Applied Machine Learning in R - Advanced | Lending Club

Beat Kessler, Kilian Dinkelaker

2025-07-04

Introduction

As part of the course Applied Machine Learning in R Advanced at the GSERM Summer School at the University of St. Gallen, the task was to develop the best possible machine learning model for predicting credit defaults using Lending Club data within the course week. Different teams were formed to compete in a challenge format, comparing their models daily based on their predictive accuracy.

The present work was developed in collaboration between Beat Kessler and Kilian Dinkelaker and continuously reflects the results obtained in accordance with the result structure required in the Lending Club Final Project Information Document.

Exploratory Data Analysis | EDA

To gain an initial understanding of the data, a preliminary data exploration was conducted. In this process, the provided training dataset (lending_club_train) was loaded, and the basic structure, data quality, as well as individual features and variables of the dataset were examined in more detail.

options(width = 80)

```
#library(DataExplorer)
library(ggplot2)

# Load the dataset
lc_train <- read.csv("/Users/beatkessler/Desktop/lct.csv",
                     header = TRUE, stringsAsFactors = TRUE)

# Check structure

print(lc_train[1:40, 1:5]) # show first 40 Rows and first 5 columns
```

```
##    id default loan_amnt       term                          emp_title
## 1   1       0     16000  36 months                Toledo Molding And Die
## 2   2       0     13200  36 months
## 3   3       0     18350  60 months                   District Supervisor
## 4   4       0      6025  36 months
## 5   5       1     35000  60 months                       ELAR Consultant
## 6   6       0     16900  36 months                       Agency Producer
## 7   7       0     10000  36 months           Material Handler Supervisor
## 8   8       0      2500  36 months                                  Sales
## 9   9       0     21000  60 months                         Ernst & Young
## 10 10       1     14975  36 months                     Registered Nurse
```

```
## 11 11       0     12700  60 months                         Office Manager
## 12 12       1     14425  60 months                                    mso
## 13 13       0     26000  36 months  Application Systems Analyst Engineer
## 14 14       0     12000  60 months                      Help Desk Analyst
## 15 15       0     25000  60 months                             Accountant
## 16 16       0     15000  36 months                              inspector
## 17 17       0     20000  36 months                  Bristol Myers Squibb
## 18 18       0      5000  36 months
## 19 19       0     28000  36 months                              Sys Admin
## 20 20       0      8800  36 months                                  Agent
## 21 21       0     16000  36 months
## 22 22       0     15000  60 months                               Director
## 23 23       0      8000  36 months                                 Driver
## 24 24       0     15000  36 months      Senior Program Design Specialist
## 25 25       1     25000  60 months             Central Office Technician
## 26 26       0     12000  36 months                          wellpoint inc
## 27 27       0     11500  36 months                               Shipping
## 28 28       0     31050  60 months Finance Systems and Reporting Manager
## 29 29       0      8000  36 months                  Correctional officer
## 30 30       0     15000  60 months                         HVAC DESIGNER
## 31 31       0     12000  36 months                               manager
## 32 32       0     11000  36 months                           The Hartford
## 33 33       0     16000  60 months          Staff Administrative Analyst
## 34 34       0      5000  36 months                           phlebotomist
## 35 35       0     15000  36 months                                    FMC
## 36 36       0     24000  36 months                      Sales Executive
## 37 37       0     21000  36 months                        Avionics tech
## 38 38       0      7000  36 months                                 USPS
## 39 39       1     10625  36 months
## 40 40       0      3000  36 months                     assistant manager
```

```r
# Ensure default is a factor (0 = no, 1 = yes)
lc_train$default <- factor(lc_train$default, levels = c(0, 1), labels = c("no", "yes"))

# Check distribution
table(lc_train$default[0:100]) # Check Distribution for 100 Rows
```

```
##
##  no yes
##  83  17
```

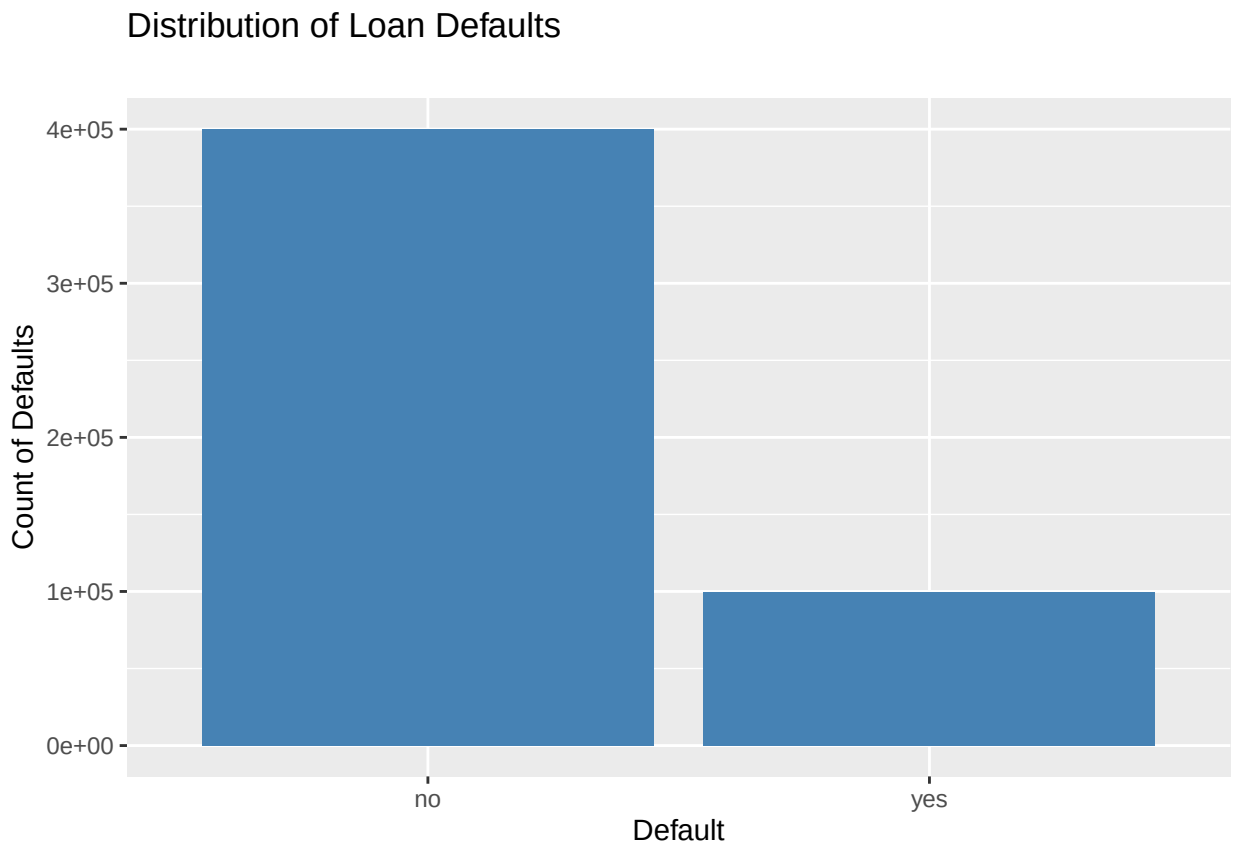
To gain an initial understanding of the data, a preliminary data exploration was conducted. As shown in the code, the dataset was first loaded, and the structure of the dataset was analyzed using the str() function to get a rough overview of the available data. It was revealed that the dataset contains a total of 500,000 observations and 103 variables. The different data types of the individual variables were also identified. The dataset includes a mixture of int (integers), num (numeric values with decimals), factor (categorical variables with levels), and logical data types, which represent TRUE or FALSE values.

In the next step, special attention was given to the target variable default. Using the ggplot library, a closer look was taken at this variable.

```
# Bar Chart of Default Distribution

print(lc_train$default[0:10])
```

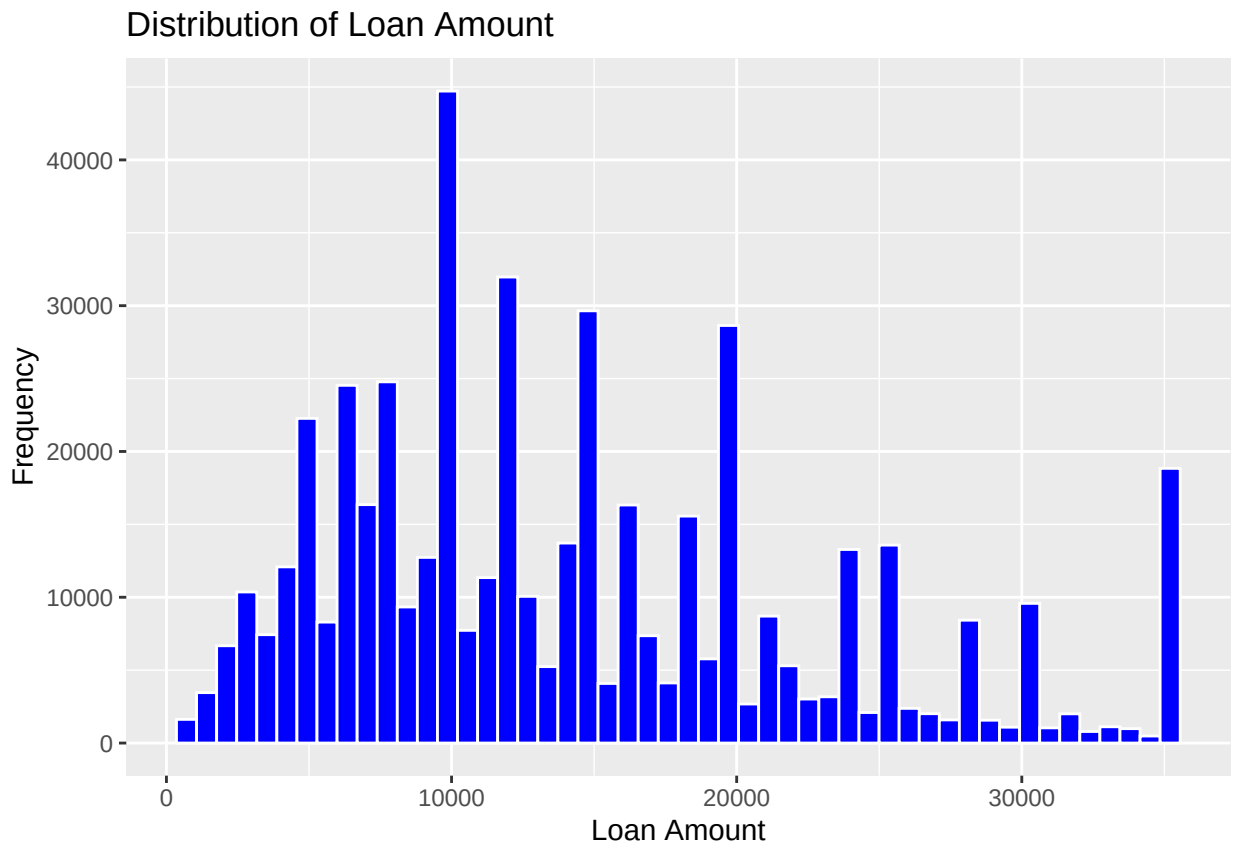
```
##  [1] no  no  no  no  yes no  no  no  no  yes
## Levels: no yes
```

```
print(
  ggplot(lc_train, aes(x = default)) +
    geom_bar(fill = "steelblue") +
    labs(x = "Default", y = "Count of Defaults",
         title = "Distribution of Loan Defaults") +
    theme(plot.title = element_text(margin = margin(b = 20)))
)
```



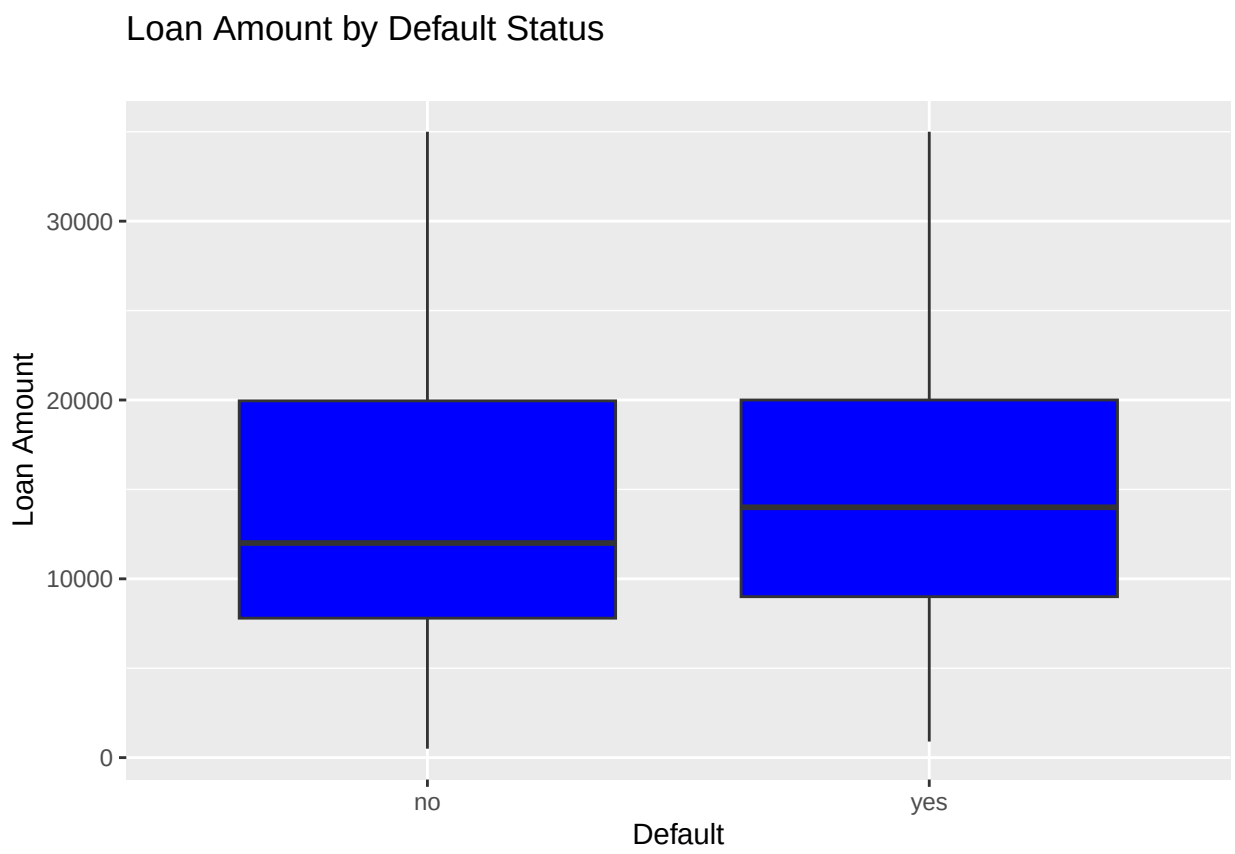
At first, a bar chart was created to show the distribution of defaults between individuals who repaid their loans and those who did not. As visible in the bar chart, the majority of observations fall under individuals without loan default (shown in the plot as default = 0), while a significantly smaller portion represents individuals who defaulted on their loans (default = 1). Visually estimated from the plot, there appear to be up to four times more cases without default compared to cases with default. While this result was not surprising to us, it was still helpful to develop a clearer understanding of the distribution.

```r
# Histogram of Loan Amount

ggplot(lc_train, aes(x = loan_amnt)) +
  geom_histogram(bins = 50, fill = "blue", color = "white") +
  labs(title = "Distribution of Loan Amount", x = "Loan Amount", y = "Frequency")
```



