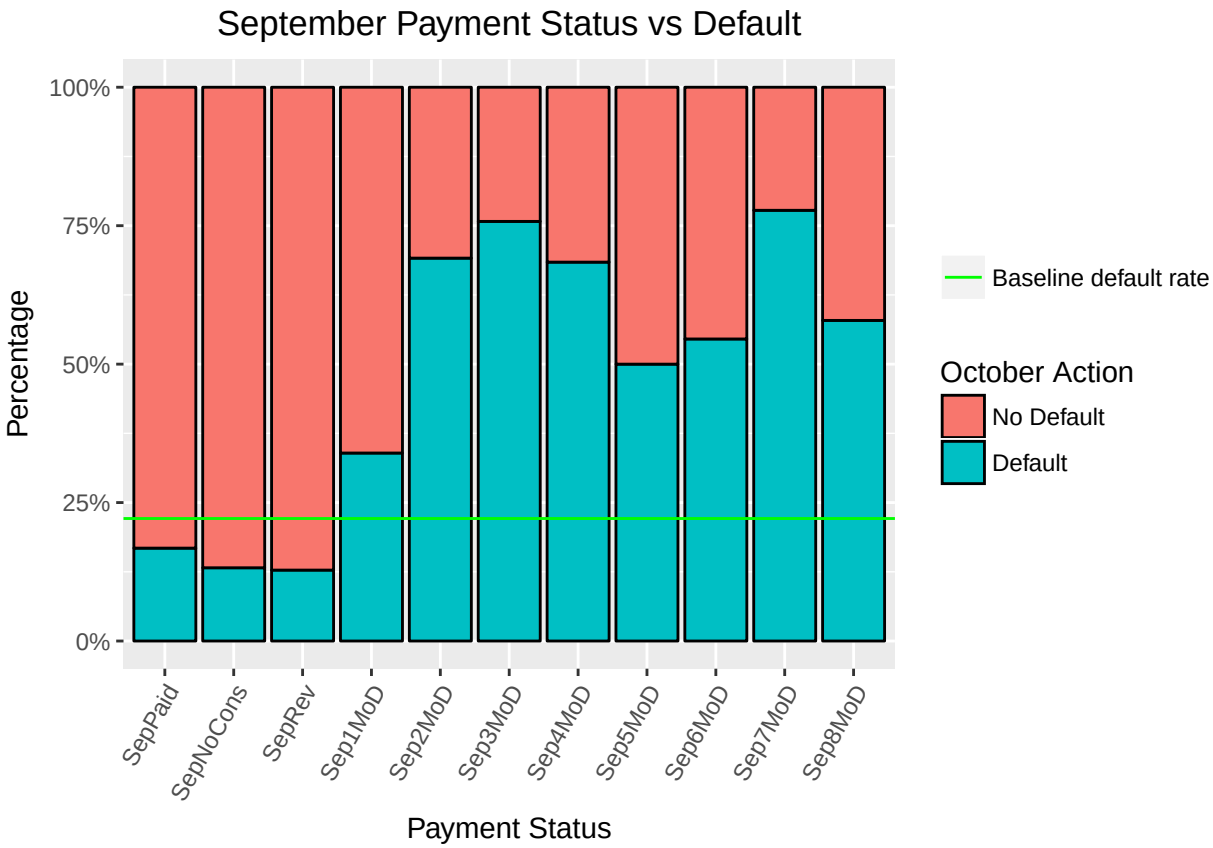
Males have a 24.24% default rate while females have a 20.73% default rate. There is a wide disparity between the two groups, with males well above the baseline default rate and females well below.

Marriage Status



Marriage status provides a mixed picture. Both married and divorced borrowers default around 23.5% of the time while single borrowers who haven't been married default 20.92% of the time.

