Loan Me Money?

Charles Westby

12/10/2017

Contents

Synopsis

If a person walks into a bank needing a loan, will he or she be approved? Certain criteria determine whether or not a person will be approved for a loan. This report explores the relationship between many common factors that determine whether or not a person will be approved for a bank loan. These factors include the applicant's income, co-signer income, credit history, education level and assets. It will also see which factors have the most impact on whether a loan will be approved or not. In the end, a machine learning model will be created that will predict whether a person will be approved for a loan or not, based on given factors.

Exploratory Analysis

Loading Data and Packages

```r
library(dplyr)
library(tidyr)
library(ggplot2)
library(gridExtra)
library(caret)
library(caretEnsemble)
library(VIM)

data <- read.csv("train-file.csv")
```

Previewing The Data

Structure of Data

```r
str(data)
```

```
## 'data.frame':    614 obs. of  13 variables:
##  $ Loan_ID          : Factor w/ 614 levels "LP001002","LP001003",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender           : Factor w/ 3 levels "","Female","Male": 3 3 3 3 3 3 3 3 3 3 ...
##  $ Married          : Factor w/ 3 levels "","No","Yes": 2 3 3 3 2 3 3 3 3 3 ...
##  $ Dependents       : Factor w/ 5 levels "","0","1","2",..: 2 3 2 2 2 4 2 5 4 3 ...
##  $ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1 2 1 1 2 1 1 1 ...
##  $ Self_Employed    : Factor w/ 3 levels "","No","Yes": 2 2 3 2 2 3 2 2 2 2 ...
##  $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
##  $ CoapplicantIncome: num  0 1508 0 2358 0 ...
##  $ LoanAmount       : int  NA 128 66 120 141 267 95 158 168 349 ...
##  $ Loan_Amount_Term : int  360 360 360 360 360 360 360 360 360 360 ...
##  $ Credit_History   : int  1 1 1 1 1 1 1 0 1 1 ...
##  $ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3 3 3 3 2 3 2 ...
##  $ Loan_Status      : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 1 2 1 ...
```

First Six Records

```r
head(data)
```

```
##    Loan_ID Gender Married Dependents    Education Self_Employed
## 1 LP001002   Male      No          0     Graduate            No
## 2 LP001003   Male     Yes          1     Graduate            No
## 3 LP001005   Male     Yes          0     Graduate           Yes
## 4 LP001006   Male     Yes          0 Not Graduate            No
## 5 LP001008   Male      No          0     Graduate            No
## 6 LP001011   Male     Yes          2     Graduate           Yes
##   ApplicantIncome CoapplicantIncome LoanAmount Loan_Amount_Term
## 1            5849                 0         NA              360
## 2            4583              1508        128              360
## 3            3000                 0         66              360
## 4            2583              2358        120              360
## 5            6000                 0        141              360
## 6            5417              4196        267              360
##   Credit_History Property_Area Loan_Status
## 1              1         Urban           Y
## 2              1         Rural           N
## 3              1         Urban           Y
## 4              1         Urban           Y
## 5              1         Urban           Y
## 6              1         Urban           Y
```

Summary of Data

```r
summary(data)
```

```
##       Loan_ID       Gender     Married    Dependents        Education
```

```
##  LP001002:  1         : 13      :  3     : 15     Graduate    :480
##  LP001003:  1   Female:112   No :213   0 :345     Not Graduate:134
##  LP001005:  1   Male  :489   Yes:398   1 :102
##  LP001006:  1                          2 :101
##  LP001008:  1                          3+: 51
##  LP001011:  1
##  (Other) :608
##  Self_Employed ApplicantIncome CoapplicantIncome   LoanAmount
##     : 32       Min.   :  150   Min.   :    0     Min.   :  9.0
##  No :500       1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.0
##  Yes: 82       Median : 3812   Median : 1188     Median :128.0
##                Mean   : 5403   Mean   : 1621     Mean   :146.4
##                3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:168.0
##                Max.   :81000   Max.   :41667     Max.   :700.0
##                                                  NA's   :22
##  Loan_Amount_Term Credit_History     Property_Area Loan_Status
##  Min.   : 12      Min.   :0.0000   Rural    :179   N:192
##  1st Qu.:360      1st Qu.:1.0000   Semiurban:233   Y:422
##  Median :360      Median :1.0000   Urban    :202
##  Mean   :342      Mean   :0.8422
##  3rd Qu.:360      3rd Qu.:1.0000
##  Max.   :480      Max.   :1.0000
##  NA's   :14       NA's   :50
```