In the second chart, we chose a histogram to visualize the distribution of loan amounts. The X-axis represents the loan amount, and the Y-axis shows the frequency of these loans. As shown in the plot, a loan amount of around 10,000 appears most frequently in the dataset. However, it is difficult to derive a clear trend from the plot. For instance, within the range of 10,000 to 20,000, a relatively broad range on the X-axis, several higher frequencies are visible. There is also a noticeable peak at the upper end of the spectrum around 40,000.

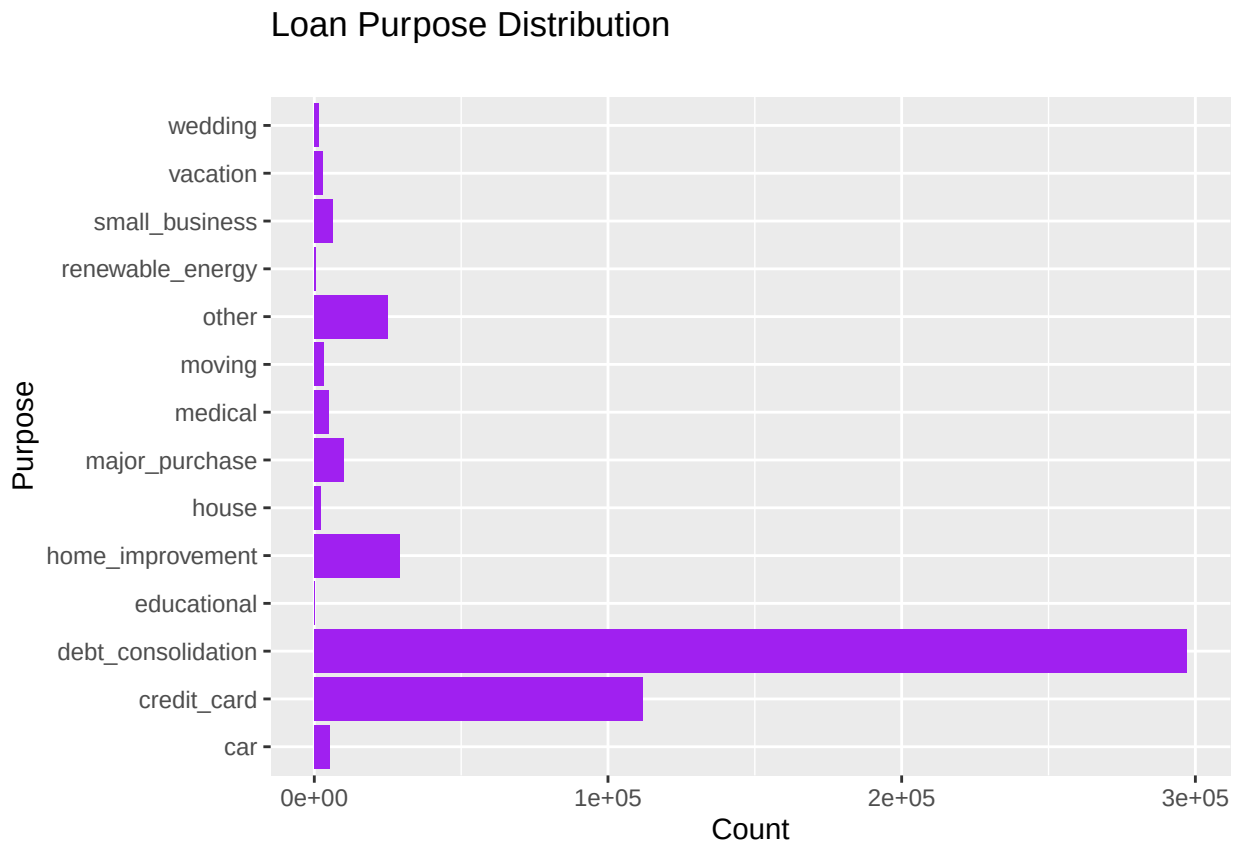
```r
# Boxplot of Loan Amount by Default

ggplot(lc_train, aes(x = default, y = loan_amnt)) +
  geom_boxplot(fill = "blue") +
  labs(title = "Loan Amount by Default Status", x = "Default", y = "Loan Amount") +
  theme(plot.title = element_text(margin = margin(b = 20)))
```



Another chart we used to gain better insight was a boxplot comparing the loan amounts between repayers and non-repayers. As visible in the boxplot, the differences are relatively small. For default = 0, meaning individuals who repaid their loans, the quartiles are closer together, and the median is slightly higher compared to default = 1. However, the interquartile ranges are for both groups relatively similar.

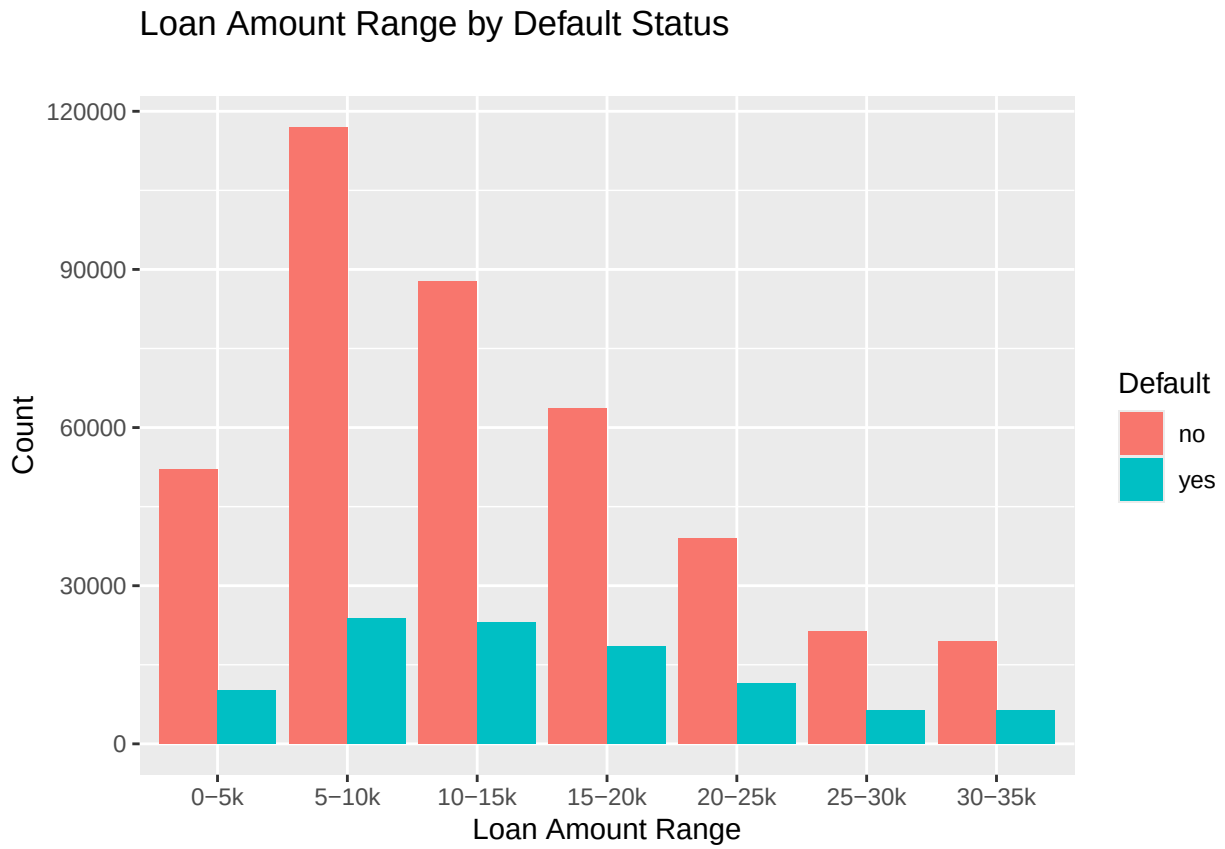
Overall, while there is a difference in medians for boith groups, the overlap in distribution implies that just the loan amount alone may be not a good indicator for default risk.

```r
# Loan Purpose Distribution Bar Chart

ggplot(lc_train, aes(x = purpose)) +
  geom_bar(fill = "purple") +
  coord_flip() +
  labs(title = "Loan Purpose Distribution", x = "Purpose", y = "Count") +
  theme(plot.title = element_text(margin = margin(b = 20)))
```



Another interesting aspect we considered, beyond the numerical distributions, was the question of why people take out loans in the first place. The results showed a relatively clear picture. Most loans were taken to consolidate existing debt or to pay off credit card balances. Other reasons, such as buying a car, going on vacation, or getting married, were significantly less common.

```r
# Grouped Bar Chart: Loan Amount Bins by Default

lc_train$loan_range <- cut(lc_train$loan_amnt,
                           breaks = seq(0, 35000, by = 5000),
                           labels = c("0-5k", "5-10k", "10-15k", "15-20k",
                                      "20-25k", "25-30k", "30-35k"),
                           include.lowest = TRUE)

ggplot(lc_train, aes(x = loan_range, fill = default)) +
  geom_bar(position = "dodge") +
  labs(title = "Loan Amount Range by Default Status",
       x = "Loan Amount Range", y = "Count", fill = "Default") +
  theme(plot.title = element_text(margin = margin(b = 20)))
```



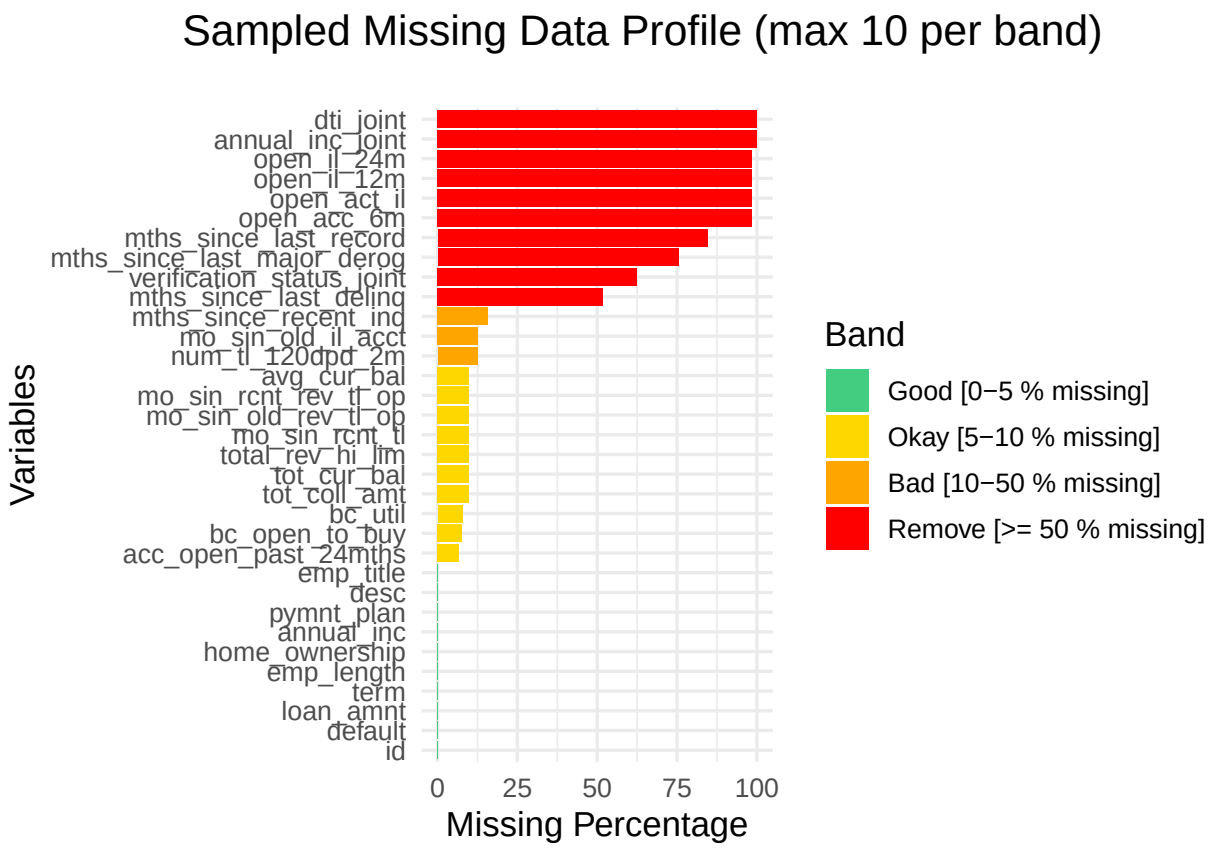
After discussing the initial charts within the team, we focused on the question of whether the most common loan ranges differ significantly between people who repay their loans and those who do not. Our assumption was that, especially with higher loan amounts, proportionally fewer people repay their loans, and that the ratio would trend toward 50 percent. To investigate this, we decided to create a groupped bar chart showing default = 0 and default = 1 side by side, allowing for a comparison of how the ratio changes. Since loan_amnt is numeric and therefore continuous—resulting in many unique values in the dataset, we decided to group the loan amounts into ranges for each bar to better identify trends. To create this type of chart, we had to convert the target variable default into a factor to enable a clear color distinction between the two classes and to display the bars side by side. As seen in the plot and code, we defined ranges in steps of 5,000, starting from 0–5k up to 30–35k. Contrary to our hypothesis that the ratio between those who repay and those who do not would become more balanced at higher loan amounts, this assumption was not confirmed.

