ETHAN BHOJANI

IB MATHEMATICS STATISTICS OPTION

ON

PREDICTING LOAN DEFAULTS

Overview

This paper seeks to use existing data from lenders to fully understand the myriad of factors that affect the probability that the borrower will default on the loan, also known as the loan being charged off. Further, the paper will utilize the results to build a model, which can be used to predict the chance an borrower defaults, and based on the risk, either deny or grant a loan. In the paper, important code blocks and graphs will be included, but the code will be included in the end as well.The paper will be divided into several sections as follows:

Contents

1] Brief Explanation of Loans and Terms

Before delving into the data and the math behind the paper, note the following important terminology that will be repeated throughout the paper. Also included are the factor names, both in the abbreviated form as they appear in the raw data, the full name, and the definition.

1) Lender: The party that transfers money to another party, with the expectation of being paid back usually with interest (defined later)
2) Borrower: The party that accepts money from the lender
3) Interest: Money paid at a set rate by the borrower to the lender until all the money from the loan is returned
4) Default: When the borrower does not pay back the loan, it is said that the borrower defaulted. Used in conjunction, when a loan is not repaid, the loan is said to be charged off.
5) Loan Amount (loan_amnt): The dollar size of the loan
6) Interest Rate (int_rate): Amount of interest due per period
7) Grade (grade): The credit worthiness of a borrower based on past financial history
8) Employment Length (emp_length): Length of time the borrower has been employed
9) Home Ownership (home_ownership): Whether the borrower has property and how they control it whether it be rent, own, or mortgage
10) Mortgage: When a bank takes ownership of a borrower's home in exchange for a loan. Once the loan is repaid, the ownership of the home is transferred back to the borrower.
11) Annual Income (annual_inc): Income received per year by the borrower
12) Loan Status (loan_status): Whether the loan was charged off (defaulted) or full paid

2] Preparing the Data

The data collected for the purposes of building this model is from the popular loan company, Lending Club - see the reference page for specific links. The data is all the loans made by Lending Club from the year 2007 to 2011.

There are important points to note regarding the data and how it was collected. Because the data is all loans from Lending Club, it is not a sample but a population. Thus, note that it would be trivial to carry out forms of testing since we are not trying to make assumptions on a population based on the sample, rather we already have the sample. Instead, we can use this complete population to train a model and forecast future loan defaults.

Since this data is from Lending Club, we cannot reliably extrapolate this model to other forms of loans, or loans given in different time periods since different times or companies may have different borrower demographics and habits.

The following is the process of cleaning the data. Only the important parts of the code will be included, while the complete code will be attached at the end.

```
#install.packages("openxlsx")
library("openxlsx")
loans <- read.xlsx("loan trial .xlsx", 1, colNames = TRUE)
library("dplyr")
library("stringr")
library("tidyr")
#install.packages("gmodels")
library("gmodels")
head(loans)
colnames(loans)
```

Firstly, we load the data into the R library. Using the head function we can preview the loan data and using the colnames we can find out column names of this dataset, and then subset the important factors.

```
loans2 <- loans[, c(3, 7, 9, 12, 13, 14, 17)]
summary(loans2)
colnames(loans2)
```

After subsetting we can see that our new dataset has the following variables.

```
> summary(loans2)
  loan_amnt          int_rate            grade             emp_length        home_ownership
 Length:42538       Length:42538       Length:42538       Length:42538       Length:42538
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
  annual_inc        loan_status
 Length:42538       Length:42538
 Class :character   Class :character
 Mode  :character   Mode  :character
> colnames(loans2)
[1] "loan_amnt"      "int_rate"       "grade"          "emp_length"     "home_ownership"
[6] "annual_inc"     "loan_status"
```

As visible in the code above, each variable is visible, but in an odd format. For example, the loan amount should be a numeric value, yet all the numbers are formatted as variables. After reformatting this dataset with functions we get:

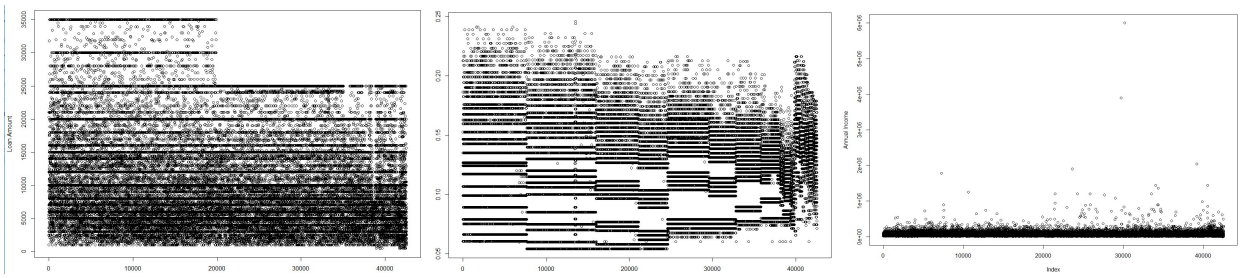
```
> summary(loans4)
   loan_amnt        int_rate       grade          emp_length     home_ownership    annual_inc
 Min.   :  500   Min.   :0.0542   A:10179   10+ years: 9369   MORTGAGE:18959   Min.   :   1896
 1st Qu.: 5200   1st Qu.:0.0963   B:12389   < 1 year : 5058   NONE     :    4   1st Qu.:  40000
 Median : 9700   Median :0.1199   C: 8740   2 years  : 4743   OTHER    :  136   Median :  59000
 Mean   :11090   Mean   :0.1217   D: 6016   3 years  : 4364   OWN      : 3251   Mean   :  69137
 3rd Qu.:15000   3rd Qu.:0.1472   E: 3394   4 years  : 3649   RENT     :20181   3rd Qu.:  82500
 Max.   :35000   Max.   :0.2459   F: 1301   1 year   : 3595                     Max.   :6000000
                                  G:  512   (Other)  :11753

  loan_status
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.1512
 3rd Qu.:0.0000
 Max.   :1.0000
```

Notice how the loan status has changed from characters to 0's and 1's. We have reformatted the data so that 0's represent fully paid loans, and 1's represent a defaulted loan. The loan amount, interest rate, annual income, and loan status are all numerics, while the credit grade, employment rate, home ownership are factors, or categorical variables.

Finally, using this summary, we will remove any extreme data points. Since we have a very large dataset of 50,000 data points, applying the conventional rule of

$1.5 * \textit{Interquartile Range} + Q3 \textit{ and } Q1 - 1.5 * \textit{Interquartile Range}$ will remove many data points not around the median. Rather, we will look at graphs and determine any extreme or improbable values.

```
plot(loans4$loan_amnt, ylab = "Loan Amount")
plot(loans4$int_rate, ylab = "Interest Rate")
plot(loans4$annual_inc, ylab = "Annual Income")
```



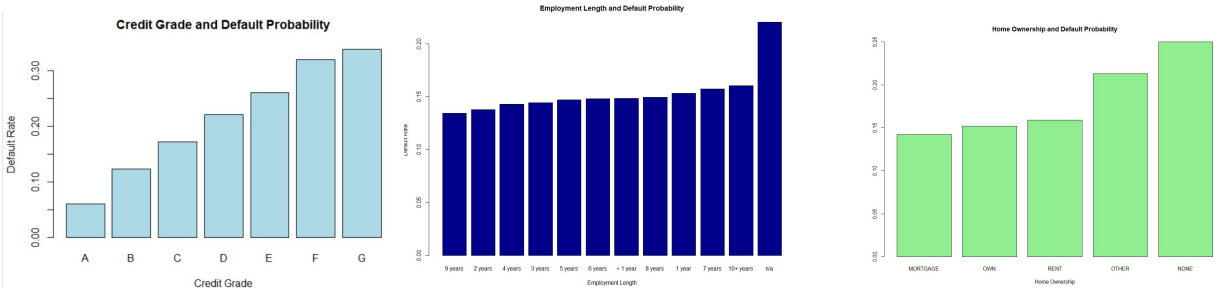
Most notably, the maximum value in the annual income - rightmost graph - is 6 million dollars. It makes little sense for multi millionaires to get small loans from Lending Club, so we will remove these outliers that could skew the results. The others seem centered and without any large outliers.

3] Graphing Variables

This section is devoted to understanding at first, different implications of these variables by graphing them. In later chapters, we will build a GLM - general logistic model - to quantify this relationship between borrower traits and default rate.