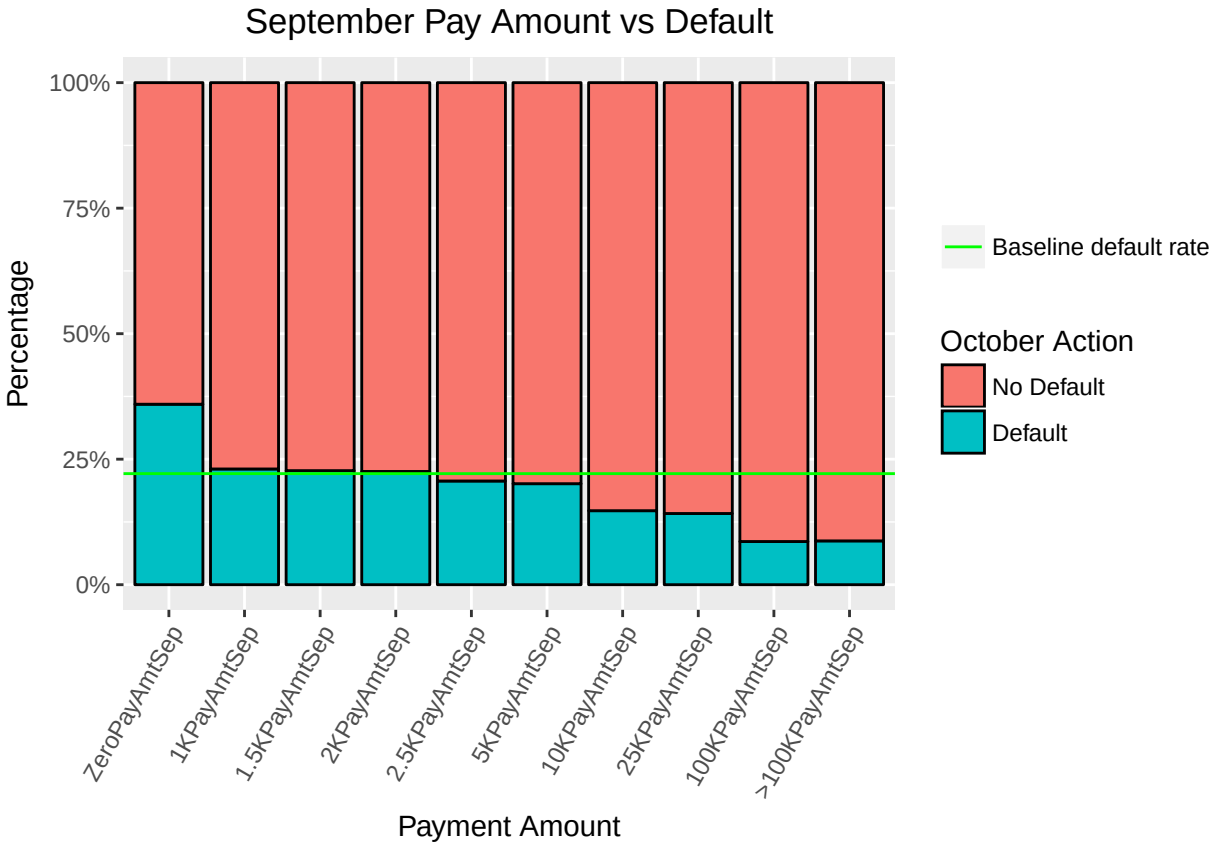
Monthly Payment Status



Only September is shown since all six months have very similar default rates for each payment status. The payment status shows a much lower default rate than the baseline for those that fall into the No Consumption, Paid in full or Revolving bins. These bins represent borrowers that are prompt in paying off their monthly debt.

Loans that are two or more months behind see default rates that are 50% or higher regardless of what month they are behind. We see that there is no middle ground when it comes to payment status. If a cardholder is behind they are much more likely to default compared to the baseline. If they are on time, they are much less likely to default compared to the baseline.

Monthly Payment Amount



Only September is shown since all six months have very similar default rates for each payment amount. We see a very clear trend with the payment amount variables. The lower the payment, the higher the default rate. As payment amounts reached the 100K and >100K amounts, default rates hovered around 8%, well below the baseline. Clearly the higher the monthly payment amount, the lower the default rate.

Machine Learning

In this section we used machine-learning methods to build three different models that predict the probability of either a default or no default across the loans in the data set. We built a logistic regression model, a classification and regression tree (CART), and a random forest model.

We saw earlier that the default rate across all loans is 22.12%, which means that 77.88% of loans are not defaulting. Our goal was to improve upon the baseline with each model.

Logistic Regression

We first split the data into a training data set and testing data set. The training set was used to discover potentially predictive relationships. The testing set was used to assess the performance of these relationships.