At the end of our initial exploratory analysis, we generated an automated report using the DataExplorer package, which provided us with further insights into the dataset in a compact report format. In addition to some basic statistics we had already identified—such as the number of observations and variables and the count of discrete and continuous columns—the report also provided valuable information about missing values (i.e., NAs) per variable. Of particular interest to us was the graphical overview of how much data is missing for each column. The variables were categorized based on their completeness. As a result, we could roughly classify the variables into three main categories based on how complete they are. This insight—knowing which variables are mostly complete and which are not—is essential for building machine learning models that will use these variables as input. Additional analyses in the automatic created report included univariate distributions, histograms of various variables, bar charts, and QQ plots.

To ensure that the automatically generated plots in the report are correct, we decided to create an additional plot listing variables according to their percentage of missing values. Since this serves purely as a validation check, only 10 variables per defined category were plotted. The importance of these results is particularly high because the selection of complete variables forms the basis for our initial models. The results of this variable completeness check are shown in the plot below and are consistent with the report.

```r
library(dplyr)
library(tidyr)
library(ggplot2)

# Missing Percentage per Column
missing_df <- lc_train %>%
  summarise(across(everything(), ~ mean(is.na(.)) * 100)) %>%
  pivot_longer(everything(), names_to = "Variable", values_to = "MissingPercent")

# Categorise into bands with labels
missing_df <- missing_df %>%
  mutate(Band = case_when(
    MissingPercent <= 5 ~ "Good [0-5 % missing]",
    MissingPercent > 5 & MissingPercent <= 10 ~ "Okay [5-10 % missing]",
    MissingPercent > 10 & MissingPercent <= 50 ~ "Bad [10-50 % missing]",
    MissingPercent > 50 ~ "Remove [>= 50 % missing]"
  ))

# Band Factor
missing_df$Band <- factor(missing_df$Band,
                          levels = c("Good [0-5 % missing]",
                                     "Okay [5-10 % missing]",
                                     "Bad [10-50 % missing]",
                                     "Remove [>= 50 % missing]"))

# Sample 10 per Category
set.seed(123) # reproducible sampling
missing_sample <- missing_df %>%
  group_by(Band) %>%
  slice_head(n = 10) %>%
  ungroup()

# Order Variable Categories
missing_sample <- missing_sample %>%
  arrange(Band, MissingPercent) %>%
  mutate(Variable = factor(Variable, levels = Variable))
```

```r
# Plot
ggplot(missing_sample, aes(x = Variable, y = MissingPercent, fill = Band)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Sampled Missing Data Profile (max 10 per band)",
       x = "Variables",
       y = "Missing Percentage") +
  scale_fill_manual(values = c("Good [0-5 % missing]" = "seagreen3",
                               "Okay [5-10 % missing]" = "gold",
                               "Bad [10-50 % missing]" = "orange",
                               "Remove [>= 50 % missing]" = "red")) +
  theme_minimal(base_size = 13) +
  theme(plot.title = element_text(hjust = 0, margin = margin(b = 20, l = -90)),
)
```



In summary, the exploratory data analysis of the Lending Club dataset provided us with essential initial insights. As expected, initial investigations showed that the majority of borrowers repay their loans properly, while only a small proportion do not. Simple visualizations such as bar charts and boxplots helped us better understand early hypotheses about the relationship between loan amount and repayment behavior. The evaluation of loan purposes also revealed typical reasons why individuals apply for loans. With the help of the DataExplorer library, we were able to generate an automated report that offered an overview of missing values within the dataset. The findings from this exploratory data analysis provide an important foundation for the development of the first machine learning models, which are described in the following chapters.

First Logistic Regression Model

After internal discussions and the initial course sessions, we decided to begin by creating a logistic regression model. We chose this model type in order to develop a first score as efficiently and quickly as possible.

```
## Set Up Model ##
# Load the Train Dataset
lc <- read.csv("/Users/beatkessler/Desktop/lct.csv")

# Check Completeness
print(sum(is.na(lc$delinq_amnt))) # Output 0
```

```
## [1] 0
```

```
print(sum(is.na(lc$acc_now_delinq))) # Output 0
```

```
## [1] 0
```

```
print(sum(is.na(lc$initial_list_status))) # Output 0
```

```
## [1] 0
```

```
print(sum(is.na(lc$revol_bal))) # Output 0
```

```
## [1] 0
```

```
print(sum(is.na(lc$pub_rec))) # Output 0
```

```
## [1] 0
```

```
print(sum(is.na(lc$mort_acc))) # e.g many values are missing here
```

```
## [1] 33998
```

```
# Set seed for reproducibility
set.seed(300)

# Split Training and Testing Data
data_split <- initial_split(lc, prop = 0.9, strata = default)

sub_lc_train <- training(data_split)
sub_lc_test <- testing(data_split)

# Fit Logistic Regression Model
logit_lc <- glm(default ~ delinq_amnt + acc_now_delinq + initial_list_status + revol_bal + pub_rec,  #
                data = sub_lc_train,
                family = binomial(link = "logit"))

# Model Summary
summary(logit_lc)
```

```
##
## Call:
## glm(formula = default ~ delinq_amnt + acc_now_delinq + initial_list_status +
##     revol_bal + pub_rec, family = binomial(link = "logit"), data = sub_lc_train)
##
## Coefficients:
##                         Estimate Std. Error  z value Pr(>|z|)
## (Intercept)          -1.426e+00  6.158e-03 -231.582  < 2e-16 ***
## delinq_amnt           3.984e-06  5.401e-06    0.738  0.46081
## acc_now_delinq        1.482e-01  4.811e-02    3.080  0.00207 **
## initial_list_statusw  1.213e-01  7.512e-03   16.148  < 2e-16 ***
## revol_bal            -2.319e-06  2.093e-07  -11.079  < 2e-16 ***
## pub_rec               8.730e-02  6.415e-03   13.607  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 449481  on 449998  degrees of freedom
## Residual deviance: 448846  on 449993  degrees of freedom
## AIC: 448858
##
## Number of Fisher Scoring iterations: 4
```

As shown in the code, the dataset was first loaded using a standard approach. For the logistic regression model, we used the appropriate R syntax. A generalized linear model (GLM) was chosen for the modeling. We also selected the logistic link function "logit". This function transforms the estimated probabilities into a linear relationship and ensures that the model outputs, as desired in our case, remain within the interval [0,1].

To begin, we deliberately chose a small set of five features. Based on our EDA analysis in the previous chapter, we selected six variables that showed a high degree of completeness and, based on their descriptions, seemed likely to have a significant influence on the target variable. We additionally verified the completeness of the selected variables using a simple print() function to confirm that the findings from the report were correct. The result of 0 NAs confirmed our assumption.

Accordingly, we selected the variables delinq_amnt, acc_now_delinq, initial_list_status, revol_bal, and pub_rec. The target variable used was default, since it directly represents the outcome of interest in our analysis.

Finally, we used the summary() function to check whether the variables we had selected based on intuition actually had a significant influence on the target variable. According to the output, the variables acc_now_delinq, initial_list_status, revol_bal, and pub_rec had particularly low p-values—below 5 percent and are therefore statistically significant. Only the variable delinq_amnt appears to have little influence on the target variable default.

```r
## Submit predicted Values ##

# Load the Lending Club test dataset
logit_lc_test <- read.csv("/Users/beatkessler/Desktop/lct.csv")

# Use the trained logistic regression model
logit_lc_test$default_prob <- predict(logit_lc, logit_lc_test, type = "response")

summary(logit_lc_test$default_prob)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0003113 0.1893645 0.1947681 0.1993051 0.2088348 0.9976790
```

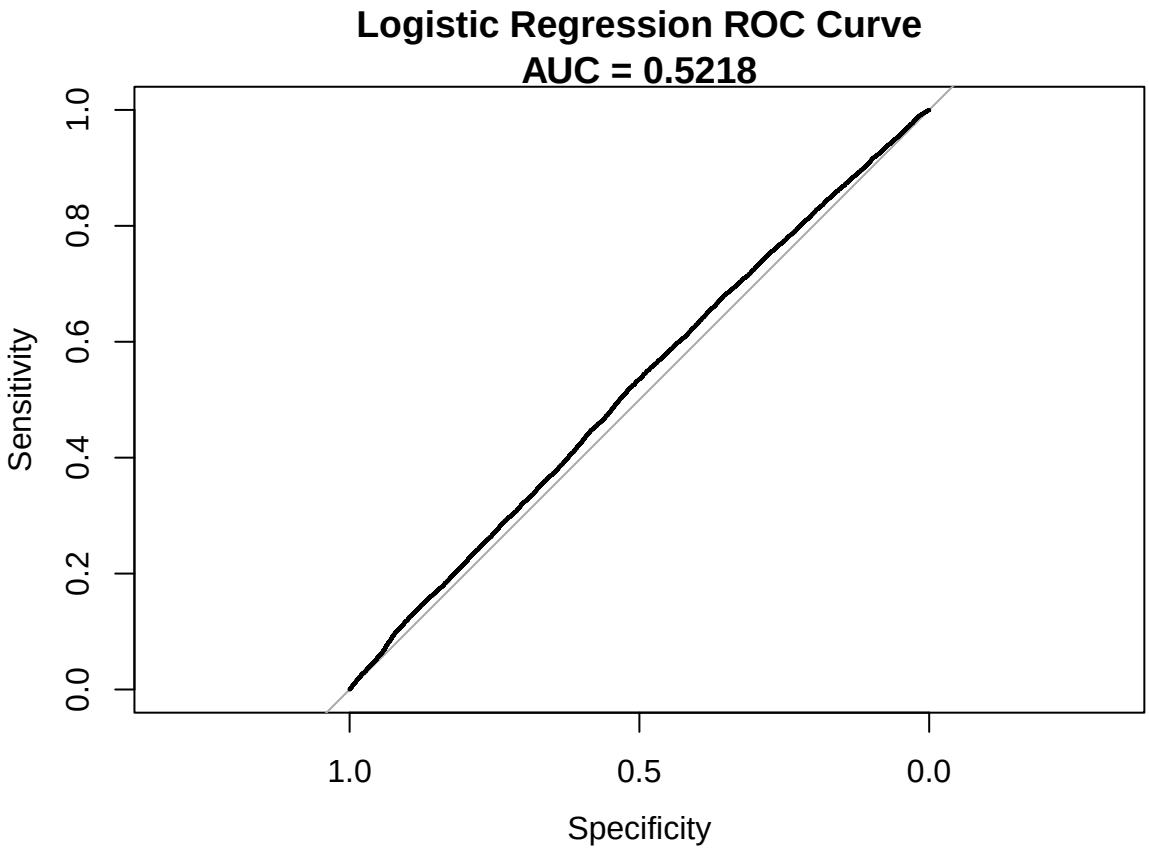
```r
# Save predicted default probabilities to a CSV for submission
write.csv(logit_lc_test$default_prob, "logit Modell 1_Kessler_Dinkelaker.csv")

## AUC Calculation ##

library(pROC)

# Predict probabilities on test data
sub_lc_pred <- predict(logit_lc, newdata = sub_lc_test, type = "response")

# Calculate ROC and AUC
roc_logit <- roc(response = sub_lc_test$default, predictor = sub_lc_pred)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
# Print AUC
auc_logit <- auc(roc_logit)
print(paste("AUC =", round(auc_logit, 4)))
```

```
## [1] "AUC = 0.5218"
```

```r
# Plot ROC Curve
plot(roc_logit, main = paste("Logistic Regression ROC Curve\nAUC =", round(auc_logit, 4)))
```



In order to independently evaluate the performance and accuracy of our models, we calculated the AUC (Area Under the Curve) and the ROC (Receiver Operating Characteristic) curve. Using the predict() function, we predicted the probabilities for the test data, and then created the ROC curve and calculated the AUC value with the package called pROC.