Note that with categorical data, this is rather simple, as the relationship can be easily plotted. However with numerical variables, it is hard to find any meaning in a graph of this data considering there are thousands of different numbers in the data, yet the outcome can only be a binomial “0” or “1.”Thus, we will only graph the categorical variables, and wait till we run regression to understand the implications of the numeric variables.

```
grade_plot <- CrossTable(loans4$grade , loans4$loan_status, prop.r = TRUE,
                         prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
grade_rate <- grade_plot$prop.row[,2]
barplot(grade_rate, xlab= "Credit Grade", ylab = "Default Rate", main = "Credit Grade and Default Probability", col = "lightblue")

el_plot <- CrossTable(loans4$emp_length, loans4$loan_status, prop.r = TRUE,
                      prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
el_rate <- el_plot$prop.row[,2]
el_rate2 <- el_rate[order(el_rate)]
barplot(el_rate2, xlab= "Employment Length", ylab = "Default Rate", main = "Employment Length and Default Probability", col = "darkblue")

home_plot <- CrossTable(loans4$home_ownership, loans4$loan_status, prop.r = TRUE,
                        prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
home_rate <- home_plot$prop.row[,2]
home_rate2 <- home_rate[order(home_rate)]
barplot(home_rate2, xlab= "Home Ownership", ylab = "Default Rate", main = "Home Ownership and Default Probability", col = "lightgreen")
```



With these graphs, we can see that as credit grade worsens from "A" down, the default rates rise. This is rather intuitive, the lessening trustworthiness rating would indicate a greater chance a borrower doesn't pay back his debt. The other two variables seem less straightforward. One would think that the longer one works the more reliable that party is, yet the graph shows a pattern where the longer a borrower works the more likely they are to default. Further, one would think that a borrower who owns his house would have less financial pressures and thus less likely to default, yet a borrower who already has his house on mortgage for some reason is less likely to default then a homeowner.

With this information, we can already see how not every variable collected in this dataset is useful. While some have clear implications on default rates, others are unclear and insignificant.

4] Sampling and Building the Training and Testing Data

With this data, we hope to build a model that can accurately predict future defaults. To do this we run regression to get a generalized logistic model. However, first we will introduce the concept of overfitting.

For the sake of explanation, let's take an example of a weather station. Let's say that a statistician has decided to build a model to predict the weather. He takes the forecast data for the past two week and finds the pattern that every wednesday it rains. Now the statistician is excited that he has found a pattern yet when he attempts to model the future, he finds that his predictions are horrible wrong. What happened was the issue of overfitting. If the dataset is too specific and complex, then the model will begin to describe not relationships between variables but the random error in data.

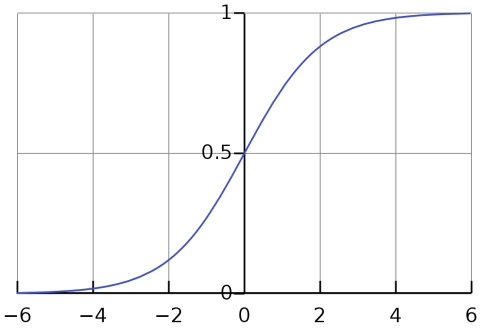
To avoid this problem, we split our data of 42,592 loans into two categories, training and testing data. This way, we can build our model in only the training data, and test it on different loan entries to make sure our model is predicting relationships, not simply remembering irrelevant random errors in data. The following code will randomize the order of loans, then split the data in a set of 32,592 loans in training data, and 10,000 loans in testing data. We will never touch the testing data until we want to evaluate the model, no part of the testing data will be used to build the model.

```
121  loans5 <- loans4[sample(nrow(loans4)),]
122  training <- loans5[1:32529,]
123  testing <- loans5[32530:42529,]
```

5] General Logistic Model and Significance

We will use the general logistic model for this study. It is the most commonly used model for predicting binary events - that is events with only two outcomes "0" and "1," The model is based on logistic regression that would determine the probability of a success, or a "1." While the math behind the GLM is very convoluted and complicated, the following is a simple explanation of logistic regression.

Firstly, let us understand a logistic function. A logistic function is a function that takes an input, $t \, \varepsilon \, |R$ and outputs a value between 0 and 1. The function is $\sigma(t) = \frac{1}{1+e^{-t}}$. The graph is shown below.