Our training set used 75% of the observations, and our testing set used 25%. The set.seed variable was used to make sure the dependent variable (default) was well balanced in both the training and testing sets.

```
# Split data into training and testing set with a 75/25 ratio
set.seed(64)
split = sample.split(dcc$DefaultOct05, SplitRatio = 0.75)
dccTrain = subset(dcc, split == TRUE)
dccTest = subset(dcc, split == FALSE)
```

```r
# Check rows for both the training and testing set
nrow(dccTrain)
```

```
## [1] 22500
```

```r
nrow(dccTest)
```

```
## [1] 7500
```

We see there are 22,500 rows in the training data set and 7,500 in the testing data set, which is consistent with a 75/25 split.

We used a backward selection approach in building the logistic regression model, which starts with a large number of independent variables, and removes those with no significance.

Model1

```r
# Model 1 starts with almost all the independent variables
dccLog1 = glm(DefaultOct05 ~ LimitAmt + Gender + Education + Marriage + Age +
                StatusSep05 + StatusAug05 + StatusJul05 + StatusJun05 + StatusMay05 +
                StatusApr05 + PayAmtSep05 + PayAmtAug05 + PayAmtMay05 + PayAmtApr05 +
                BalSep05 + BalAug05 + BalJul05 + BalJun05 + BalMay05 + BalApr05,
              data=dccTrain, family=binomial)

summary(dccLog1)
```

```
##
## Call:
## glm(formula = DefaultOct05 ~ LimitAmt + Gender + Education +
##     Marriage + Age + StatusSep05 + StatusAug05 + StatusJul05 +
##     StatusJun05 + StatusMay05 + StatusApr05 + PayAmtSep05 + PayAmtAug05 +
##     PayAmtMay05 + PayAmtApr05 + BalSep05 + BalAug05 + BalJul05 +
##     BalJun05 + BalMay05 + BalApr05, family = binomial, data = dccTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1067  -0.6993  -0.5482  -0.2893   3.5383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.687e-01  1.370e-01  -6.342 2.27e-10 ***
## LimitAmt    -6.821e-07  1.804e-07  -3.781 0.000156 ***
## Gender      -1.108e-01  3.540e-02  -3.129 0.001752 **
## Education   -1.647e-02  2.566e-02  -0.642 0.520960
## Marriage    -1.446e-01  3.644e-02  -3.969 7.22e-05 ***
## Age          6.969e-03  2.048e-03   3.403 0.000666 ***
## StatusSep05  5.740e-01  2.044e-02  28.078  < 2e-16 ***
## StatusAug05  9.571e-02  2.331e-02   4.105 4.04e-05 ***
## StatusJul05  7.012e-02  2.611e-02   2.686 0.007235 **
## StatusJun05  1.385e-02  2.885e-02   0.480 0.631193
## StatusMay05  3.701e-02  3.095e-02   1.196 0.231738
## StatusApr05  1.471e-02  2.561e-02   0.574 0.565730
## PayAmtSep05 -1.811e-05  2.971e-06  -6.096 1.09e-09 ***
## PayAmtAug05 -7.498e-06  2.249e-06  -3.335 0.000854 ***
## PayAmtMay05 -7.128e-06  2.216e-06  -3.216 0.001299 **
## PayAmtApr05 -2.125e-06  1.505e-06  -1.413 0.157788
```

```
## BalSep05    -6.055e-06  1.304e-06  -4.642 3.45e-06 ***
## BalAug05     4.127e-06  1.703e-06   2.423 0.015387 *
## BalJul05     7.717e-07  1.353e-06   0.570 0.568505
## BalJun05     8.548e-07  1.214e-06   0.704 0.481495
## BalMay05    -2.965e-06  1.644e-06  -1.803 0.071374 .
## BalApr05     2.420e-06  1.441e-06   1.679 0.093164 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23779  on 22499  degrees of freedom
## Residual deviance: 20891  on 22478  degrees of freedom
## AIC: 20935
##
## Number of Fisher Scoring iterations: 6
```

In the first model, 7 variables were not significant. We removed these variables and reran the model.