For our first model, as shown in the plot, we achieved an AUC value of only 0.5218. This means that the model performs barely better than pure chance. Pure chance would actually result in a value of 0.5. The ROC curve also illustrates this clearly, as it lies close to the diagonal, which represents a purely random classifier.

Data Preperation and Feature Engineering | Improvement Logistic Regression Model

After receiving the initial results, our team discussed the first steps to improve the performance of our model. One key factor in this process was data preparation and feature engineering.

As a first measure to enhance performance, we decided to reuse the previously implemented logistic regression model. However, this time we added additional complete variables from the dataset, as well as several incomplete ones. The newly selected variables had to be processed accordingly as part of data preparation, with missing values imputed using the respective column means.

In total, we decided to include in total five complete variables, same Variables like before, and six new incomplete variables. As in our earlier approach, the selection of these variables was largely based on intuition and subjective judgment.

Additionally, we chose to create a Missing Value Indicator (MVI) for each of the incomplete variables. This allowed us to retain information about which values were originally missing, which we expected to further improve the overall model performance.

```
## Check Completeness of Variables ##

# Consistent Variables Examples
print(sum(is.na(lc$zip_code)))
```

```
## [1] 0
```

```
print(sum(is.na(lc$purpose)))
```

```
## [1] 0
```

```
# print(sum(is.na(lc$desc))) # Output 0
# print(sum(is.na(lc$pymnt_plan))) # Output 0
# print(sum(is.na(lc$loan_amnt))) # Output 0
# print(sum(is.na(lc$fico_range_low))) # Output 0
# print(sum(is.na(lc$fico_range_high))) # Output 0
# print(sum(is.na(lc$home_ownership))) # Output 0

# Inconsistent Variables Examples
print(sum(is.na(lc$mort_acc))) # Output: 33998
```

```
## [1] 33998
```

```
print(sum(is.na(lc$acc_open_past_24mths))) # Output: 33998
```

```
## [1] 33998
```

```
print(sum(is.na(lc$num_rev_accts))) # Output: 48591
```

```
## [1] 48591
```

Mean Imputation

At the beginning of our data preparation, we first addressed mean imputation. In this process, we applied mean imputation, where missing values are replaced by the mean of the respective column.

To do this, we used the mean() function with the argument na.rm = TRUE to ensure that missing values (NA) are excluded from the mean calculation. This guarantees that the computed average is based only on available values.

To verify that the imputation was successful, we used a simple print statement to inspect the variable mort_acc before and after the imputation. This allowed us to confirm that all NA values had been replaced.

As shown in the output, the missing values were successfully filled with the respective mean values.

```
lc <- read.csv("/Users/beatkessler/Desktop/lct.csv")
lc_imputed <- lc

# 2. Create MVI columns BEFORE splitting
library(dplyr)

lc_imputed <- lc_imputed %>%
  mutate(
    mort_acc_MVI = if_else(is.na(mort_acc), 1, 0),
    acc_open_past_24mths_MVI = if_else(is.na(acc_open_past_24mths), 1, 0),
    num_sats_MVI = if_else(is.na(num_sats), 1, 0),
    num_rev_accts_MVI = if_else(is.na(num_rev_accts), 1, 0),
    total_bc_limit_MVI = if_else(is.na(total_bc_limit), 1, 0),
    bc_util_MVI = if_else(is.na(bc_util), 1, 0)
  )
```

Missing Value Indicator [MVI]

After completing mean imputation, we then, as shown in the previous code, focused on creating Missing Value Indicators (MVI) for the relevant variables. For each variable containing missing values, we created a new column that systematically captures whether a value is missing.

In these new columns, each row is assigned either a 1 if the original value was missing, or a 0 if a valid value was present. As mentioned earlier in the introduction, the main goal of MVI is to reduce information loss. When performing imputation alone, we risk losing information about the fact that a value was originally missing. MVIs help preserve this information by explicitly indicating where values were absent in the original dataset.

This, in turn, allows the machine learning model to learn from patterns of missingness, which can lead to better predictive performance.

To implement the MVI columns, we used the mutate() function, which allows us to add new columns or overwrite existing ones. Within mutate(), we used the if_else() function to check whether a value is NA, and based on that, we inserted either a 0 or a 1 into the new indicator column.

After completing our initial data preparation, we were able to build an improved logistic model, as shown in the code snippet below.

Improvement Logistic Regression Model with Missing Value Indicators [MVI]

```r
#Splitting
library(rsample)
data_split <- initial_split(lc_imputed, prop = 0.9, strata = "default")
sub_lc_train2 <- training(data_split)
sub_lc_test2 <- testing(data_split)

# Improved Logistic Regression Model
logit_lc_2 <- glm(default ~
                    delinq_amnt + acc_now_delinq + initial_list_status +
                    revol_bal + pub_rec +
                    mort_acc + mort_acc_MVI +
                    acc_open_past_24mths + acc_open_past_24mths_MVI +
                    num_sats + num_sats_MVI +
                    num_rev_accts + num_rev_accts_MVI +
                    total_bc_limit + total_bc_limit_MVI +
                    bc_util + bc_util_MVI,                    # In Total 17 Variables (12 new ones)
                  data = sub_lc_train2,
                  family = binomial(link = "logit"))
```

```r
## AUC Calculation ##

# Predict probabilities on test data
sub_lc_pred_2 <- predict(logit_lc_2, newdata = sub_lc_test2, type = "response")

# Calculate ROC and AUC
roc_logit_2 <- roc(response = sub_lc_test2$default, predictor = sub_lc_pred_2)
```

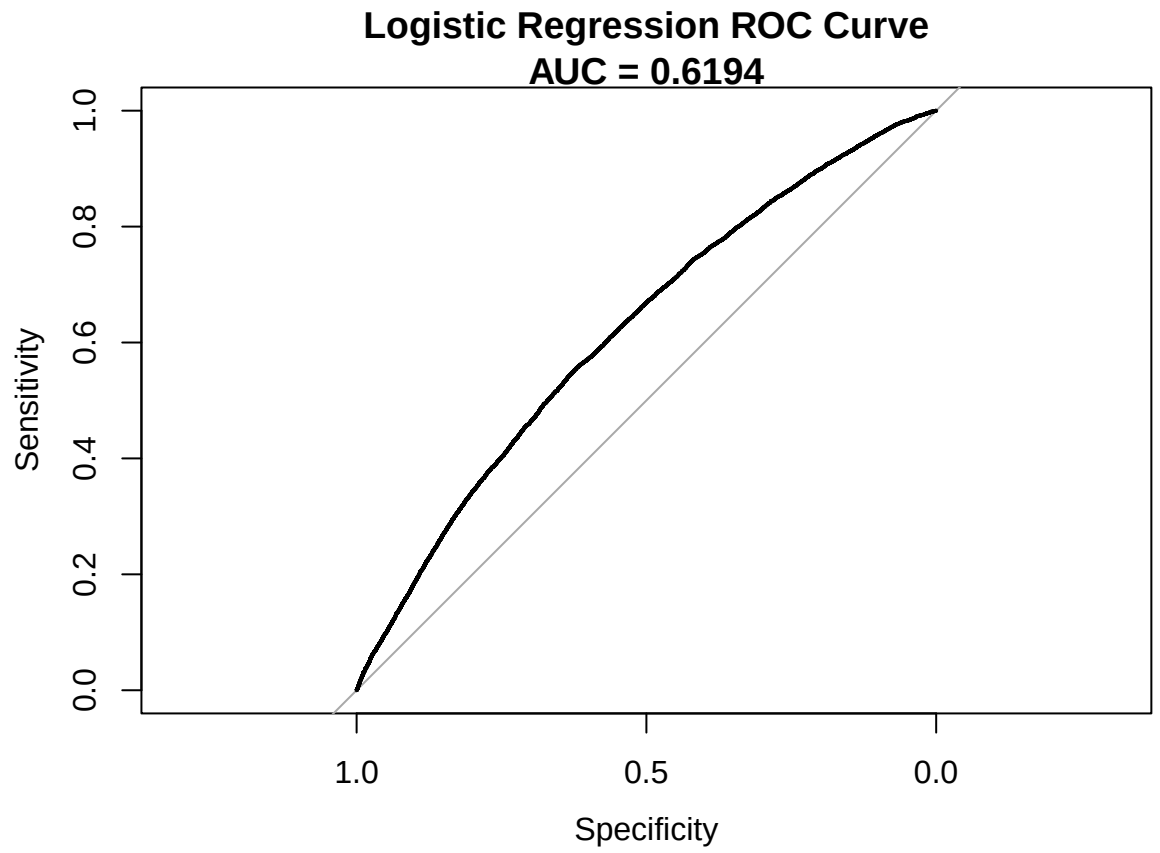
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# Print AUC
auc_logit <- auc(roc_logit_2)
print(paste("AUC =", round(auc_logit, 4)))
```

```
## [1] "AUC = 0.6194"
```

```r
# Plot ROC Curve
plot(roc_logit_2, main = paste("Logistic Regression ROC Curve\nAUC =", round(auc_logit, 4)))
```



```
### Submission for Brett Lantz ###

logit_lc_test <- read.csv("/Users/beatkessler/Desktop/lending_club_test.csv")

logit_lc_test$default_prob <- predict(logit_lc, logit_lc_test, type = "response")

summary(logit_lc_test$default_prob)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.004167 0.189392 0.195827 0.199353 0.208891 0.515028
```

```
write.csv(logit_lc_test$default_prob, "logit Modell 2.csv")
```

After performing mean imputation and adding missing value indicators in the previously described chapter on data preparation, we were able to build a new logistic model that extends the previously used model with new variables. In total, the model now contains 17 variables, of which 6 are newly added Variables. The variables with missing value indicators were included as separate variables, resulting in a total of 12 newly added variables. The commands and model modifications are exactly the same as in the previous model; only the number of variables has increased.

Finally, we calculated the AUC score again and achieved an improved value of 0.6148. This confirms our assumption that mean imputation and missing value indicators lead to significantly better performance. However, although the newly achieved performance is considerably higher than the previous value of 0.5218, it is still relatively low, leaving much room for improvement.

Improvement Logistic Regression Model with MVI + Mean Imputation

```
## More Features ##

lc <- read.csv("/Users/beatkessler/Desktop/lct.csv")

lc_imputed_2 <- lc_imputed #copy of the existing list with new added features

## Mean Imputation

columns_to_impute <- c(
  "pub_rec_bankruptcies", "total_bal_ex_mort", "mths_since_recent_bc",
  "bc_open_to_buy", "percent_bc_gt_75", "num_bc_sats",
  "total_il_high_credit_limit", "tot_hi_cred_lim", "num_tl_op_past_12m",
  "num_tl_90g_dpd_24m", "num_tl_30dpd", "num_rev_tl_bal_gt_0",
  "num_op_rev_tl", "num_il_tl", "num_bc_tl", "num_actv_rev_tl",
  "num_actv_bc_tl", "num_accts_ever_120_pd", "mo_sin_rcnt_tl",
  "total_rev_hi_lim", "tot_cur_bal", "tot_coll_amt",
  "mo_sin_rcnt_rev_tl_op", "mo_sin_old_rev_tl_op", "avg_cur_bal",
  "pct_tl_nvr_dlq", "num_tl_120dpd_2m", "mo_sin_old_il_acct",
  "mths_since_recent_inq"
)

# Loop
for (col in columns_to_impute) {
  lc_imputed_2[[col]][is.na(lc_imputed_2[[col]])] <- mean(lc_imputed_2[[col]], na.rm = TRUE)
```

```
  cat("NAs in", col, "nach Imputation:", sum(is.na(lc_imputed_2[[col]])), "\n") # implemented in lc_imp
}
```

```
## NAs in pub_rec_bankruptcies nach Imputation: 0
## NAs in total_bal_ex_mort nach Imputation: 0
## NAs in mths_since_recent_bc nach Imputation: 0
## NAs in bc_open_to_buy nach Imputation: 0
## NAs in percent_bc_gt_75 nach Imputation: 0
## NAs in num_bc_sats nach Imputation: 0
## NAs in total_il_high_credit_limit nach Imputation: 0
## NAs in tot_hi_cred_lim nach Imputation: 0
## NAs in num_tl_op_past_12m nach Imputation: 0
## NAs in num_tl_90g_dpd_24m nach Imputation: 0
## NAs in num_tl_30dpd nach Imputation: 0
## NAs in num_rev_tl_bal_gt_0 nach Imputation: 0
## NAs in num_op_rev_tl nach Imputation: 0
## NAs in num_il_tl nach Imputation: 0
## NAs in num_bc_tl nach Imputation: 0
## NAs in num_actv_rev_tl nach Imputation: 0
## NAs in num_actv_bc_tl nach Imputation: 0
## NAs in num_accts_ever_120_pd nach Imputation: 0
## NAs in mo_sin_rcnt_tl nach Imputation: 0
## NAs in total_rev_hi_lim nach Imputation: 0
## NAs in tot_cur_bal nach Imputation: 0
## NAs in tot_coll_amt nach Imputation: 0
## NAs in mo_sin_rcnt_rev_tl_op nach Imputation: 0
## NAs in mo_sin_old_rev_tl_op nach Imputation: 0
## NAs in avg_cur_bal nach Imputation: 0
## NAs in pct_tl_nvr_dlq nach Imputation: 0
## NAs in num_tl_120dpd_2m nach Imputation: 0
## NAs in mo_sin_old_il_acct nach Imputation: 0
## NAs in mths_since_recent_inq nach Imputation: 0
```

```
## MVI ##

impute_vars <- c(
  "pub_rec_bankruptcies", "total_bal_ex_mort", "mths_since_recent_bc", "bc_open_to_buy",
  "percent_bc_gt_75", "num_bc_sats", "total_il_high_credit_limit", "tot_hi_cred_lim",
  "num_tl_op_past_12m", "num_tl_90g_dpd_24m", "num_tl_30dpd", "num_rev_tl_bal_gt_0",
  "num_op_rev_tl", "num_il_tl", "num_bc_tl", "num_actv_rev_tl", "num_actv_bc_tl",
  "num_accts_ever_120_pd", "mo_sin_rcnt_tl", "total_rev_hi_lim", "tot_cur_bal",
  "tot_coll_amt", "mo_sin_rcnt_rev_tl_op", "mo_sin_old_rev_tl_op", "avg_cur_bal",
  "pct_tl_nvr_dlq", "num_tl_120dpd_2m", "mo_sin_old_il_acct", "mths_since_recent_inq"
)

# Loop
for (var in impute_vars) {
  mvi_name <- paste0(var, "_MVI")
  lc_imputed_2[[mvi_name]] <- ifelse(is.na(lc_imputed_2[[var]]), 1, 0) # implemented in lc_imputed_2
}
```

To further improve our performance, we decided to continue using mean imputation and missing value indicators, but in this model we included significantly more variables than before. However, the procedure

remained the same as previously. Due to the large number of variables in this example, we worked with loops to keep the code as concise as possible.