Tables

```r
table(data$Loan_Amount_Term)
```

```
##
##  12  36  60  84 120 180 240 300 360 480
##   1   2   2   4   3  44   4  13 512  15
```

```r
table(data$Credit_History)
```

```
##
##   0   1
##  89 475
```

When previewing this data, it shows that there are 614 records withs 13 different attributes. Some are factor variables and others are numeric or integers. The data also contains many missing values in its records. There are 13 missing values for Gender, 3 missing values for Married, 15 missing values for Dependents, 32 missing values for Self_Employed, 14 missing values for Loan_Amount_Term and 50 missing values for Credit_History. Of these records, 192 people were denied the loan and 422 were approved.

Upon preview it was determined that Credit_History should be converted to a factor variable. This entry has only 1's, 0's and a few missing values. The 0 represents those who have not met the necessary guidelines for Credit History and 1 represents those who have.

Manipulating Data

Creating Factor Variables

```r
#Creating Factor Variables
data$Credit_History <- factor(data$Credit_History, labels = c("No", "Yes"))
```

Subsetting Data

```r
data_sub <- data %>%
  select(-Loan_ID)
```

Imputing Missing Values

```r
#Using kNN imputation
data_sub <- kNN(data_sub)

#Removing Variables Created by Imputation
data_sub <- data_sub %>%
  select(-(Gender_imp:Loan_Status_imp))
```

New Summary

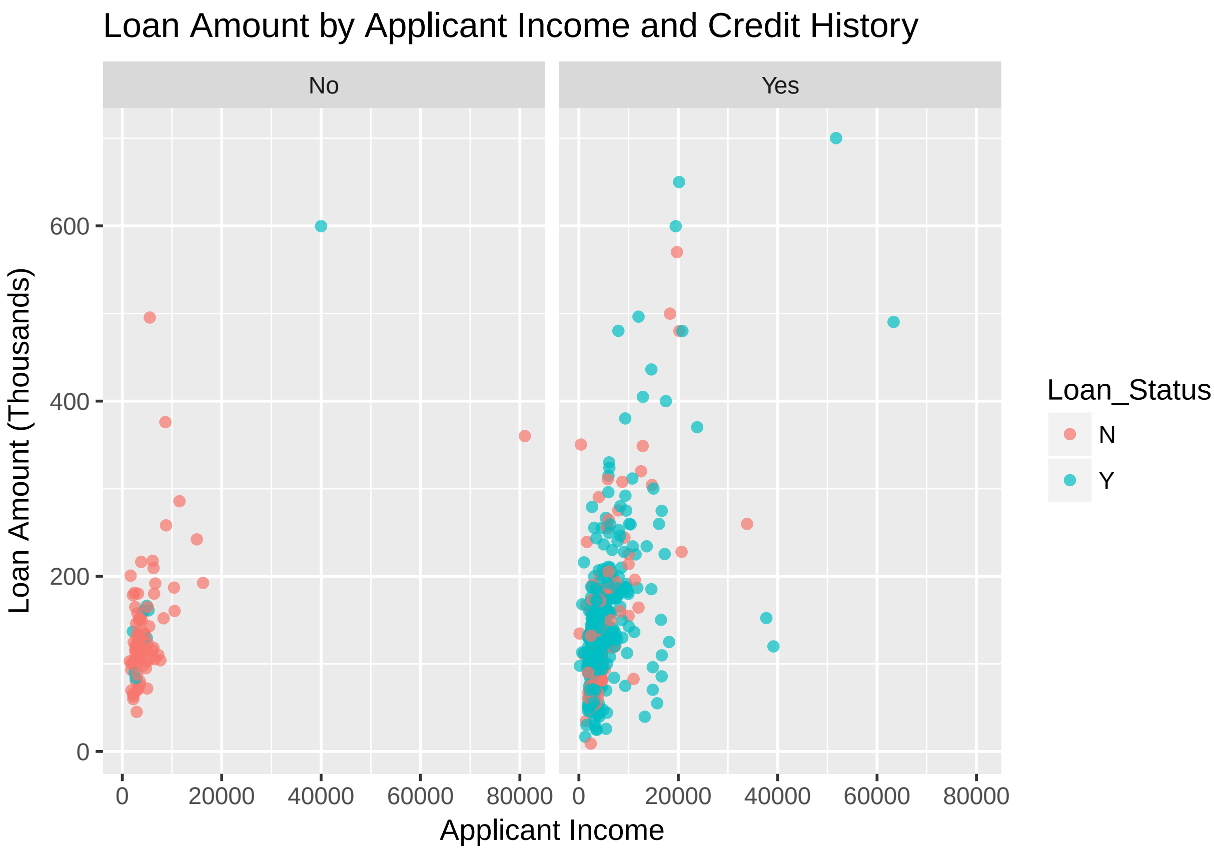
```r
summary(data_sub)
```

```
##     Gender     Married    Dependents        Education    Self_Employed
##        : 13       :  3      : 15     Graduate     :480      : 32
##  Female:112   No :213    0 :345     Not Graduate:134    No :500
##  Male  :489   Yes:398    1 :102                         Yes: 82
##                          2 :101
##                          3+: 51
##
##  ApplicantIncome CoapplicantIncome   LoanAmount    Loan_Amount_Term
##  Min.   :  150   Min.   :    0     Min.   :  9.0   Min.   : 12.0
##  1st Qu.: 2878   1st Qu.:    0     1st Qu.:100.0   1st Qu.:360.0
##  Median : 3812   Median : 1188     Median :128.0   Median :360.0
##  Mean   : 5403   Mean   : 1621     Mean   :146.0   Mean   :342.4
##  3rd Qu.: 5795   3rd Qu.: 2297     3rd Qu.:166.8   3rd Qu.:360.0
##  Max.   :81000   Max.   :41667     Max.   :700.0   Max.   :480.0
##  Credit_History   Property_Area Loan_Status
##  No : 90         Rural    :179   N:192
##  Yes:524         Semiurban:233   Y:422
##                  Urban    :202
##
##
##
```