Model2

```
# Model 2 removes insignificant variables
dccLog2 = glm(DefaultOct05 ~ LimitAmt + Gender + Marriage + Age +
                StatusSep05 + StatusAug05 + StatusJul05 + PayAmtSep05 + PayAmtAug05 +
                PayAmtMay05 + BalSep05 + BalAug05 + BalMay05 + BalApr05,
              data=dccTrain, family=binomial)

summary(dccLog2)
```

```
##
## Call:
## glm(formula = DefaultOct05 ~ LimitAmt + Gender + Marriage + Age +
##     StatusSep05 + StatusAug05 + StatusJul05 + PayAmtSep05 + PayAmtAug05 +
##     PayAmtMay05 + BalSep05 + BalAug05 + BalMay05 + BalApr05,
##     family = binomial, data = dccTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1097  -0.6972  -0.5480  -0.2919   3.5601
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.028e-01  1.298e-01  -6.954 3.54e-12 ***
## LimitAmt    -7.490e-07  1.725e-07  -4.343 1.40e-05 ***
## Gender      -1.116e-01  3.535e-02  -3.156 0.001598 **
## Marriage    -1.426e-01  3.622e-02  -3.936 8.30e-05 ***
## Age          6.731e-03  2.012e-03   3.344 0.000825 ***
## StatusSep05  5.799e-01  2.029e-02  28.577  < 2e-16 ***
## StatusAug05  1.015e-01  2.305e-02   4.405 1.06e-05 ***
## StatusJul05  1.021e-01  2.154e-02   4.743 2.11e-06 ***
## PayAmtSep05 -1.867e-05  2.970e-06  -6.287 3.24e-10 ***
## PayAmtAug05 -6.270e-06  1.989e-06  -3.152 0.001621 **
## PayAmtMay05 -7.891e-06  2.196e-06  -3.594 0.000326 ***
## BalSep05    -6.296e-06  1.301e-06  -4.839 1.30e-06 ***
## BalAug05     4.934e-06  1.441e-06   3.425 0.000614 ***
```

```
## BalMay05    -2.397e-06  1.467e-06  -1.634 0.102188
## BalApr05     3.054e-06  1.403e-06   2.176 0.029539 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23779  on 22499  degrees of freedom
## Residual deviance: 20903  on 22485  degrees of freedom
## AIC: 20933
##
## Number of Fisher Scoring iterations: 6
```

We see an improvement in the AIC score from 20935 to 20933, but one of the remaining variables was not significant. Once again, we removed this variable and reran the model.

Model3

```
# Model 3 removes one more that is insignificant
dccLog3 = glm(DefaultOct05 ~ LimitAmt + Gender + Marriage + Age +
                StatusSep05 + StatusAug05 + StatusJul05 + PayAmtSep05 + PayAmtAug05 +
                PayAmtMay05 + BalSep05 + BalAug05 + BalApr05,
              data=dccTrain, family=binomial)

summary(dccLog3)
```

```
##
## Call:
## glm(formula = DefaultOct05 ~ LimitAmt + Gender + Marriage + Age +
##     StatusSep05 + StatusAug05 + StatusJul05 + PayAmtSep05 + PayAmtAug05 +
##     PayAmtMay05 + BalSep05 + BalAug05 + BalApr05, family = binomial,
##     data = dccTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1082  -0.6969  -0.5477  -0.2931   3.6157
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.031e-01  1.298e-01  -6.957 3.47e-12 ***
## LimitAmt    -7.727e-07  1.719e-07  -4.495 6.97e-06 ***
## Gender      -1.119e-01  3.534e-02  -3.165 0.001549 **
## Marriage    -1.426e-01  3.623e-02  -3.936 8.30e-05 ***
## Age          6.772e-03  2.012e-03   3.365 0.000764 ***
## StatusSep05  5.798e-01  2.029e-02  28.571  < 2e-16 ***
## StatusAug05  1.012e-01  2.305e-02   4.392 1.12e-05 ***
## StatusJul05  1.017e-01  2.153e-02   4.723 2.33e-06 ***
## PayAmtSep05 -1.888e-05  2.974e-06  -6.348 2.18e-10 ***
## PayAmtAug05 -6.723e-06  1.973e-06  -3.406 0.000658 ***
## PayAmtMay05 -6.319e-06  1.970e-06  -3.207 0.001340 **
## BalSep05    -6.385e-06  1.303e-06  -4.900 9.57e-07 ***
## BalAug05     4.659e-06  1.434e-06   3.249 0.001156 **
## BalApr05     1.036e-06  6.431e-07   1.610 0.107379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23779  on 22499  degrees of freedom
## Residual deviance: 20906  on 22486  degrees of freedom
## AIC: 20934
##
## Number of Fisher Scoring iterations: 6
```