```
## Split ##

# Split Training and Testing Data
data_split <- initial_split(lc_imputed_2, prop = 0.9, strata = default)

sub_lc_train3 <- training(data_split)
sub_lc_test3 <- testing(data_split)

logit_lc_improvement <- glm(default ~
  delinq_amnt + acc_now_delinq + initial_list_status + revol_bal + pub_rec +
  mort_acc + mort_acc_MVI +
  acc_open_past_24mths + acc_open_past_24mths_MVI +
  num_sats + num_sats_MVI +
  num_rev_accts + num_rev_accts_MVI +
  total_bal_ex_mort + total_bal_ex_mort_MVI +
  mths_since_recent_bc + mths_since_recent_bc_MVI +
  bc_open_to_buy + bc_open_to_buy_MVI +
  percent_bc_gt_75 + percent_bc_gt_75_MVI +
  num_bc_sats + num_bc_sats_MVI +
  total_il_high_credit_limit + total_il_high_credit_limit_MVI +
  tot_hi_cred_lim + tot_hi_cred_lim_MVI +
  num_tl_op_past_12m + num_tl_op_past_12m_MVI +
  num_tl_90g_dpd_24m + num_tl_90g_dpd_24m_MVI +
  num_tl_30dpd + num_tl_30dpd_MVI +
  num_rev_tl_bal_gt_0 + num_rev_tl_bal_gt_0_MVI +
  num_op_rev_tl + num_op_rev_tl_MVI +
  num_il_tl + num_il_tl_MVI +
  num_bc_tl + num_bc_tl_MVI +
  num_actv_rev_tl + num_actv_rev_tl_MVI +
  num_actv_bc_tl + num_actv_bc_tl_MVI +
  num_accts_ever_120_pd + num_accts_ever_120_pd_MVI +
  mo_sin_rcnt_tl + mo_sin_rcnt_tl_MVI +
  total_rev_hi_lim + total_rev_hi_lim_MVI +
  tot_cur_bal + tot_cur_bal_MVI +
  tot_coll_amt + tot_coll_amt_MVI +
  mo_sin_rcnt_rev_tl_op + mo_sin_rcnt_rev_tl_op_MVI +
  mo_sin_old_rev_tl_op + mo_sin_old_rev_tl_op_MVI +
  avg_cur_bal + avg_cur_bal_MVI +
  pct_tl_nvr_dlq + pct_tl_nvr_dlq_MVI +
  num_tl_120dpd_2m + num_tl_120dpd_2m_MVI +
  mo_sin_old_il_acct + mo_sin_old_il_acct_MVI +
  mths_since_recent_inq + mths_since_recent_inq_MVI +
  pub_rec_bankruptcies + pub_rec_bankruptcies_MVI +
  total_bc_limit + total_bc_limit_MVI +
  bc_util + bc_util_MVI,
  data = sub_lc_train3,
  family = binomial(link = "logit")
)

summary(logit_lc_improvement)
```

```
##
## Call:
## glm(formula = default ~ delinq_amnt + acc_now_delinq + initial_list_status +
##     revol_bal + pub_rec + mort_acc + mort_acc_MVI + acc_open_past_24mths +
##     acc_open_past_24mths_MVI + num_sats + num_sats_MVI + num_rev_accts +
##     num_rev_accts_MVI + total_bal_ex_mort + total_bal_ex_mort_MVI +
##     mths_since_recent_bc + mths_since_recent_bc_MVI + bc_open_to_buy +
##     bc_open_to_buy_MVI + percent_bc_gt_75 + percent_bc_gt_75_MVI +
##     num_bc_sats + num_bc_sats_MVI + total_il_high_credit_limit +
##     total_il_high_credit_limit_MVI + tot_hi_cred_lim + tot_hi_cred_lim_MVI +
##     num_tl_op_past_12m + num_tl_op_past_12m_MVI + num_tl_90g_dpd_24m +
##     num_tl_90g_dpd_24m_MVI + num_tl_30dpd + num_tl_30dpd_MVI +
##     num_rev_tl_bal_gt_0 + num_rev_tl_bal_gt_0_MVI + num_op_rev_tl +
##     num_op_rev_tl_MVI + num_il_tl + num_il_tl_MVI + num_bc_tl +
##     num_bc_tl_MVI + num_actv_rev_tl + num_actv_rev_tl_MVI + num_actv_bc_tl +
##     num_actv_bc_tl_MVI + num_accts_ever_120_pd + num_accts_ever_120_pd_MVI +
##     mo_sin_rcnt_tl + mo_sin_rcnt_tl_MVI + total_rev_hi_lim +
##     total_rev_hi_lim_MVI + tot_cur_bal + tot_cur_bal_MVI + tot_coll_amt +
##     tot_coll_amt_MVI + mo_sin_rcnt_rev_tl_op + mo_sin_rcnt_rev_tl_op_MVI +
##     mo_sin_old_rev_tl_op + mo_sin_old_rev_tl_op_MVI + avg_cur_bal +
##     avg_cur_bal_MVI + pct_tl_nvr_dlq + pct_tl_nvr_dlq_MVI + num_tl_120dpd_2m +
##     num_tl_120dpd_2m_MVI + mo_sin_old_il_acct + mo_sin_old_il_acct_MVI +
##     mths_since_recent_inq + mths_since_recent_inq_MVI + pub_rec_bankruptcies +
##     pub_rec_bankruptcies_MVI + total_bc_limit + total_bc_limit_MVI +
##     bc_util + bc_util_MVI, family = binomial(link = "logit"),
##     data = sub_lc_train3)
##
## Coefficients: (35 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.330e+00  6.295e-02 -21.135  < 2e-16 ***
## delinq_amnt                   1.299e-05  5.765e-06   2.254 0.024219 *
## acc_now_delinq                1.930e-01  9.810e-02   1.968 0.049114 *
## initial_list_statusw          1.329e-01  8.039e-03  16.528  < 2e-16 ***
## revol_bal                     5.491e-06  9.218e-07   5.957 2.57e-09 ***
## pub_rec                       2.247e-02  8.104e-03   2.773 0.005562 **
## mort_acc                     -3.417e-02  2.603e-03 -13.130  < 2e-16 ***
## mort_acc_MVI                         NA         NA      NA       NA
## acc_open_past_24mths          7.266e-02  2.087e-03  34.822  < 2e-16 ***
## acc_open_past_24mths_MVI             NA         NA      NA       NA
## num_sats                     -9.127e-03  2.000e-03  -4.564 5.02e-06 ***
## num_sats_MVI                         NA         NA      NA       NA
## num_rev_accts                -1.220e-02  1.584e-03  -7.697 1.39e-14 ***
## num_rev_accts_MVI                    NA         NA      NA       NA
## total_bal_ex_mort             4.020e-06  4.307e-07   9.333  < 2e-16 ***
## total_bal_ex_mort_MVI                NA         NA      NA       NA
## mths_since_recent_bc         -2.054e-03  1.978e-04 -10.385  < 2e-16 ***
## mths_since_recent_bc_MVI             NA         NA      NA       NA
## bc_open_to_buy               -7.028e-06  1.195e-06  -5.879 4.12e-09 ***
## bc_open_to_buy_MVI                   NA         NA      NA       NA
## percent_bc_gt_75              4.240e-03  2.110e-04  20.090  < 2e-16 ***
## percent_bc_gt_75_MVI                 NA         NA      NA       NA
## num_bc_sats                   2.001e-02  3.984e-03   5.023 5.10e-07 ***
## num_bc_sats_MVI                      NA         NA      NA       NA
## total_il_high_credit_limit   -2.137e-06  4.157e-07  -5.141 2.73e-07 ***
```

```
## total_il_high_credit_limit_MVI         NA         NA      NA       NA
## tot_hi_cred_lim                -4.792e-07  2.585e-07  -1.854 0.063791 .
## tot_hi_cred_lim_MVI                    NA         NA      NA       NA
## num_tl_op_past_12m              3.602e-02  3.444e-03  10.457  < 2e-16 ***
## num_tl_op_past_12m_MVI                 NA         NA      NA       NA
## num_tl_90g_dpd_24m              6.670e-02  8.248e-03   8.086 6.15e-16 ***
## num_tl_90g_dpd_24m_MVI                 NA         NA      NA       NA
## num_tl_30dpd                    1.652e-01  1.162e-01   1.422 0.154978
## num_tl_30dpd_MVI                       NA         NA      NA       NA
## num_rev_tl_bal_gt_0             8.243e-02  8.634e-03   9.548  < 2e-16 ***
## num_rev_tl_bal_gt_0_MVI                NA         NA      NA       NA
## num_op_rev_tl                   8.619e-03  3.393e-03   2.540 0.011087 *
## num_op_rev_tl_MVI                      NA         NA      NA       NA
## num_il_tl                      -5.650e-03  8.482e-04  -6.661 2.72e-11 ***
## num_il_tl_MVI                          NA         NA      NA       NA
## num_bc_tl                      -1.858e-03  2.348e-03  -0.791 0.428682
## num_bc_tl_MVI                          NA         NA      NA       NA
## num_actv_rev_tl                -2.847e-02  8.262e-03  -3.446 0.000570 ***
## num_actv_rev_tl_MVI                    NA         NA      NA       NA
## num_actv_bc_tl                 -2.642e-02  5.182e-03  -5.098 3.43e-07 ***
## num_actv_bc_tl_MVI                     NA         NA      NA       NA
## num_accts_ever_120_pd           4.331e-03  3.990e-03   1.085 0.277745
## num_accts_ever_120_pd_MVI              NA         NA      NA       NA
## mo_sin_rcnt_tl                 -7.697e-03  7.165e-04 -10.743  < 2e-16 ***
## mo_sin_rcnt_tl_MVI                     NA         NA      NA       NA
## total_rev_hi_lim               -6.838e-06  7.936e-07  -8.617  < 2e-16 ***
## total_rev_hi_lim_MVI                   NA         NA      NA       NA
## tot_cur_bal                    -5.726e-07  2.857e-07  -2.004 0.045043 *
## tot_cur_bal_MVI                        NA         NA      NA       NA
## tot_coll_amt                   -1.091e-07  3.946e-07  -0.277 0.782130
## tot_coll_amt_MVI                       NA         NA      NA       NA
## mo_sin_rcnt_rev_tl_op           2.578e-03  4.180e-04   6.166 6.99e-10 ***
## mo_sin_rcnt_rev_tl_op_MVI              NA         NA      NA       NA
## mo_sin_old_rev_tl_op           -3.473e-04  5.246e-05  -6.619 3.62e-11 ***
## mo_sin_old_rev_tl_op_MVI               NA         NA      NA       NA
## avg_cur_bal                    -2.870e-06  7.680e-07  -3.737 0.000186 ***
## avg_cur_bal_MVI                        NA         NA      NA       NA
## pct_tl_nvr_dlq                 -2.803e-03  6.061e-04  -4.625 3.75e-06 ***
## pct_tl_nvr_dlq_MVI                     NA         NA      NA       NA
## num_tl_120dpd_2m               -1.017e-01  1.652e-01  -0.615 0.538233
## num_tl_120dpd_2m_MVI                   NA         NA      NA       NA
## mo_sin_old_il_acct             -9.623e-05  8.742e-05  -1.101 0.270977
## mo_sin_old_il_acct_MVI                 NA         NA      NA       NA
## mths_since_recent_inq          -1.680e-02  7.819e-04 -21.488  < 2e-16 ***
## mths_since_recent_inq_MVI              NA         NA      NA       NA
## pub_rec_bankruptcies           -2.189e-02  1.368e-02  -1.600 0.109578
## pub_rec_bankruptcies_MVI               NA         NA      NA       NA
## total_bc_limit                  1.397e-06  6.322e-07   2.209 0.027143 *
## total_bc_limit_MVI                     NA         NA      NA       NA
## bc_util                         2.324e-04  3.245e-04   0.716 0.473946
## bc_util_MVI                            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 406792  on 402115  degrees of freedom
## Residual deviance: 390962  on 402075  degrees of freedom
##   (47883 observations deleted due to missingness)
## AIC: 391044
##
## Number of Fisher Scoring iterations: 5
```

As with the previous models, the dataset was split into training and test data here as well. The syntax and structure of the text remain the same compared to the previous logistic model. The only difference lies in the newly added variables and the corresponding modified data input. The complete list of variables used can be seen in the output of the code. It is noticeable that, according to the output, 35 variables could not be estimated due to singularities. Based on our research, this could be due to a lack of variance in these variables.

The output also shows the results regarding the statistical significance of the individual variables. Each variable that has at least one * symbol next to its p-value is below the 5% significance level, and is therefore significant for the model results. As can be seen in the code output, this applies to the vast majority of all variables used, indicating that a good selection of variables was made.