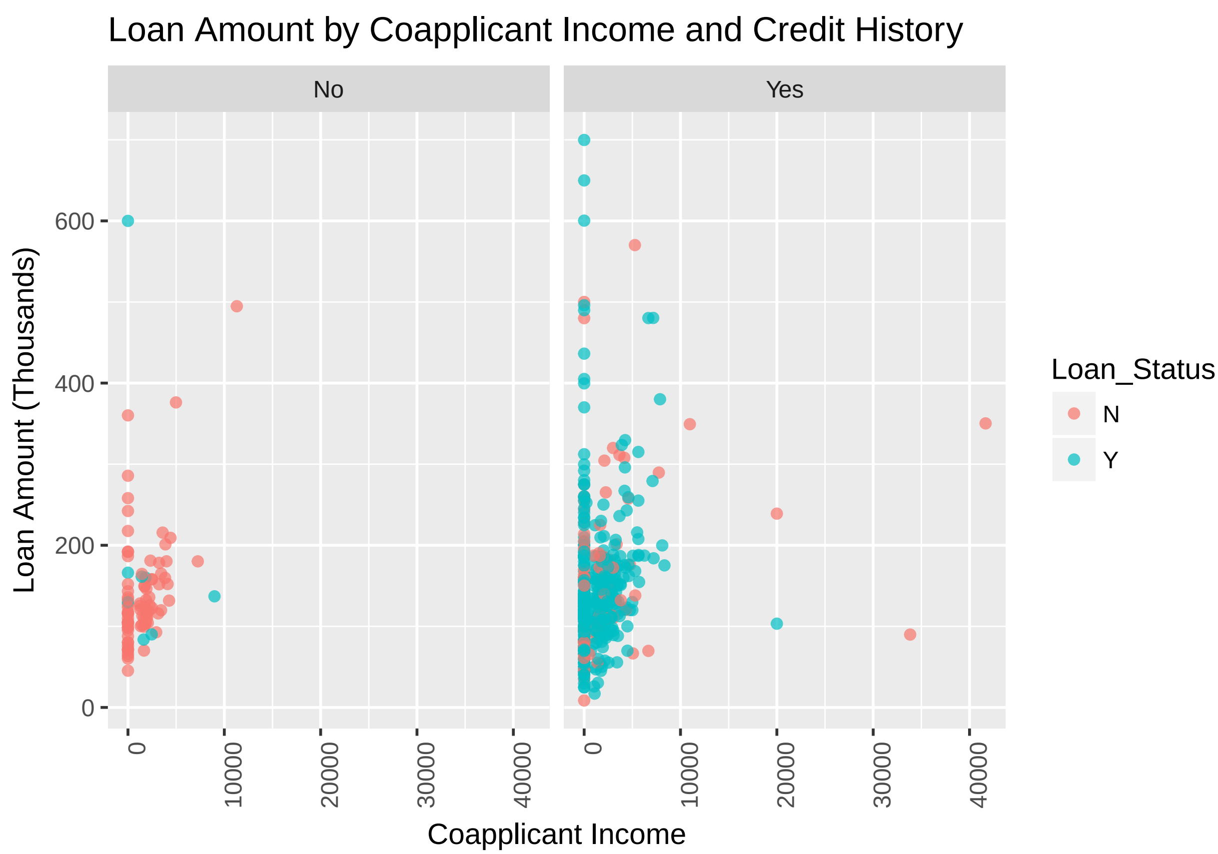
Visualizing the Data

```r
ggplot(data_sub, aes(x=ApplicantIncome, y = LoanAmount, col = Loan_Status)) +
  geom_jitter(alpha = 0.7) +
  facet_grid(. ~ Credit_History) +
```