The April Balance amount was insignificant. After removing this variable we reran the model once more, however the AIC score was slightly higher (moving from 20933 to 20934)

Model4

```
# Model 4 shows all remaining variables with significance
dccLog4 = glm(DefaultOct05 ~ LimitAmt + Gender + Marriage + Age +
                StatusSep05 + StatusAug05 + StatusJul05 + PayAmtSep05 + PayAmtAug05 +
                PayAmtMay05 + BalSep05 + BalAug05,
              data=dccTrain, family=binomial)

summary(dccLog4)
```

```
##
## Call:
## glm(formula = DefaultOct05 ~ LimitAmt + Gender + Marriage + Age +
##     StatusSep05 + StatusAug05 + StatusJul05 + PayAmtSep05 + PayAmtAug05 +
##     PayAmtMay05 + BalSep05 + BalAug05, family = binomial, data = dccTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1156  -0.6971  -0.5483  -0.2933   3.6443
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.058e-01  1.298e-01  -6.979 2.97e-12 ***
## LimitAmt    -7.458e-07  1.709e-07  -4.364 1.28e-05 ***
## Gender      -1.102e-01  3.533e-02  -3.120 0.001811 **
## Marriage    -1.431e-01  3.622e-02  -3.952 7.75e-05 ***
## Age          6.754e-03  2.012e-03   3.357 0.000788 ***
## StatusSep05  5.813e-01  2.027e-02  28.673  < 2e-16 ***
## StatusAug05  1.014e-01  2.304e-02   4.400 1.08e-05 ***
## StatusJul05  1.042e-01  2.148e-02   4.850 1.23e-06 ***
## PayAmtSep05 -1.926e-05  2.966e-06  -6.494 8.34e-11 ***
## PayAmtAug05 -6.315e-06  1.950e-06  -3.238 0.001202 **
## PayAmtMay05 -5.568e-06  1.901e-06  -2.930 0.003394 **
## BalSep05    -6.420e-06  1.297e-06  -4.949 7.46e-07 ***
## BalAug05     5.395e-06  1.352e-06   3.991 6.57e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23779  on 22499  degrees of freedom
## Residual deviance: 20908  on 22487  degrees of freedom
## AIC: 20934
```

```
##
## Number of Fisher Scoring iterations: 6
```

Model 4 appeared to be the best mix of independent variables as it delivered an AIC score of 20934, and showed significance of between 0 and 0.001 or between 0.001 and 0.01 for all independent variables. The coefficients for Age, payment status, and the August balance variable are positive, which indicates that higher values are indicative of a higher chance of default.

Predict the data

By using a threshold value, we can convert our probabilities to predictions. If the probability of default is greater than the threshold, then we predict default. If it's below, then we predict no default.

Using a threshold of 0.5, we examined how our model works with the test data set.

```
# Prediction on Test data
predictdccTest = predict(dccLog4, type="response", newdata = dccTest)
table(dccTest$DefaultOct05, predictdccTest > 0.5)
```

```
##
##     FALSE TRUE
##   0  5682  159
##   1  1251  408
```

The matrix above shows the breakdown of all defaults. The rows are the acutual result (0 = no default, 1 = default), the columns are the predictions. We see that out of 7,500 occurrences, we have 6090 correct predictions (5682 loans predicted to be paid off, 408 predicted to default).

However, 1251 loans did default that were predicted to be paid off. We also see that of the loans predicted to default, 159 were actually paid off.

Our overall accuracy measures the number of correct predictions divided by the total number of loans.

```
# Check for accuracy of model
TestAccuracy = (5682 + 408)/(5682 + 408 + 1251 + 159)
TestAccuracy
```

```
## [1] 0.812
```

The accuracy of the model is 81.2%.

This tells us that our model is a better predictor of default compared to a random guess, or our baseline (77.88%).

Classification Tree

While logistic regression shows how an independent variable may be predictive for a specific outcome, it's difficult to understand which factors are most important, and to evaluate what the prediction will be for a new case.

The classification tree also predicts the probability of a specific outcome. By following a split in the tree, we can predict the most frequent outcome in the training set that followed the same path.