```r
## AUC Calculation ##

# Predict probabilities on test data
sub_lc_pred_3 <- predict(logit_lc_improvement, newdata = sub_lc_test3, type = "response")

# Calculate ROC and AUC
roc_logit <- roc(response = sub_lc_test3$default, predictor = sub_lc_pred_3)
```

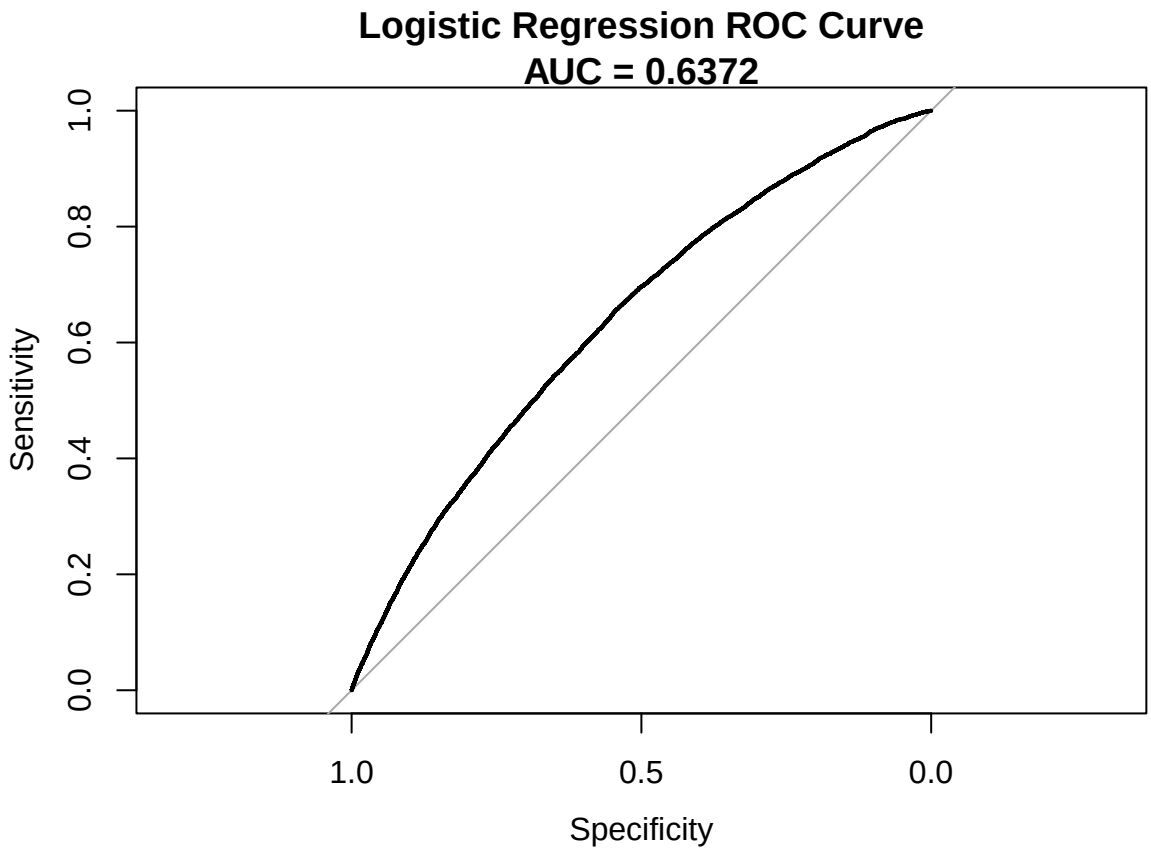
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# Print AUC
auc_logit <- auc(roc_logit)
print(paste("AUC =", round(auc_logit, 4)))
```

```
## [1] "AUC = 0.6372"
```

```r
# Plot ROC Curve
plot(roc_logit, main = paste("Logistic Regression ROC Curve\nAUC =", round(auc_logit, 4)))
```



In this further improved model, we achieved an AUC value of 0.6364. Despite the significantly larger number of variables, which were supplemented with mean imputation and missing value indicators, the improvement

was only minimal. Our AUC score increased only slightly, from 0.6148 to 0.6364, which is practically negligible.

This result was very surprising to us because, given the substantial increase in the number of variables, we expected a higher improvement in AUC. Therefore, we decided to try out new types of machine learning models to ultimately achieve a better score.

Decision Tree Model

A new type of machine learning model that we wanted to try out was a decision tree model.

```
# Load libraries
library(C50)
library(rpart)
```

```
##
## Attaching package: 'rpart'

## The following object is masked from 'package:dials':
##
##     prune
```

```
library(rpart.plot)

lc <- read.csv("/Users/beatkessler/Desktop/lct.csv")
str(lc[0:5])
```

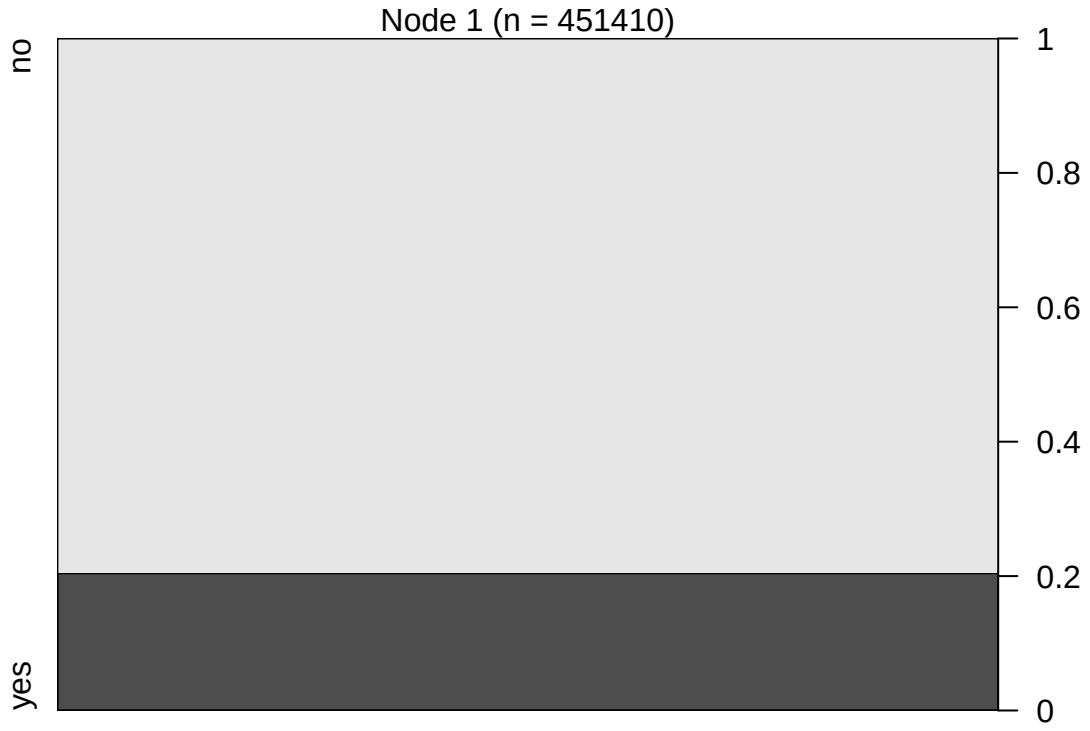
```
## 'data.frame':    500000 obs. of  5 variables:
##  $ id       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ default  : int  0 0 0 0 1 0 0 0 0 1 ...
##  $ loan_amnt: int  16000 13200 18350 6025 35000 16900 10000 2500 21000 14975 ...
##  $ term     : chr  " 36 months" " 36 months" " 60 months" " 36 months" ...
##  $ emp_title: chr  "Toledo Molding And Die" "" "District Supervisor" "" ...
```

```
lc$default <- factor(lc$default, levels = c(0, 1), labels = c("no", "yes"))

library(C50)

# Decision Tree Model

m.c50 <- C5.0(default ~ tot_hi_cred_lim + total_bal_ex_mort +
                total_bc_limit + total_il_high_credit_limit,
              data = lc)

predict(m.c50, lc[1:10, ], type = "prob")
```

```
##          no      yes
## 1  0.800704 0.199296
## 2  0.800704 0.199296
## 3  0.800704 0.199296
## 4  0.800704 0.199296
## 5  0.800704 0.199296
```

```
## 6  0.800704 0.199296
## 7  0.800704 0.199296
## 8  0.800704 0.199296
## 9  0.800704 0.199296
## 10 0.800704 0.199296
```

```r
# summary(m.c50)
# Visualization

m.rpart <- rpart(default ~ tot_hi_cred_lim + total_bal_ex_mort +
                   total_bc_limit + total_il_high_credit_limit,
                 data = lc, method = "class")

# rpart.plot(m.rpart, type = 2, extra = 104) # other Visualization Plot

plot(m.c50)
```



To run this new type of machine learning model, we once again imported the dataset and created a model using a total of four variables (tot_hi_cred_lim, total_bal_ex_mort, total_bc_limit, total_il_high_credit_limit) with the C5.0 library as well as with rpart. The goal, as with all previous models, was to predict the default variable as accurately as possible.

However, in the final plot, it can be seen that the tree created only a single node and did not split any further. This means that the model was unable to find a meaningful separation of the target variable.

After discussions with other teams, it turned out that the majority faced the same issues. Possible reasons for this could be varied. For example, class imbalance might be the problem, or the selected variables simply do

not differ sufficiently. Other reasons could include general data problems such as missing values or unscaled outliers.

Since many other teams had similar problems and due to the limited remaining time in the course, we decided not to pursue the decision tree any further and instead, as described in the following section, to dedicate our time to a random forest model.

Random Forest Model

```
# Load libraries and data
library(caret)
library(pROC)
lc_imputed_train <- read.csv("/Users/beatkessler/Desktop/lct.csv", stringsAsFactors = FALSE)
set.seed(300)

# MVI
impute_vars <- c(
  "pub_rec", "revol_bal", "loan_amnt", "open_acc",
  "inq_last_6mths", "fico_range_high", "fico_range_low"
)

# Add MVI columns
for (var in impute_vars) {
  mvi_name <- paste0(var, "_MVI")
  lc_imputed_train[[mvi_name]] <- ifelse(is.na(lc_imputed_train[[var]]), 1, 0)
}

# Imputation

# Impute pub_rec
lc_imputed_train$pub_rec[is.na(lc_imputed_train$pub_rec)] <- mean(lc_imputed_train$pub_rec, na.rm = TRU
print(sum(is.na(lc_imputed_train$pub_rec)))
```

```
## [1] 0
```

```
# Impute revol_bal
lc_imputed_train$revol_bal[is.na(lc_imputed_train$revol_bal)] <- mean(lc_imputed_train$revol_bal, na.rm
print(sum(is.na(lc_imputed_train$revol_bal)))
```

```
## [1] 0
```

```
# Impute loan_amnt
lc_imputed_train$loan_amnt[is.na(lc_imputed_train$loan_amnt)] <- mean(lc_imputed_train$loan_amnt, na.rm
print(sum(is.na(lc_imputed_train$loan_amnt)))
```

```
## [1] 0
```

```
# Impute open_acc
lc_imputed_train$open_acc[is.na(lc_imputed_train$open_acc)] <- mean(lc_imputed_train$open_acc, na.rm = 
print(sum(is.na(lc_imputed_train$open_acc)))
```

```
## [1] 0
```

```
# Impute inq_last_6mths
lc_imputed_train$inq_last_6mths[is.na(lc_imputed_train$inq_last_6mths)] <- mean(lc_imputed_train$inq_las
print(sum(is.na(lc_imputed_train$inq_last_6mths)))
```

```
## [1] 0
```

```
# Impute fico_range_high
lc_imputed_train$fico_range_high[is.na(lc_imputed_train$fico_range_high)] <- mean(lc_imputed_train$fico
print(sum(is.na(lc_imputed_train$fico_range_high)))
```

```
## [1] 0
```

```
# Impute fico_range_low
lc_imputed_train$fico_range_low[is.na(lc_imputed_train$fico_range_low)] <- mean(lc_imputed_train$fico_ra
print(sum(is.na(lc_imputed_train$fico_range_low)))
```

```
## [1] 0
```

```
lc_imputed_train$default <- factor(lc_imputed_train$default, levels = c(0, 1), labels = c("no", "defaul

# Training
ctrl <- trainControl(
  method = "cv", number = 2,
  selectionFunction = "best",
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = TRUE,
  allowParallel = FALSE,
  returnData = TRUE,
  trim = TRUE,
  verboseIter = TRUE
)

grid.ranger <- expand.grid(mtry = 5,
                           splitrule = "gini",
                           min.node.size = 5)

m.ranger <- train(default ~ pub_rec + open_acc + loan_amnt + term + purpose + fico_range_low + fico_rang
                  method = "ranger",
                  trControl = ctrl,
                  tuneGrid = grid.ranger,
                  metric = "ROC")
```

```
## + Fold1: mtry=5, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 25%. Estimated remaining time: 1 minute, 35 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 1 minute, 3 seconds.
## Growing trees.. Progress: 74%. Estimated remaining time: 32 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 1 seconds.
## - Fold1: mtry=5, splitrule=gini, min.node.size=5
```

```
## + Fold2: mtry=5, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 24%. Estimated remaining time: 1 minute, 37 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 1 minute, 5 seconds.
## Growing trees.. Progress: 73%. Estimated remaining time: 34 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 3 seconds.
## - Fold2: mtry=5, splitrule=gini, min.node.size=5
## Aggregating results
## Fitting final model on full training set
## Growing trees.. Progress: 11%. Estimated remaining time: 4 minutes, 21 seconds.
## Growing trees.. Progress: 22%. Estimated remaining time: 3 minutes, 42 seconds.
## Growing trees.. Progress: 33%. Estimated remaining time: 3 minutes, 7 seconds.
## Growing trees.. Progress: 45%. Estimated remaining time: 2 minutes, 35 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 2 minutes, 4 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 1 minute, 34 seconds.
## Growing trees.. Progress: 78%. Estimated remaining time: 1 minute, 3 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 32 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 1 seconds.
```

```r
# Prediction/ Generating prediction probabilities

lc_imputed_test <- read.csv("lending_club_test.csv", stringsAsFactors = FALSE)

lc_imputed_test$prob <- predict(m.ranger, lc_imputed_test, type = "prob")[, 2]

# write.csv(lc_imputed_2_test[c("id", "prob")], "20250618_GSERM_MLwR_Day 3 Assignment_rfRanger_Version1
```

The performance improvements of the logistic regression models, combined with diminishing returns and results peaking at approx. 0.67 AUC encouraged us to tackle a more advanced model. Random forest quickly became our choice as it was just discussed in class.