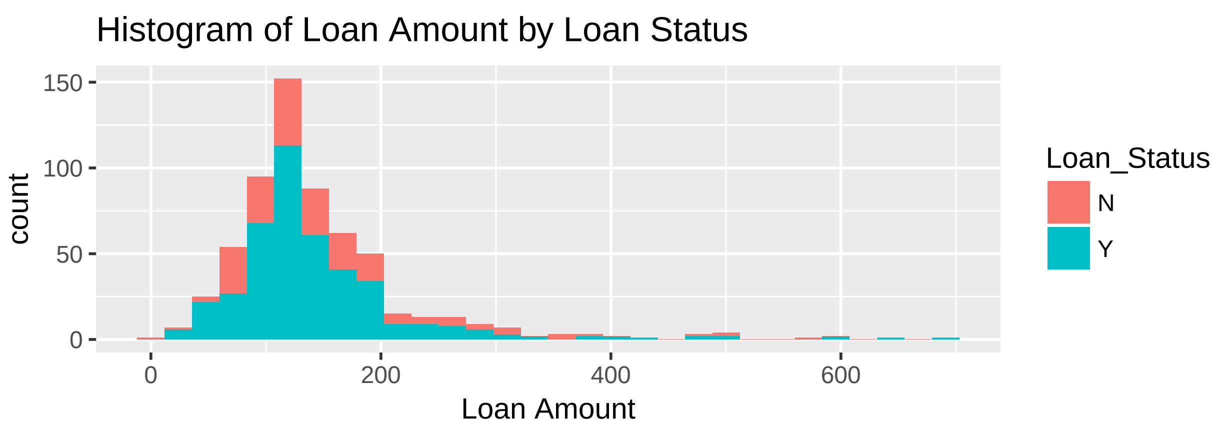
```
  labs(title = "Loan Amount by Applicant Income and Credit History",
       x = "Applicant Income", y = "Loan Amount (Thousands)")
```