Now, let us assume that there are two explanatory variables, $x_1 and \, x_2$ with coefficients $\beta_1 \; and \; \beta_2$ that affect some event x. Then, we can write this as $t = \beta_0 + \beta_1 x_1 and \, \beta_2 x_2$ and to create the equation, $p(x) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x_1 and \beta_2 x_2)}}$ to calculate the probability of event x occurring. β_0 is the intercept, or the default

probability when all other explanatory variables are 0. The generalized logistic model will calculate the optimal coefficients, β_1, β_2 to create a model that is most accurate.

Let us know calculate the general logistic model. In our initial generation:

```
126 glm(formula = loan_status ~ grade, family = "binomial", data = training)
127 glm(loan_status ~ ., family = "binomial", data = training)
128 test_model <- glm(loan_status ~ ., family = "binomial", data = training)
129 summary(test_model)
```

```
Call:
glm(formula = loan_status ~ ., family = "binomial", data = training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2244  -0.6212  -0.5038  -0.3417   3.9469

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)         -3.143e+00  1.230e-01 -25.551  < 2e-16 ***
loan_amnt            2.705e-06  2.431e-06   1.113  0.26585
int_rate             8.661e+00  1.400e+00   6.185 6.21e-10 ***
gradeB               4.906e-01  7.613e-02   6.444 1.16e-10 ***
gradeC               6.296e-01  1.032e-01   6.100 1.06e-09 ***
gradeD               8.060e-01  1.288e-01   6.257 3.92e-10 ***
gradeE               8.973e-01  1.538e-01   5.833 5.45e-09 ***
gradeF               1.018e+00  1.845e-01   5.519 3.42e-08 ***
gradeG               1.012e+00  2.168e-01   4.670 3.01e-06 ***
emp_length1 year     1.489e-02  7.126e-02   0.209  0.83448
emp_length10+ years  1.562e-01  5.869e-02   2.661  0.00778 **
emp_length2 years   -1.557e-01  6.818e-02  -2.284  0.02236 *
emp_length3 years   -3.745e-02  6.874e-02  -0.545  0.58589
emp_length4 years   -8.057e-02  7.249e-02  -1.112  0.26634
emp_length5 years    3.995e-02  7.304e-02   0.547  0.58437
emp_length6 years    1.755e-03  8.274e-02   0.021  0.98308
emp_length7 years    8.549e-02  8.768e-02   0.975  0.32955
emp_length8 years    8.710e-02  9.419e-02   0.925  0.35508
emp_length9 years   -8.760e-02  1.046e-01  -0.838  0.40224
emp_lengthn/a        5.411e-01  9.884e-02   5.474 4.39e-08 ***
home_ownershipNONE  -9.129e+00  1.136e+02  -0.080  0.93597
home_ownershipOTHER  3.932e-01  2.441e-01   1.611  0.10727
home_ownershipOWN   -1.502e-02  6.336e-02  -0.237  0.81257
home_ownershipRENT  -8.818e-03  3.602e-02  -0.245  0.80661
annual_inc          -5.030e-06  4.861e-07 -10.348  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

With this generation of the model, we can see the model generated the coefficients for each explanatory variable, as well as a p value, or a probability of an event occurring at random. The lower the p-value the lower this extreme of a result occurred simply by a chance, so the more significant the results. In the table, significance is denoted by the number of stars. As expected, the credit grade, income, interest rate, and employment grade are strongly correlated to the default rate, while the loan amount and homeownership are not.

In our final model, we will disregard home ownership and the loan amount variables because of their insignificance.

```
134 complete_model <- glm(loan_status ~ int_rate + grade + annual_inc + emp_length, family = "binomial", data = training)
```

6] Forecasting and Types of Accuracy

With our generated model, we can now predict future defaults. Lets begin to evaluate our model with the testing data that we had set aside.

```
137 predict(complete_model, newdata = testing[1,], type = "response")
138 testing$loan_status[1]
```

```
> predict(complete_model, newdata = testing[1,], type = "response")
    36417
0.1924344
> testing$loan_status[1]
[1] 0
```

We can see that for this particular borrower, based on his answers to the explanatory variables, he has a 19.24344% percent chance of defaulting. In reality, the borrower did not default, denoted by the "0" in the loan status.

Lets predict the default rate for the entire testing dataset.