Following the example in class, the ranger approach within the caret package was employed. We ran two separate models entailing one smaller set of features (shown in the chunk above) and one more extensive feature set. Given our progress on encoding missing value imputations and indicators, we only included imputed values. This analysis did not yet include MVIs which might have been one of the reasons why both initial Random Forest models turned out worse according to feedback by Brett than the previously best-performing logit model. It became evident that even more sophistication was necessary. Thus, we turned to an ensemble approach outlined in the subsequent section.

Ensemble | Random Forest and Logistic Regression Model

```r
## Prep ##
set.seed(300)
library(rsample)
library(caret)
library(pROC)

## Import ##
ensemb_lc_train <- read.csv("/Users/beatkessler/Desktop/lct.csv", stringsAsFactors = FALSE)

ensemb_lc_train$default <- factor(ensemb_lc_train$default, levels = c(0, 1), labels = c("no", "default")
ensemb_lc_train$initial_list_status= as.factor(ensemb_lc_train$initial_list_status)
ensemb_lc_train$home_ownership= as.factor(ensemb_lc_train$home_ownership)
```

```r
ensemb_lc_train$purpose= as.factor(ensemb_lc_train$purpose)
ensemb_lc_train$addr_state= as.factor(ensemb_lc_train$addr_state)
ensemb_lc_train$term= as.factor(ensemb_lc_train$term)
ensemb_lc_train$num_rev_accts= as.numeric(ensemb_lc_train$num_rev_accts)

#ensemb_lc_train$fico_range_low = as.numeric(unlist(ensemb_lc_train$fico_range_low))
#ensemb_lc_train$fico_range_high = as.numeric(unlist(ensemb_lc_train$fico_range_high))

## MVIs ##
impute_vars <- c(
  "pub_rec_bankruptcies", "total_bal_ex_mort", "mths_since_recent_bc", "bc_open_to_buy",
  "percent_bc_gt_75", "num_bc_sats", "total_il_high_credit_limit", "tot_hi_cred_lim",
  "num_tl_op_past_12m", "num_tl_90g_dpd_24m", "num_tl_30dpd", "num_rev_tl_bal_gt_0",
  "num_op_rev_tl", "num_il_tl", "num_bc_tl", "num_actv_rev_tl", "num_actv_bc_tl",
  "num_accts_ever_120_pd", "mo_sin_rcnt_tl", "total_rev_hi_lim", "tot_cur_bal",
  "tot_coll_amt", "mo_sin_rcnt_rev_tl_op", "mo_sin_old_rev_tl_op", "avg_cur_bal",
  "pct_tl_nvr_dlq", "num_tl_120dpd_2m", "mo_sin_old_il_acct", "mths_since_recent_inq",
  "annual_inc", "delinq_2yrs", "loan_amnt", "dti", "fico_range_low", "fico_range_high",
  "inq_last_6mths", "open_acc", "mths_since_last_delinq", "mths_since_last_record",
  "collections_12_mths_ex_med", "mths_since_last_major_derog",
  "mths_since_recent_bc_dlq", "mths_since_recent_revol_delinq",
  "num_rev_accts", "num_sats", "acc_open_past_24mths", "bc_util", "inq_last_12m", "total_cu_tl",
  "term"
)

# Add MVI columns
for (var in impute_vars) {
  mvi_name <- paste0(var, "_MVI")
  ensemb_lc_train[[mvi_name]] <- ifelse(is.na(ensemb_lc_train[[var]]), 1, 0)
}

# List all column names to impute
columns_to_impute <- c(
  "pub_rec_bankruptcies", "total_bal_ex_mort", "mths_since_recent_bc", "bc_open_to_buy",
  "percent_bc_gt_75", "num_bc_sats", "total_il_high_credit_limit", "tot_hi_cred_lim",
  "num_tl_op_past_12m", "num_tl_90g_dpd_24m", "num_tl_30dpd", "num_rev_tl_bal_gt_0",
  "num_op_rev_tl", "num_il_tl", "num_bc_tl", "num_actv_rev_tl", "num_actv_bc_tl",
  "num_accts_ever_120_pd", "mo_sin_rcnt_tl", "total_rev_hi_lim", "tot_cur_bal",
  "tot_coll_amt", "mo_sin_rcnt_rev_tl_op", "mo_sin_old_rev_tl_op", "avg_cur_bal",
  "pct_tl_nvr_dlq", "num_tl_120dpd_2m", "mo_sin_old_il_acct", "mths_since_recent_inq",
  "annual_inc", "delinq_2yrs", "loan_amnt", "dti", "fico_range_low", "fico_range_high",
  "inq_last_6mths", "open_acc", "mths_since_last_delinq", "mths_since_last_record",
  "collections_12_mths_ex_med", "mths_since_last_major_derog", "mths_since_recent_bc_dlq",
  "mths_since_recent_revol_delinq", "num_rev_accts", "num_sats", "acc_open_past_24mths",
  "bc_util", "inq_last_12m", "total_cu_tl"
)

# Loop over columns and impute
for (col in columns_to_impute) {
  mean_value <- mean(ensemb_lc_train[[col]], na.rm = TRUE)
  ensemb_lc_train[[col]][is.na(ensemb_lc_train[[col]])] <- mean_value
  cat("Remaining NAs in", col, ":", sum(is.na(ensemb_lc_train[[col]])), "\n")
}
```

```
## Remaining NAs in pub_rec_bankruptcies : 0
## Remaining NAs in total_bal_ex_mort : 0
## Remaining NAs in mths_since_recent_bc : 0
## Remaining NAs in bc_open_to_buy : 0
## Remaining NAs in percent_bc_gt_75 : 0
## Remaining NAs in num_bc_sats : 0
## Remaining NAs in total_il_high_credit_limit : 0
## Remaining NAs in tot_hi_cred_lim : 0
## Remaining NAs in num_tl_op_past_12m : 0
## Remaining NAs in num_tl_90g_dpd_24m : 0
## Remaining NAs in num_tl_30dpd : 0
## Remaining NAs in num_rev_tl_bal_gt_0 : 0
## Remaining NAs in num_op_rev_tl : 0
## Remaining NAs in num_il_tl : 0
## Remaining NAs in num_bc_tl : 0
## Remaining NAs in num_actv_rev_tl : 0
## Remaining NAs in num_actv_bc_tl : 0
## Remaining NAs in num_accts_ever_120_pd : 0
## Remaining NAs in mo_sin_rcnt_tl : 0
## Remaining NAs in total_rev_hi_lim : 0
## Remaining NAs in tot_cur_bal : 0
## Remaining NAs in tot_coll_amt : 0
## Remaining NAs in mo_sin_rcnt_rev_tl_op : 0
## Remaining NAs in mo_sin_old_rev_tl_op : 0
## Remaining NAs in avg_cur_bal : 0
## Remaining NAs in pct_tl_nvr_dlq : 0
## Remaining NAs in num_tl_120dpd_2m : 0
## Remaining NAs in mo_sin_old_il_acct : 0
## Remaining NAs in mths_since_recent_inq : 0
## Remaining NAs in annual_inc : 0
## Remaining NAs in delinq_2yrs : 0
## Remaining NAs in loan_amnt : 0
## Remaining NAs in dti : 0
## Remaining NAs in fico_range_low : 0
## Remaining NAs in fico_range_high : 0
## Remaining NAs in inq_last_6mths : 0
## Remaining NAs in open_acc : 0
## Remaining NAs in mths_since_last_delinq : 0
## Remaining NAs in mths_since_last_record : 0
## Remaining NAs in collections_12_mths_ex_med : 0
## Remaining NAs in mths_since_last_major_derog : 0
## Remaining NAs in mths_since_recent_bc_dlq : 0
## Remaining NAs in mths_since_recent_revol_delinq : 0
## Remaining NAs in num_rev_accts : 0
## Remaining NAs in num_sats : 0
## Remaining NAs in acc_open_past_24mths : 0
## Remaining NAs in bc_util : 0
## Remaining NAs in inq_last_12m : 0
## Remaining NAs in total_cu_tl : 0
```

```r
## Split "Lending Club TRAIN" dataset into two train + test subsets ##
set.seed(300)
data_split <- initial_split(ensemb_lc_train, prop = 0.9, strata = default)
```

```r
sub_ensemb_lc_train <- training(data_split)
sub_ensemb_lc_test <- testing(data_split)

## Alternative split "Lending Club TRAIN" dataset into three train + test + validation subsets ##

# Define the partition (e.g. 80% of the data for training)
#sub_ensemb_lc_train_forsplit <- createDataPartition(ensemb_lc_train$default, p = .8,
#                                                    list = FALSE,
#                                                    times = 1)

# Split the dataset using the defined partition
#sub_ensemb_lc_train <- ensemb_lc_train[sub_ensemb_lc_train_forsplit, ,drop=FALSE]
#sub_ensemb_lc_test_val <- ensemb_lc_train[-sub_ensemb_lc_train_forsplit, ,drop=FALSE]

# Define a new partition to split the remaining 20%
#sub_ensemb_lc_test_forsplit <- createDataPartition(sub_ensemb_lc_test_val$default,
#                                                   p = .5,
#                                                   list = FALSE,
#                                                   times = 1)

# Split the remaining 20% of the data: 50% (test) and 50% (val)
#sub_ensemb_lc_test <- sub_ensemb_lc_test_val[-sub_ensemb_lc_test_forsplit, ,drop=FALSE]
#sub_ensemb_lc_val <- sub_ensemb_lc_test_val[sub_ensemb_lc_test_forsplit, ,drop=FALSE]

## First ensemble stage, model 1: Logit ##

sub_ensemb_lc_train_logit <- glm(default ~ mths_since_recent_bc_dlq + mths_since_recent_revol_delinq + r
summary(sub_ensemb_lc_train_logit)
```

```
##
## Call:
## glm(formula = default ~ mths_since_recent_bc_dlq + mths_since_recent_revol_delinq +
##     num_rev_accts + num_sats + acc_open_past_24mths + bc_util +
##     inq_last_12m + total_cu_tl + term + initial_list_status +
##     collections_12_mths_ex_med + mths_since_last_major_derog +
##     home_ownership + purpose + fico_range_low + fico_range_high +
##     inq_last_6mths + open_acc + mths_since_last_delinq + mths_since_last_record +
##     annual_inc + delinq_2yrs + loan_amnt + dti + delinq_amnt +
##     acc_now_delinq + initial_list_status + revol_bal + pub_rec +
##     pub_rec_bankruptcies + total_bal_ex_mort + mths_since_recent_bc +
##     bc_open_to_buy + percent_bc_gt_75 + num_bc_sats + total_il_high_credit_limit +
##     tot_hi_cred_lim + num_tl_op_past_12m + num_tl_90g_dpd_24m +
##     num_tl_30dpd + num_rev_tl_bal_gt_0 + num_op_rev_tl + num_il_tl +
##     num_bc_tl + num_actv_rev_tl + num_actv_bc_tl + num_accts_ever_120_pd +
##     mo_sin_rcnt_tl + total_rev_hi_lim + tot_cur_bal + tot_coll_amt +
##     mo_sin_rcnt_rev_tl_op + mo_sin_old_rev_tl_op + avg_cur_bal +
##     pct_tl_nvr_dlq + num_tl_120dpd_2m + mo_sin_old_il_acct +
##     mths_since_recent_inq + pub_rec_bankruptcies_MVI + total_bal_ex_mort_MVI +
##     mths_since_recent_bc_MVI + bc_open_to_buy_MVI + percent_bc_gt_75_MVI +
##     num_bc_sats_MVI + total_il_high_credit_limit_MVI + tot_hi_cred_lim_MVI +
##     num_tl_op_past_12m_MVI + num_tl_90g_dpd_24m_MVI + num_tl_30dpd_MVI +
##     num_rev_tl_bal_gt_0_MVI + num_op_rev_tl_MVI + num_il_tl_MVI +
##     num_bc_tl_MVI + num_actv_rev_tl_MVI + num_actv_bc_tl_MVI +
```

```
##     num_accts_ever_120_pd_MVI + mo_sin_rcnt_tl_MVI + total_rev_hi_lim_MVI +
##     tot_cur_bal_MVI + tot_coll_amt_MVI + mo_sin_rcnt_rev_tl_op_MVI +
##     mo_sin_old_rev_tl_op_MVI + avg_cur_bal_MVI + pct_tl_nvr_dlq_MVI +
##     num_tl_120dpd_2m_MVI + mo_sin_old_il_acct_MVI + mths_since_recent_inq_MVI +
##     annual_inc_MVI + delinq_2yrs_MVI + loan_amnt_MVI + dti_MVI +
##     fico_range_low_MVI + fico_range_high_MVI + inq_last_6mths_MVI +
##     open_acc_MVI + mths_since_last_delinq_MVI + mths_since_last_record_MVI +
##     collections_12_mths_ex_med_MVI + mths_since_last_major_derog_MVI +
##     mths_since_recent_bc_dlq_MVI + mths_since_recent_revol_delinq_MVI +
##     num_rev_accts_MVI + num_sats_MVI + acc_open_past_24mths_MVI +
##     bc_util_MVI + inq_last_12m_MVI + total_cu_tl_MVI + term_MVI,
##     family = binomial(link = "logit"), data = sub_ensemb_lc_train)
##
## Coefficients: (27 not defined because of singularities)
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.073e+00  2.468e+00  -0.840 0.400926