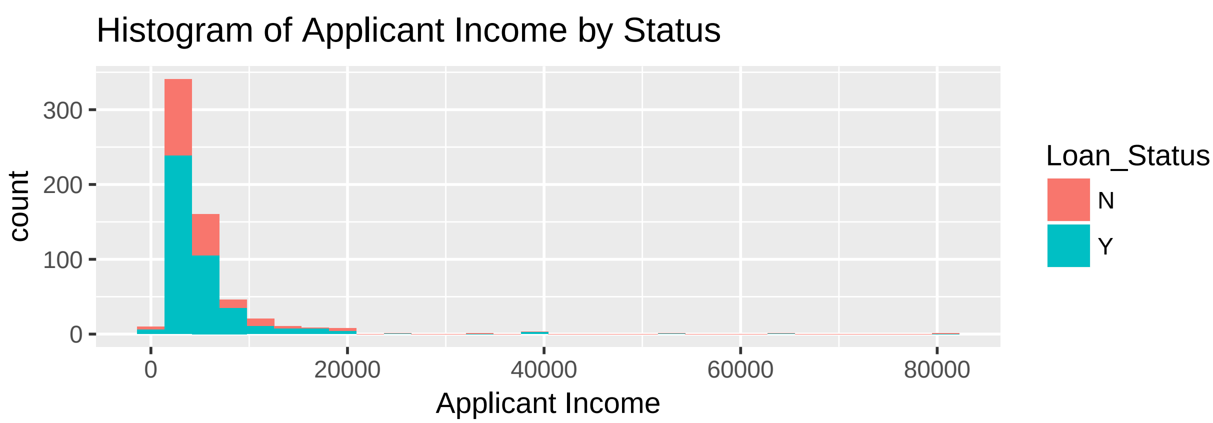



```
ggplot(data_sub, aes(x=CoapplicantIncome, y = LoanAmount, col = Loan_Status)) +
  geom_jitter(alpha = 0.7) +
  facet_grid(. ~ Credit_History) +
  labs(title = "Loan Amount by Coapplicant Income and Credit History",
       x = "Coapplicant Income", y = "Loan Amount (Thousands)") +
   theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
hist_loan <- ggplot(data_sub, aes(x=LoanAmount, fill=Loan_Status)) +
  geom_histogram() +
  labs(title="Histogram of Loan Amount by Loan Status",
       x="Loan Amount")
hist_income <- ggplot(data_sub, aes(x=ApplicantIncome, fill=Loan_Status)) +
  geom_histogram() +
  labs(title="Histogram of Applicant Income by Status",
       x = "Applicant Income")
grid.arrange(hist_loan, hist_income, nrow = 2)
```





Further Investigation

Poor Credit History But Approved

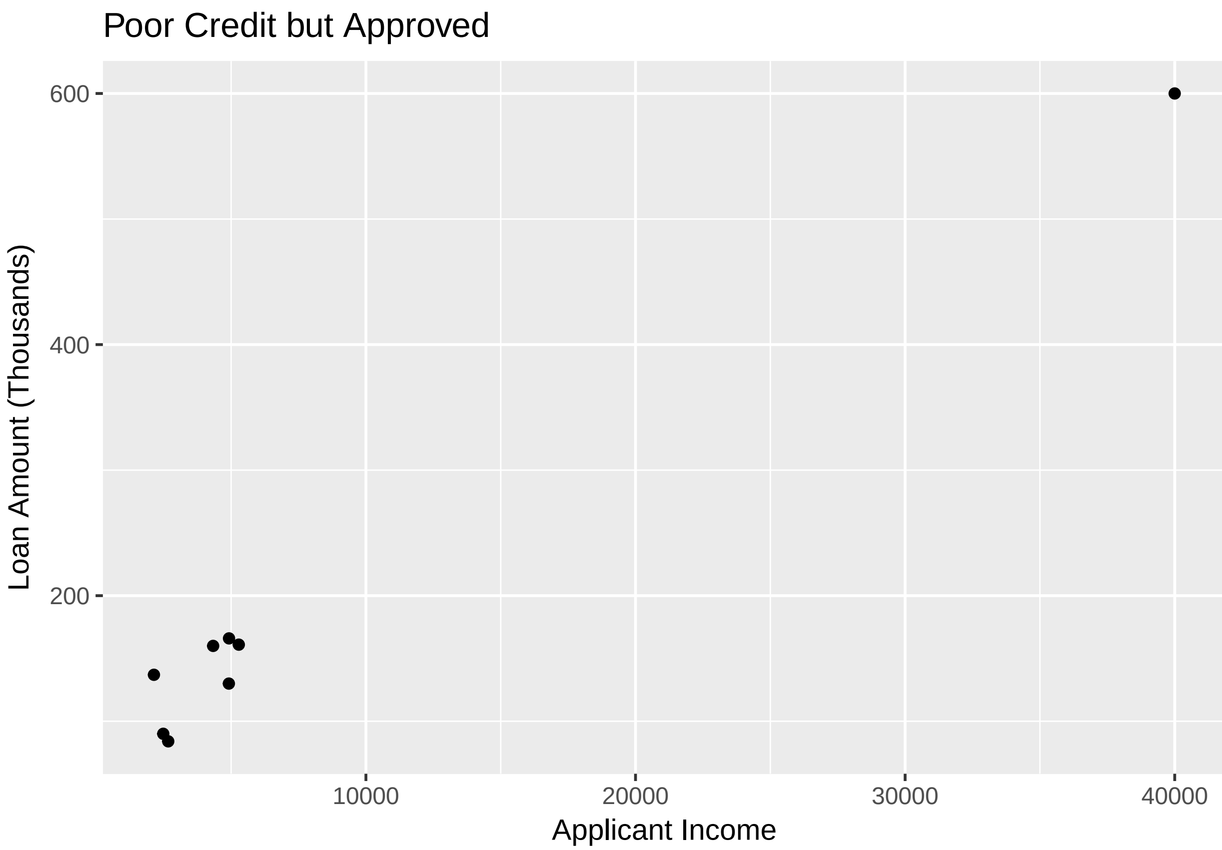
```r
#Finding Which Loans were approved and hadn't met credit history guidelines
poor_credit <- data_sub %>%
  filter(Loan_Status == "Y" & Credit_History == "No")
summary(poor_credit)
```

```
##     Gender   Married Dependents        Education Self_Employed
##        :0       :0      :0      Graduate     :7      :0
##  Female:2   No :4    0 :4       Not Graduate:1   No :8
##  Male  :6   Yes:4    1 :1                        Yes:0
##                      2 :1
##                      3+:2
##
##  ApplicantIncome CoapplicantIncome   LoanAmount     Loan_Amount_Term
##  Min.   : 2137   Min.   :    0      Min.   : 84.0   Min.   :180
##  1st Qu.: 2621   1st Qu.:    0      1st Qu.:120.0   1st Qu.:315
##  Median : 4625   Median :1528       Median :148.5   Median :360
##  Mean   : 8343   Mean   :2039       Mean   :191.0   Mean   :315
##  3rd Qu.: 5014   3rd Qu.:1975       3rd Qu.:162.2   3rd Qu.:360
##  Max.   :39999   Max.   :8980       Max.   :600.0   Max.   :360
##  Credit_History   Property_Area Loan_Status
##  No :8          Rural    :2     N:0
##  Yes:0          Semiurban:4     Y:8
##                 Urban    :2
```

```
##
##
##
```