```
143 model_results <- predict(complete_model, newdata = testing, type = "response")
```

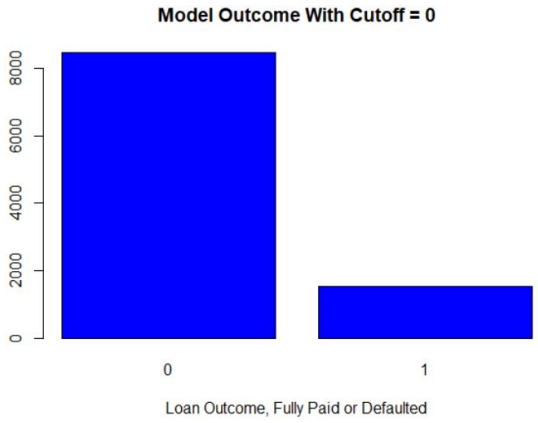
Now that we have an entire list of default rates, lets begin to evaluate our model. Before we begin our evaluation, we need to discuss how loans are approved. Banks and lending institutions will set a cutoff level, or a level that defines the acceptable amount of risk. For example, with a cutoff of 0.25, the bank will grant a loan to any person with a predicted default rate of at or below 25%, and reject any loan application above 25%. The accuracy of this model will vary at different cutoff levels, and we will manipulate the cutoff level to get different results.

To read the barplots attached, the bars are divided into green and blue sections. The x axis displays the actual outcome (default or paid) while the blue bars represent loans the model accepted and green represents the loans the model rejected.

<u>Cutoff = 1.00</u>

At a cutoff of 1, we are saying that anyone with a risk below 100% can get a loan, basically anyone with a heartbeat. The outcome:

```
cutoff <- 1
results_1 <- ifelse(model_results > cutoff,1,0)
sum(results_1)
outcomes_1 <- table(results_1, testing$loan_status)
barplot(outcomes_1, xlab = "Loan Outcome, Fully Paid or Defaulted", col = c(rep(c("blue","green"))), main = "Model Outcome With Cutoff = 0")
accuracy_1 <- sum(diag(outcomes_1)) / nrow(testing)
accuracy_1
accuracy_1d <- 0 / sum(testing$loan_status)
accuracy_1d
```

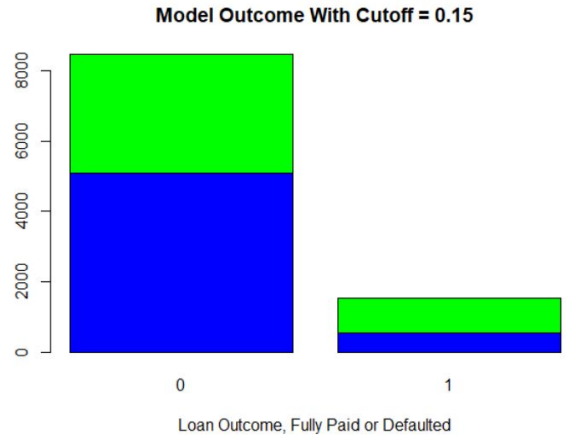



```
> accuracy_1 <- sum(diag(outcomes_1)) / nrow(testing)
> accuracy_1
[1] 0.8456
> accuracy_1d <- 0 / sum(testing$loan_status)
> accuracy_1d
[1] 0
```

From the barplot, we can see that the model accepted all the loans, and out of those loans, more than 8000 were fully paid while around 2000 defaulted. Thus, its accuracy overall is 84.56%. However, accuracy is defined as $\frac{right\ choices}{total\ choices}$ where $right\ choices = correct\ paid\ back + correct\ defaults$. It is likely that it is more important to find a potential default that a potential borrower who will repay the loan ince that represents a loss of money from the bank. We can see that since the model did not accurately identify any of the defaulters and reject them, it has an default accuracy of 0%.

Cutoff = 0.15 More Conservative Model