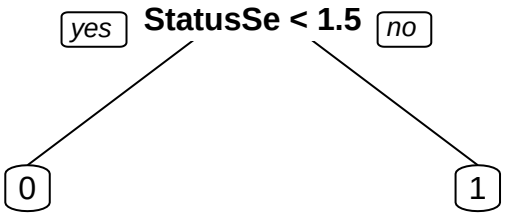
We used the same training and testing as with our logistic regression model.

```
# Split data and create train/test sets
set.seed(64)
split = sample.split(dcc$DefaultOct05, SplitRatio = 0.75)
dccTrain = subset(dcc, split == TRUE)
dccTest = subset(dcc, split == FALSE)
```

We then built the model, examined the tree and its accuracy.

```
# Build classification tree model with all variables
dccTree = rpart(DefaultOct05 ~ LimitAmtBin + Gender1 + Education1 + Marriage1 + AgeBin + StatusSep05 +
                  StatusAug05 + StatusJul05 + StatusJun05 + StatusMay05 + StatusApr05 +
                  PayAmtSep05 + PayAmtAug05 + PayAmtJul05 + PayAmtJun05 + PayAmtMay05 +
                  PayAmtApr05 + BalSep05 + BalAug05 + BalJul05 + BalJun05 + BalMay05 + BalApr05,
                data = dccTrain, method = "class", control = rpart.control(minbucket = 25))

# Look at tree
prp(dccTree)
```



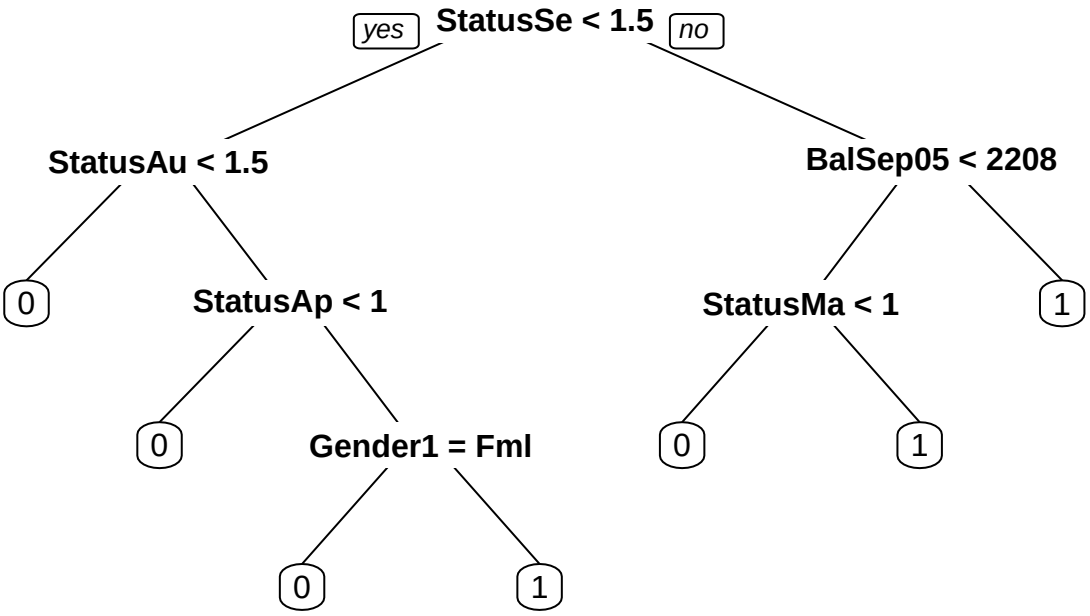
The tree shows the payment status for September variable as its only node. If a loan is less than 1.5 months behind, then it is less likely to default. If a loan is more than 1.5 months behind its more likely to default.

Only one node is in the tree, which may be confusing since our model includes all 23 independent variables. An important note is that the September payment status variable precedes our dependent variable by one month. This may give extra weight to September in the model. In practice, a financial institution would not be limited to the previous six months of data, and there may not be an ending date for every loan since credit card debt is revolving. This may explain why there is only one node for this specific study.

In order to force more nodes in the tree, we can change the complexity parameter (cp). We can look at complexity parameters of .0025 and .001. Then we can compare the accuracy of all three CART models we've created. This may give more insight as to whether a tree with one node is suspicious, or the most accurate predictor of default.