Graphing Poor Credit

```
ggplot(poor_credit, aes(x=ApplicantIncome, y = LoanAmount)) +
  geom_point() +
  labs(title = "Poor Credit but Approved", x= "Applicant Income",
       y= "Loan Amount (Thousands)")
```



Here the Loan_Amount_Term and Credit_History variables are turned into factor variables. Also the Loan_ID variable was removed because each record has a unique value here. In addition, the missing values for the data were imputed using kNN or k-Nearest Neighbor imputation. This imputation replaces the missing data with a value similar to other comparable records.

When graphing the data, Credit History appears to be critical factor. In fact, after futher investigation, it is determined that there were only 8 out of 422 approvals for a loan, where the applicant's Credit History did not meet the guidelines. A deeper look at these 8 applicants shows that none were self-employed. Also most of these loans was less than $200,000. However one was for $600,000, but that applicant had an income of around $400,000. This analysis shows that Credit History is significant when deciding whether to approve or deny a loan.

Machine Learning Models

Partitioning The Data

```
set.seed(366284)
inTrain <- createDataPartition(y = data_sub$Loan_Status, p = 0.7, list=FALSE)
train <- data_sub[inTrain, ]
test <- data_sub[-inTrain, ]
```

Ensemble Model

Building Model List

```
control <- trainControl(method = "repeatedcv", number = 10, repeats = 3, savePredictions = TRUE, classP

algorithmList <- c('lda', 'C5.0', 'ranger', 'treebag', 'bagEarth', 'gbm', 'glmnet', 'glm')

models <- caretList(Loan_Status ~ ., train, trControl = control, methodList = algorithmList)
```