```
cutoff <- .15
results_15 <- ifelse(model_results > cutoff,1,0)
sum(results_15)
outcomes_15 <- table(results_15, testing$loan_status)
barplot(outcomes_15, xlab = "Loan Outcome, Fully Paid or Defaulted", col = c(rep(c("blue","green"))), main = "Model Outcome With Cutoff = 0.15")
accuracy_15 <- sum(diag(outcomes_15)) / nrow(testing)
accuracy_15
accuracy_15d <- outcomes_15[2,2] / sum(testing$loan_status)
accuracy_15d
```

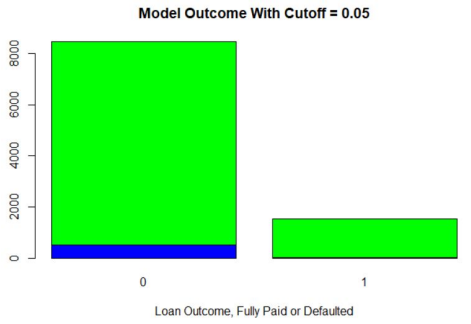



```
> accuracy_15
[1] 0.6092
> accuracy_15d <- outcomes_15[2,2] / sum(testing$loan_status)
> accuracy_15d
[1] 0.6483161
```

As we decrease the loan cutoff, we can see that the model is now beginning to reject applicants. It correctly diagnoses 60.92% of the loans. Its loan default accuracy is also improved at 64.82%.

Cutoff = 0.05, Aggressive Model

```
cutoff <- .05
results_05 <- ifelse(model_results > cutoff,1,0)
sum(results_05)
outcomes_05 <- table(results_05, testing$loan_status)
barplot(outcomes_05, xlab = "Loan Outcome, Fully Paid or Defaulted", col = c(rep(c("blue","green"))), main = "Model Outcome With Cutoff = 0.05")
accuracy_05 <- sum(diag(outcomes_05)) / nrow(testing)
accuracy_05
accuracy_05d <- outcomes_05[2,2] / sum(testing$loan_status)
accuracy_05d
```

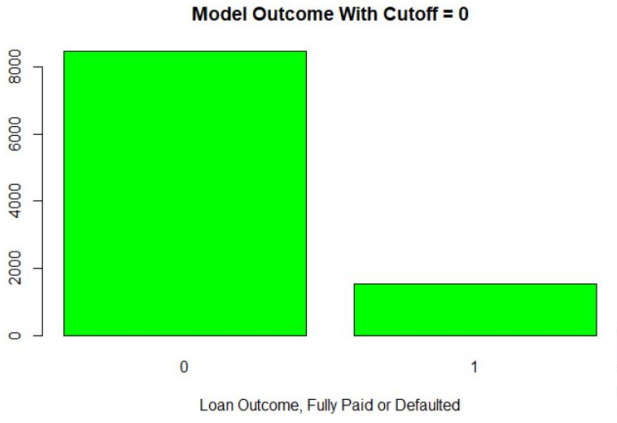



```
> accuracy_05
[1] 0.2049
> accuracy_05d <- outcomes_05[2,2] / sum(testing$loan_status)
> accuracy_05d
[1] 0.990285
```

Now that the cutoff level is extremely low, the model can almost perfectly filter out the future defaulters. It does however come at a large cost of also excluding many loan applicants that would have paid back their loans. The default accuracy was at 99.02%, yet the overall accuracy was only 20.49%.

Cutoff = 0.00

```
cutoff <- .00
results_0 <- ifelse(model_results > cutoff,1,0)
sum(results_0)
outcomes_0 <- table(results_0, testing$loan_status)
barplot(outcomes_0, xlab = "Loan Outcome, Fully Paid or Defaulted", col = "green",  main = "Model Outcome With Cutoff = 0")
accuracy_0 <- outcomes_0[1,2] / nrow(testing)
accuracy_0
accuracy_0d <- outcomes_0[1,2] / sum(testing$loan_status)
accuracy_0d
```