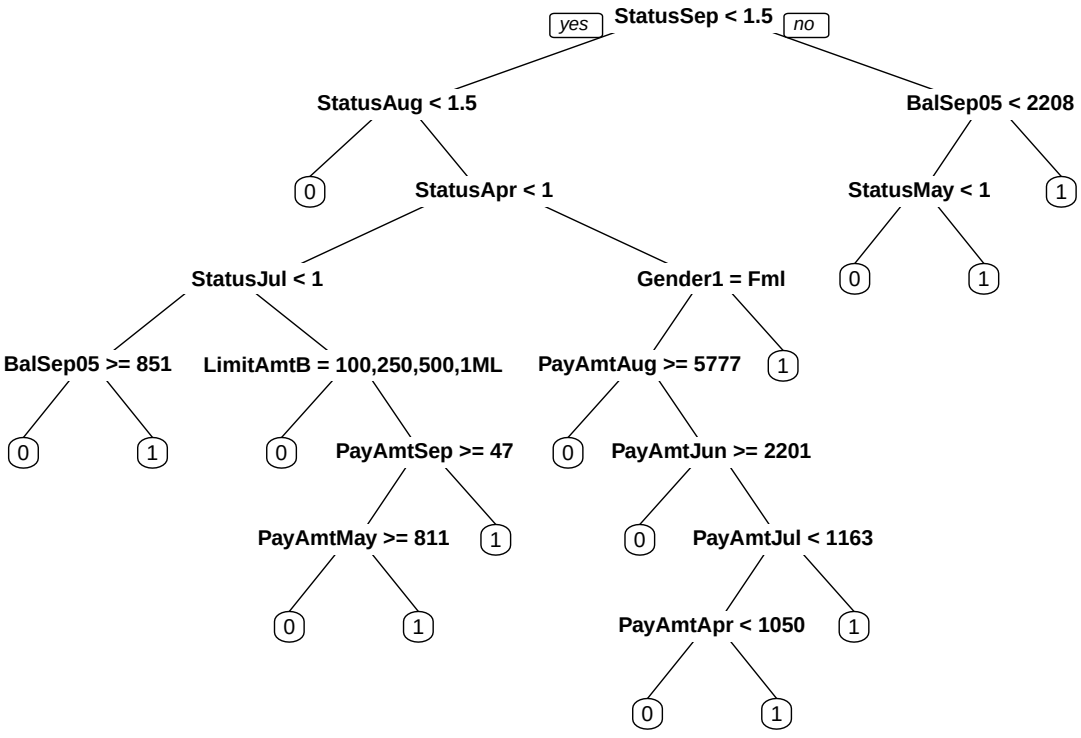
```
# Build classification tree model2 with all variables and cp = .0025
dccTree2 = rpart(DefaultOct05 ~ LimitAmtBin + Gender1 + Education1 + Marriage1 + AgeBin +
                  StatusSep05 + StatusAug05 + StatusJul05 + StatusJun05 + StatusMay05 +
                  StatusApr05 + PayAmtSep05 + PayAmtAug05 + PayAmtJul05 + PayAmtJun05 +
                  PayAmtMay05 + PayAmtApr05 + BalSep05 + BalAug05 + BalJul05 +
                  BalJun05 + BalMay05 + BalApr05,
                data = dccTrain, method = "class",
                control = rpart.control(minbucket = 25, cp = .0025))

# Look at tree
prp(dccTree2)
```




```
# Build classification tree model3 with all variables and cp = .001
dccTree3 = rpart(DefaultOct05 ~ LimitAmtBin + Gender1 + Education1 + Marriage1 + AgeBin +
                   StatusSep05 + StatusAug05 + StatusJul05 + StatusJun05 + StatusMay05 +
                   StatusApr05 + PayAmtSep05 + PayAmtAug05 + PayAmtJul05 + PayAmtJun05 +
                   PayAmtMay05 + PayAmtApr05 + BalSep05 + BalAug05 + BalJul05 +
                   BalJun05 + BalMay05 + BalApr05,
                 data = dccTrain, method = "class",
                 control = rpart.control(minbucket = 25, cp = .001))

# Look at tree
prp(dccTree3)
```



The second model with a cp of .0025 has multiple splits, while the third model with a cp of .001 has an even greater number of splits. The first node for all three models is the September payment status variable. The other payment status variables also appear quite frequently at other nodes.