Viewing Model

```
results <- resamples(models)
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, C5.0, ranger, treebag, bagEarth, gbm, glmnet, glm
## Number of resamples: 30
##
## Accuracy
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda      0.7209302 0.7674419 0.7906977 0.7949361 0.8139535 0.8837209    0
## C5.0     0.6744186 0.7674419 0.7906977 0.7895626 0.8128461 0.8837209    0
## ranger   0.7209302 0.7674419 0.7906977 0.7957113 0.8139535 0.8837209    0
## treebag  0.6744186 0.7209302 0.7587209 0.7594340 0.7942653 0.8604651    0
## bagEarth 0.7209302 0.7674419 0.7906977 0.7957113 0.8139535 0.8837209    0
## gbm      0.7209302 0.7674419 0.7906977 0.7957113 0.8139535 0.8837209    0
## glmnet   0.7209302 0.7674419 0.7906977 0.7957113 0.8139535 0.8837209    0
## glm      0.7209302 0.7674419 0.7906977 0.7910601 0.8139535 0.8837209    0
##
## Kappa
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lda      0.2205438 0.3633099 0.4283604 0.4420722 0.4918095 0.6906475    0
## C5.0     0.1300578 0.3797874 0.4214419 0.4378585 0.4918095 0.6906475    0
## ranger   0.2205438 0.3786127 0.4283604 0.4436601 0.4918095 0.6906475    0
## treebag  0.1994681 0.2961994 0.3660963 0.3892092 0.4802604 0.6377858    0
## bagEarth 0.2205438 0.3786127 0.4283604 0.4436601 0.4918095 0.6906475    0
## gbm      0.2205438 0.3786127 0.4283604 0.4436601 0.4918095 0.6906475    0
```

```
## glmnet   0.2205438 0.3786127 0.4283604 0.4436601 0.4918095 0.6906475    0
## glm      0.2205438 0.3387758 0.4214419 0.4348255 0.4918095 0.6906475    0
```

Creating Ensemble

C5.0 Ensemble

```r
stack_C5 <- caretStack(models, method = "C5.0", trControl = trainControl(method = "repeatedcv", number =
stack_C5
```

```
## A C5.0 ensemble of 2 base models: lda, C5.0, ranger, treebag, bagEarth, gbm, glmnet, glm
##
## Ensemble results:
## C5.0
##
## 1293 samples
##    8 predictor
##    2 classes: 'N', 'Y'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 1164, 1164, 1164, 1164, 1164, 1163, ...
## Resampling results across tuning parameters:
##
##   model  winnow  trials  Accuracy   Kappa
##   rules  FALSE    1      0.7958234  0.4439821
##   rules  FALSE   10      0.7909257  0.4377383
##   rules  FALSE   20      0.7911841  0.4388200
##   rules   TRUE    1      0.7958234  0.4439821
##   rules   TRUE   10      0.7942869  0.4412192
##   rules   TRUE   20      0.7942869  0.4412192
##   tree   FALSE    1      0.7958234  0.4439821
##   tree   FALSE   10      0.7914445  0.4379784
##   tree   FALSE   20      0.7917029  0.4390601
##   tree    TRUE    1      0.7958234  0.4439821
##   tree    TRUE   10      0.7942869  0.4412192
##   tree    TRUE   20      0.7942869  0.4412192
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were trials = 1, model = rules
##  and winnow = TRUE.
```

Testing Model

```r
predictions_C5 <- predict(stack_C5, test)
confusionMatrix(predictions_C5, test$Loan_Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  27   0
##          Y  30 126
##
```

```
##                Accuracy : 0.8361
##                  95% CI : (0.7743, 0.8866)
##     No Information Rate : 0.6885
##     P-Value [Acc > NIR] : 3.956e-06
##
##                   Kappa : 0.5534
##  Mcnemar's Test P-Value : 1.192e-07
##
##             Sensitivity : 0.4737
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8077
##              Prevalence : 0.3115
##          Detection Rate : 0.1475
##    Detection Prevalence : 0.1475
##       Balanced Accuracy : 0.7368
##
##        'Positive' Class : N
##
```