```
> accuracy_0
[1] 0.1544
> accuracy_0d <- outcomes_0[1,2] / sum(testing$loan_status)
> accuracy_0d
[1] 1
```

This cutoff level was rather trivial to do, but it serves to complete our demonstration of the importance of the cutoff rate. This model perfectly removes all the defaulters, yet suffers from a mere 15.44% accuracy and will earn no money since it rejected all its loan applications - since every loan has some chance of defaulting.

7] **Conclusion and Future Implications**

An important discussion touched upon at the end of this paper is the proper cutoff level. While it is important to have a cutoff level small enough that we can properly reject all the future defaulters, it is also important to have an cutoff level high enough that the bank can lend enough money and earn money on interest. Each bank will determine how risky they want their lending strategy to be and select the optimal cutoff level accordingly.

The methods used in this paper are currently in place in virtually every sector. We can see modelling used in risk analysis, be it insurance, lending, trading and stock strategies, or in consulting. Any field with a need to predict or forecast future outcomes utilities modelling to some degree. Most high

level fields also employ much more advanced models used rather than just logistic regression. A more interesting utilization of modelling is its application to the healthcare industry, from insurance to identifying diseases and predicting outbreaks. Machine learning seems to be the next step after modelling, where trends and defining lines are identified without clear rules or paths as seen in statistical modelling.

In all, in this paper, we identified significant explanatory variables that would affect default rates, build a model based on these variables, and then evaluated the model over the testing data. Thus, this paper has shown how data can be used not only to identify important trends, but that using logistic regression and the GLM, we have the ability to quantify these relationships and then apply them to forecast future events. In the process, we have found that interest rate, credit grade, income, employment length are all significant factors, while loan amount and household ownership are insignificant. All this can be used to more efficiently filter loan applications and accept only the optimal ones.

8] Code

```
ls()
rm(list=ls())

getwd()
setwd("C:/Users/ethan/Desktop/loan project/")
list.files()

#install.packages("openxlsx")
library("openxlsx")
loans <- read.xlsx("loan trial .xlsx", 1, colNames = TRUE)
library("dplyr")
library("stringr")
library("tidyr")
#install.packages("gmodels")
library("gmodels")
head(loans)
colnames(loans)

tempDF <- loans
tempDF[] <- lapply(loans, as.character)
colnames(loans) <- tempDF[1, ]
loans <- loans[-1 ,]
tempDF <- NULL
colnames(loans)

loans2 <- loans[, c(3, 7, 9, 12, 13, 14, 17)]
summary(loans2)
colnames(loans2)
loans3 <- loans2
loans3$loan_amnt <- as.numeric(loans2$loan_amnt)
loans3$annual_inc <- as.numeric(loans2$annual_inc)
loans3$int_rate <- as.numeric(loans2$int_rate)
loans3$home_ownership <- as.factor(loans2$home_ownership)
loans3$loan_status <- as.factor(loans2$loan_status)
loans3$grade <- as.factor(loans2$grade)
loans3$emp_length <- as.factor(loans2$emp_length)

summary(loans3)
loans4 <- na.omit(loans3)
summary(loans4)

loans4$loan_status <- sapply(loans4$loan_status, tolower)

temp_list <- loans4$loan_status
temp_list[grepl("paid", loans4$loan_status)] <- 0
##search for charged, replace with 1's
temp_list[grepl("charged", loans4$loan_status)] <- 1
head(temp_list)
loans4$loan_status <- temp_list
temp_list <- NULL
```

```
loans4$loan_status <- as.numeric(loans4$loan_status)

summary(loans4)
CrossTable(loans4$grade , loans4$loan_status, prop.r = TRUE,
           prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
CrossTable(loans4$emp_length, loans4$loan_status, prop.r = TRUE,
           prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
CrossTable(loans4$home_ownership, loans4$loan_status, prop.r = TRUE,
           prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
grade_plot <- CrossTable(loans4$grade , loans4$loan_status, prop.r = TRUE,
                         prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
grade_rate <- grade_plot$prop.row[,2]
barplot(grade_rate, xlab= "Credit Grade", ylab = "Default Rate", main = "Credit
Grade and Default Probability", col = "lightblue")
el_plot <- CrossTable(loans4$emp_length, loans4$loan_status, prop.r = TRUE,
                      prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
el_rate <- el_plot$prop.row[,2]
el_rate2 <- el_rate[order(el_rate)]
barplot(el_rate2, xlab= "Employment Length", ylab = "Default Rate", main =
"Employment Length and Default Probability", col = "darkblue")
home_plot <- CrossTable(loans4$home_ownership, loans4$loan_status, prop.r = TRUE,
                        prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
home_rate <- home_plot$prop.row[,2]
home_rate2 <- home_rate[order(home_rate)]
barplot(home_rate2, xlab= "Home Ownership", ylab = "Default Rate", main = "Home
Ownership and Default Probability", col = "lightgreen")