Finally, we can apply each of the models to our test data.

```
# Apply three models on test set
PredictCARTdcc = predict(dccTree, newdata = dccTest, type = "class")
table(dccTest$DefaultOct05, PredictCARTdcc)
```

```
##    PredictCARTdcc
##        0    1
##   0 5615  226
##   1 1113  546
```

```
PredictCART2dcc = predict(dccTree2, newdata = dccTest, type = "class")
table(dccTest$DefaultOct05, PredictCART2dcc)
```

```
##    PredictCART2dcc
##        0    1
##   0 5594  247
##   1 1097  562
```

```
PredictCART3dcc = predict(dccTree3, newdata = dccTest, type = "class")
table(dccTest$DefaultOct05, PredictCART3dcc)
```

```
##    PredictCART3dcc
##        0    1
##   0 5532  309
##   1 1046  613
```

The confusion matrix shows the accuracy of each model.

```
# Examine accuracy of each model
CAccuracy = (5615 + 546)/(5615 + 546 + 1113 + 226)
CAccuracy
```

```
## [1] 0.8214667
```

```
CA2ccuracy = (5594 + 562)/(5594 + 562 + 1097 + 247)
CA2ccuracy
```

```
## [1] 0.8208
```

```
CA3ccuracy = (5532 + 615)/(5532 + 615 + 1046 + 309)
CA3ccuracy
```

```
## [1] 0.8193815
```

Model one has the highest accuracy, but models two and three are very close behind, and have many more branches. The September Payment Status variable appears to remain quite important in models two and three which have multiple nodes and sacrifice little in terms of accuracy.

Random Forest

Our final approach was the random forest model. Random forest was designed to improve the accuracy of CART by building a large number of CART trees. Random forest selects data randomly with replacement.

As with the previous models, we use the same split for the training and testing data sets. We then tried the model on the testing data set.

```
# Split data, build model, apply to test set
set.seed(64)
split = sample.split(dcc$DefaultOct05, SplitRatio = 0.75)
dccTrain = subset(dcc, split == TRUE)
dccTest = subset(dcc, split == FALSE)

dccTrain$Default1Oct05 = as.factor(dccTrain$Default1Oct05)
dccTest$Default1Oct05 = as.factor(dccTest$Default1Oct05)

dccForest = randomForest(Default1Oct05 ~ LimitAmtBin + Gender + Education + AgeBin + StatusSep05 +
                           StatusAug05 + StatusJul05 + StatusJun05 + StatusMay05 + StatusApr05 +
                           PayAmtSep05 + PayAmtAug05 + PayAmtJul05 + PayAmtJun05 + PayAmtMay05 +
                           PayAmtApr05 +BalSep05 + BalAug05 + BalJul05 + BalJun05 + BalMay05 +
                           BalApr05, data = dccTrain, nodesize = 25, ntree = 200)

PredictdccForest = predict(dccForest, newdata = dccTest)
table(dccTest$DefaultOct05, PredictdccForest)
```

```
##    PredictdccForest
##     No Default Default
##   0       5540     301
##   1       1064     595
```

```
# Examine accuracy
RAccuracy = (5548 + 598)/(5548 + 598 + 1061 + 293)
RAccuracy
```

```
## [1] 0.8194667
```

The accuracy of the random forest model is 81.8%, which is in between the logistic model and CART model.

Conclusions

1. Our three models improved upon the probability of a random loan being paid off (77.88%). Our logistic regression model had an accuracy of 81.2%. The CART model improved upon that number slightly with an accuracy of 82.1%, while the random forest model had an accuracy of 81.95%.
2. While we've been able to improve upon our baseline depending on the model, this is a modest improvement and may not be enough for the credit card issuer to replace any existing loan screening mechanism.
3. Payment status in the month immediately preceding the final month of the study appears to have a high level of significance. Due to the defined time periods of the study, and the revolving nature of credit card debt, this conclusion may need to be investigated further.

Recommendations

1. When assessing the risk of a loan, an important feature for analysis is the potential return, not just the likelihood of default. While this data set did not include the interest rate for each loan, it's important for credit card issuers to factor this data point into the analysis. Both the likelihood of default and potential return of a loan will allow a credit card issuer the ability to model the most and least profitable loans. This is ultimately more important than simply measuring the likelihood of default.
2. Include both the borrower's credit rating and income in future analysis. These are thought to be very important predictors of the likelihood of repayment. If both were included in the analysis, each machine

learning model might have higher accuracy in predicting default.

3. Further examine the payment status in the month preceding the final month of the study. It's important to see if the CART model conclusion is simply a result of the start and end date of the study, or if the specific payment status month is a significant predictor of default.