GLMNET Ensemble

```r
stack_glmnet <- caretStack(models, method = "glmnet", trControl = trainControl(method = "repeatedcv", nu
stack_glmnet
```

```
## A glmnet ensemble of 2 base models: lda, C5.0, ranger, treebag, bagEarth, gbm, glmnet, glm
##
## Ensemble results:
## glmnet
##
## 1293 samples
##    8 predictor
##    2 classes: 'N', 'Y'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 1163, 1163, 1163, 1163, 1164, 1164, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        Accuracy   Kappa
##   0.10   0.0004592931  0.7952786  0.4429495
##   0.10   0.0045929313  0.7960498  0.4446590
##   0.10   0.0459293129  0.7960518  0.4443434
##   0.55   0.0004592931  0.7952806  0.4429110
##   0.55   0.0045929313  0.7955370  0.4429870
##   0.55   0.0459293129  0.7957974  0.4433167
##   1.00   0.0004592931  0.7950202  0.4420624
##   1.00   0.0045929313  0.7952786  0.4424177
##   1.00   0.0459293129  0.7957974  0.4433167
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were alpha = 0.1 and lambda
##  = 0.04592931.
```

Testing Model

```r
predictions_glmnet <- predict(stack_glmnet, test)
confusionMatrix(predictions_glmnet, test$Loan_Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  27   0
##          Y  30 126
##
##                Accuracy : 0.8361
##                  95% CI : (0.7743, 0.8866)
##     No Information Rate : 0.6885
##     P-Value [Acc > NIR] : 3.956e-06
##
##                   Kappa : 0.5534
##  Mcnemar's Test P-Value : 1.192e-07
##
##             Sensitivity : 0.4737
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8077
##              Prevalence : 0.3115
##          Detection Rate : 0.1475
##    Detection Prevalence : 0.1475
##       Balanced Accuracy : 0.7368
##
##        'Positive' Class : N
##
```

Bag Ensemble

```r
stack_bag <- caretStack(models, method = "bagEarth", trControl = trainControl(method = "repeatedcv", nu
stack_bag
```

Testing Model

```r
predictions_bag <- predict(stack_bag, test)
confusionMatrix(predictions_bag, test$Loan_Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  27   0
##          Y  30 126
##
##                Accuracy : 0.8361
##                  95% CI : (0.7743, 0.8866)
##     No Information Rate : 0.6885
```

```
##     P-Value [Acc > NIR] : 3.956e-06
##
##                   Kappa : 0.5534
##  Mcnemar's Test P-Value : 1.192e-07
##
##             Sensitivity : 0.4737
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8077
##              Prevalence : 0.3115
##          Detection Rate : 0.1475
##    Detection Prevalence : 0.1475
##       Balanced Accuracy : 0.7368
##
##        'Positive' Class : N
##
```

Conclusion

When building these models the train and the test sets were created using a 70/30 split of the original data. 70% of the data was randomly selected for the train set and the rest was selected for the test set.

Next, a list of machine learning models was created. After this list was created, a test was run that would test the accuracy of each model. Many of the models performed well. The models that performed the best were the bagEarth, random forest, glmnet and gbm models. These models each had a mean accuracy of 79.57%. Any of these models would be good for making predictions, however, an ensemble model should perform better than any other model alone.

So, the next step was to use a few different algorithms to compile these models. The first method was a C5.0 ensemble. The second method was a GLMNET ensemble. The final method was a bagEarth ensemble.

When tested the C5.0 model predicted with 83.61% accuracy. The Sensitivity or True Positive Rate was 47.37% and the Specificity or True Negative Rate was 100%. Unfortunately, these rates are backwards in the model. The model was excellent when picking a person to be approved for the loan. The model was not as good when picking when a person would be rejected. The GLMNET model performed the exact way the C5.0 model performed.

However, the bagEarth model performed differently. The bagEarth model performed with an accuracy of 84.15%. This model was slightly better at picking when a person would be rejected for the loan. However, this model's True Positive Rate only increased to 49.12%. Since the bagEarth model performed the best, it will be the model that is used for the submissions.

Submitting Results

```r
final_test  <- read.csv("test-file.csv", header = TRUE)
final_test$Credit_History <- factor(final_test$Credit_History, labels = c("No", "Yes"))
final_test <- kNN(final_test)

#Removing Variables Created by Imputation
final_test <- final_test[, 1:12]

predictions_bag <- predict(stack_bag, final_test)
```

```r
final_test$Loan_Status <- predictions_bag


dim(final_test)
```

```
## [1] 367  13
```

```r
submission_bag <- final_test[, c("Loan_ID", "Loan_Status")]
write.csv(submission_bag, "loan_rf_predictions.csv", row.names = FALSE)
```