plot(loans4$loan_amnt, ylab = "Loan Amount")
plot(loans4$int_rate, ylab = "Interest Rate")
plot(loans4$annual_inc, ylab = "Annual Income")

income_outliers <- which(loans4$annual_inc > 3000000)
loans4 <- loans4[-income_outliers,]

loans5 <- loans4[sample(nrow(loans4)),]
training <- loans5[1:32529,]
testing <- loans5[32530:42529,]

glm(formula = loan_status ~ grade, family = "binomial", data = training)
glm(loan_status ~ ., family = "binomial", data = training)
test_model <- glm(loan_status ~ ., family = "binomial", data = training)
summary(test_model)
complete_model <- glm(loan_status ~ int_rate + grade + annual_inc + emp_length,
family = "binomial", data = training)

predict(complete_model, newdata = testing[1,], type = "response")
testing$loan_status[1]

model_results <- predict(complete_model, newdata = testing, type = "response")
head(model_results)

cutoff <- 1
```

```
results_1 <- ifelse(model_results > cutoff,1,0)
sum(results_1)
outcomes_1 <- table(results_1, testing$loan_status)
barplot(outcomes_1, xlab = "Loan Outcome, Fully Paid or Defaulted", col =
c(rep(c("blue","green"))), main = "Model Outcome With Cutoff = 0")
accuracy_1 <- sum(diag(outcomes_1)) / nrow(testing)
accuracy_1
accuracy_1d <- 0 / sum(testing$loan_status)
accuracy_1d

cutoff <- .15
results_15 <- ifelse(model_results > cutoff,1,0)
sum(results_15)
outcomes_15 <- table(results_15, testing$loan_status)
barplot(outcomes_15, xlab = "Loan Outcome, Fully Paid or Defaulted", col =
c(rep(c("blue","green"))), main = "Model Outcome With Cutoff = 0.15")
accuracy_15 <- sum(diag(outcomes_15)) / nrow(testing)
accuracy_15
accuracy_15d <- outcomes_15[2,2] / sum(testing$loan_status)
accuracy_15d

cutoff <- .05
results_05 <- ifelse(model_results > cutoff,1,0)
sum(results_05)
outcomes_05 <- table(results_05, testing$loan_status)
barplot(outcomes_05, xlab = "Loan Outcome, Fully Paid or Defaulted", col =
c(rep(c("blue","green"))), main = "Model Outcome With Cutoff = 0.05")
accuracy_05 <- sum(diag(outcomes_05)) / nrow(testing)
accuracy_05
accuracy_05d <- outcomes_05[2,2] / sum(testing$loan_status)
accuracy_05d

cutoff <- .00
results_0 <- ifelse(model_results > cutoff,1,0)
sum(results_0)
outcomes_0 <- table(results_0, testing$loan_status)
barplot(outcomes_0, xlab = "Loan Outcome, Fully Paid or Defaulted", col = "green",
main = "Model Outcome With Cutoff = 0")
accuracy_0 <- outcomes_0[1,2] / nrow(testing)
accuracy_0
accuracy_0d <- outcomes_0[1,2] / sum(testing$loan_status)
accuracy_0d
```

9] References

Implicit Functions. Accessed February 20, 2019. http://wmueller.com/precalculus/families/1_80.html.

"Credit Grades." Mortgage Rates and Mortgage Calculator - Mortgage101.com. Accessed February 20, 2019. https://www.mortgage101.com/article/credit-grades.

"Personal Loans Borrow up to $40,000 and Get a Low, Fixed Rate." Peer to Peer Lending & Alternative Investing. Accessed February 20, 2019. https://www.lendingclub.com/info/download-data.action.

Pritchard, Justin. "What Does It Mean When You Default?" The Balance Small Business. Accessed February 20, 2019. https://www.thebalance.com/what-happens-when-you-default-on-a-loan-315393.

Turner, Heather. "Introduction to Generalized Linear Models." Introduction to Generalized Linear Models. April 8, 22. Accessed February 14, 2019. http://statmath.wu.ac.at/courses/heather_turner/glmCourse_001.pdf.