## mths_since_recent_bc_dlq        -4.764e-04  5.575e-04  -0.855 0.392770
## mths_since_recent_revol_delinq  -1.012e-04  5.705e-04  -0.177 0.859264
## num_rev_accts                   -2.182e-02  1.661e-03 -13.135  < 2e-16 ***
## num_sats                         1.690e-02  4.121e-03   4.101 4.11e-05 ***
## acc_open_past_24mths             5.273e-02  2.126e-03  24.803  < 2e-16 ***
## bc_util                         -2.407e-03  3.262e-04  -7.379 1.60e-13 ***
## inq_last_12m                    -1.159e-02  1.049e-02  -1.106 0.268929
## total_cu_tl                     -5.697e-02  1.098e-02  -5.187 2.14e-07 ***
## term 60 months                   9.907e-01  9.433e-03 105.025  < 2e-16 ***
## initial_list_statusw             2.974e-02  8.377e-03   3.550 0.000386 ***
## collections_12_mths_ex_med       1.367e-01  2.823e-02   4.844 1.27e-06 ***
## mths_since_last_major_derog      4.929e-04  4.846e-04   1.017 0.309058
## home_ownershipNONE               3.124e-01  5.171e-01   0.604 0.545804
## home_ownershipOTHER              6.380e-01  2.729e-01   2.338 0.019407 *
## home_ownershipOWN                1.721e-01  1.435e-02  11.993  < 2e-16 ***
## home_ownershipRENT               2.584e-01  1.029e-02  25.111  < 2e-16 ***
## purposecredit_card              -1.319e-01  4.409e-02  -2.991 0.002784 **
## purposedebt_consolidation        2.327e-02  4.347e-02   0.535 0.592389
## purposeeducational               6.844e-01  1.878e-01   3.644 0.000268 ***
## purposehome_improvement          1.531e-01  4.642e-02   3.298 0.000972 ***
## purposehouse                     2.961e-01  7.063e-02   4.192 2.76e-05 ***
## purposemajor_purchase            1.403e-01  5.185e-02   2.707 0.006796 **
## purposemedical                   3.215e-01  5.807e-02   5.537 3.08e-08 ***
## purposemoving                    4.561e-01  6.273e-02   7.271 3.56e-13 ***
## purposeother                     2.749e-01  4.623e-02   5.946 2.75e-09 ***
## purposerenewable_energy          4.386e-01  1.454e-01   3.017 0.002551 **
## purposesmall_business            7.760e-01  5.301e-02  14.639  < 2e-16 ***
## purposevacation                  2.837e-01  6.733e-02   4.214 2.51e-05 ***
## purposewedding                  -2.158e-01  9.206e-02  -2.344 0.019068 *
## fico_range_low                  -1.252e+00  6.166e-01  -2.030 0.042331 *
## fico_range_high                  1.244e+00  6.166e-01   2.018 0.043571 *
## inq_last_6mths                   6.413e-02  4.852e-03  13.217  < 2e-16 ***
## open_acc                        -2.531e-02  3.617e-03  -6.998 2.59e-12 ***
## mths_since_last_delinq          -1.794e-03  4.242e-04  -4.229 2.35e-05 ***
## mths_since_last_record           1.285e-03  3.877e-04   3.314 0.000920 ***
## annual_inc                      -1.832e-06  1.507e-07 -12.156  < 2e-16 ***
## delinq_2yrs                      8.986e-02  7.911e-03  11.359  < 2e-16 ***
## loan_amnt                        2.127e-05  6.413e-07  33.176  < 2e-16 ***
```

```
## dti                                3.023e-02  6.453e-04  46.842  < 2e-16 ***
## delinq_amnt                        3.691e-06  5.976e-06   0.618 0.536793
## acc_now_delinq                     1.071e-01  1.023e-01   1.046 0.295472
## revol_bal                         -3.338e-06  6.979e-07  -4.782 1.73e-06 ***
## pub_rec                            3.314e-02  1.046e-02   3.167 0.001538 **
## pub_rec_bankruptcies              -8.032e-02  1.831e-02  -4.388 1.15e-05 ***
## total_bal_ex_mort                  4.221e-06  3.819e-07  11.051  < 2e-16 ***
## mths_since_recent_bc              -2.629e-03  1.993e-04 -13.190  < 2e-16 ***
## bc_open_to_buy                    -5.297e-06  7.987e-07  -6.632 3.31e-11 ***
## percent_bc_gt_75                   3.676e-03  2.156e-04  17.044  < 2e-16 ***
## num_bc_sats                        8.757e-03  4.335e-03   2.020 0.043370 *
## total_il_high_credit_limit        -5.147e-06  3.903e-07 -13.186  < 2e-16 ***
## tot_hi_cred_lim                   -7.459e-07  2.733e-07  -2.729 0.006343 **
## num_tl_op_past_12m                 4.572e-02  3.525e-03  12.970  < 2e-16 ***
## num_tl_90g_dpd_24m                -4.627e-02  1.298e-02  -3.564 0.000366 ***
## num_tl_30dpd                       2.774e-02  1.205e-01   0.230 0.817878
## num_rev_tl_bal_gt_0                7.694e-02  9.553e-03   8.054 8.04e-16 ***
## num_op_rev_tl                      1.249e-02  3.532e-03   3.536 0.000407 ***
## num_il_tl                         -9.668e-03  8.845e-04 -10.930  < 2e-16 ***
## num_bc_tl                          3.864e-03  2.462e-03   1.570 0.116495
## num_actv_rev_tl                   -4.411e-02  9.645e-03  -4.574 4.79e-06 ***
## num_actv_bc_tl                    -2.023e-02  5.251e-03  -3.853 0.000117 ***
## num_accts_ever_120_pd              1.079e-02  4.744e-03   2.274 0.022987 *
## mo_sin_rcnt_tl                    -2.408e-03  7.296e-04  -3.301 0.000964 ***
## total_rev_hi_lim                  -2.282e-06  6.121e-07  -3.728 0.000193 ***
## tot_cur_bal                        2.643e-07  2.993e-07   0.883 0.377105
## tot_coll_amt                      -1.288e-06  2.299e-06  -0.560 0.575306
## mo_sin_rcnt_rev_tl_op              1.606e-03  4.252e-04   3.776 0.000159 ***
## mo_sin_old_rev_tl_op              -3.595e-04  5.435e-05  -6.614 3.74e-11 ***
## avg_cur_bal                       -3.653e-06  7.911e-07  -4.618 3.87e-06 ***
## pct_tl_nvr_dlq                     4.059e-03  8.051e-04   5.042 4.62e-07 ***
## num_tl_120dpd_2m                   3.343e-02  1.671e-01   0.200 0.841461
## mo_sin_old_il_acct                -1.442e-04  9.144e-05  -1.576 0.114913
## mths_since_recent_inq             -8.391e-03  9.306e-04  -9.018  < 2e-16 ***
## pub_rec_bankruptcies_MVI           5.253e-01  1.339e-01   3.922 8.79e-05 ***
## total_bal_ex_mort_MVI             -3.158e-01  6.130e-02  -5.152 2.58e-07 ***
## mths_since_recent_bc_MVI          -3.589e-01  1.395e-01  -2.573 0.010083 *
## bc_open_to_buy_MVI                 1.600e-01  2.662e-01   0.601 0.547729
## percent_bc_gt_75_MVI               2.331e-02  1.877e-01   0.124 0.901150
## num_bc_sats_MVI                    5.989e-02  4.968e-02   1.206 0.227945
## total_il_high_credit_limit_MVI    -5.790e+00  6.217e+01  -0.093 0.925799
## tot_hi_cred_lim_MVI                       NA         NA      NA       NA
## num_tl_op_past_12m_MVI                    NA         NA      NA       NA
## num_tl_90g_dpd_24m_MVI                    NA         NA      NA       NA
## num_tl_30dpd_MVI                          NA         NA      NA       NA
## num_rev_tl_bal_gt_0_MVI                   NA         NA      NA       NA
## num_op_rev_tl_MVI                         NA         NA      NA       NA
## num_il_tl_MVI                             NA         NA      NA       NA
## num_bc_tl_MVI                             NA         NA      NA       NA
## num_actv_rev_tl_MVI                       NA         NA      NA       NA
## num_actv_bc_tl_MVI                        NA         NA      NA       NA
## num_accts_ever_120_pd_MVI                 NA         NA      NA       NA
## mo_sin_rcnt_tl_MVI                        NA         NA      NA       NA
## total_rev_hi_lim_MVI                      NA         NA      NA       NA
```

```
## tot_cur_bal_MVI                           NA         NA      NA       NA
## tot_coll_amt_MVI                          NA         NA      NA       NA
## mo_sin_rcnt_rev_tl_op_MVI           1.050e+01  4.395e+01   0.239 0.811142
## mo_sin_old_rev_tl_op_MVI                  NA         NA      NA       NA
## avg_cur_bal_MVI                     1.847e+00  7.750e-01   2.384 0.017136 *
## pct_tl_nvr_dlq_MVI                 -2.464e-01  3.105e-01  -0.793 0.427522
## num_tl_120dpd_2m_MVI                1.651e-01  3.253e-02   5.076 3.85e-07 ***
## mo_sin_old_il_acct_MVI              2.642e-01  2.423e-02  10.905  < 2e-16 ***
## mths_since_recent_inq_MVI          -1.411e-01  1.643e-02  -8.592  < 2e-16 ***
## annual_inc_MVI                            NA         NA      NA       NA
## delinq_2yrs_MVI                           NA         NA      NA       NA
## loan_amnt_MVI                             NA         NA      NA       NA
## dti_MVI                             1.066e+01  4.395e+01   0.242 0.808406
## fico_range_low_MVI                        NA         NA      NA       NA
## fico_range_high_MVI                       NA         NA      NA       NA
## inq_last_6mths_MVI                        NA         NA      NA       NA
## open_acc_MVI                              NA         NA      NA       NA
## mths_since_last_delinq_MVI         -1.808e-02  1.357e-02  -1.332 0.182956
## mths_since_last_record_MVI         -5.285e-02  2.118e-02  -2.495 0.012598 *
## collections_12_mths_ex_med_MVI     -9.725e-01  7.499e-01  -1.297 0.194679
## mths_since_last_major_derog_MVI    -4.064e-02  1.387e-02  -2.930 0.003389 **
## mths_since_recent_bc_dlq_MVI       -8.775e-02  1.624e-02  -5.404 6.51e-08 ***
## mths_since_recent_revol_delinq_MVI  6.573e-02  1.705e-02   3.856 0.000115 ***
## num_rev_accts_MVI                  -6.701e+00  4.395e+01  -0.152 0.878837
## num_sats_MVI                              NA         NA      NA       NA
## acc_open_past_24mths_MVI                  NA         NA      NA       NA
## bc_util_MVI                         3.817e-01  1.560e-01   2.447 0.014403 *
## inq_last_12m_MVI                   -2.995e-01  2.947e-02 -10.164  < 2e-16 ***
## total_cu_tl_MVI                           NA         NA      NA       NA
## term_MVI                                  NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 449481  on 449998  degrees of freedom
## Residual deviance: 407408  on 449903  degrees of freedom
## AIC: 407600
##
## Number of Fisher Scoring iterations: 7
```

```
## First ensemble stage, model 2: Random Forest/ Ranger ##

ctrl <- trainControl(
  method = "cv", number = 3,
  selectionFunction = "best",
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = TRUE,
  allowParallel = FALSE,
  returnData = TRUE,
  trim = TRUE,
  verboseIter = TRUE
)
```

```r
grid.ranger <- expand.grid(mtry = 5,
                           splitrule = "gini",
                           min.node.size = 5)

sub_ensemb_lc_train_mranger <- train(default ~ mths_since_recent_bc_dlq + mths_since_recent_revol_delinc
                                     method = "ranger",
                                     trControl = ctrl,
                                     tuneGrid = grid.ranger,
                                     metric = "ROC")
```

```
## + Fold1: mtry=5, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 13%. Estimated remaining time: 3 minutes, 20 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 2 minutes, 47 seconds.
## Growing trees.. Progress: 40%. Estimated remaining time: 2 minutes, 17 seconds.
## Growing trees.. Progress: 54%. Estimated remaining time: 1 minute, 47 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 1 minute, 16 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 45 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 13 seconds.
## - Fold1: mtry=5, splitrule=gini, min.node.size=5
## + Fold2: mtry=5, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 13%. Estimated remaining time: 3 minutes, 20 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 2 minutes, 47 seconds.
## Growing trees.. Progress: 40%. Estimated remaining time: 2 minutes, 18 seconds.
## Growing trees.. Progress: 54%. Estimated remaining time: 1 minute, 47 seconds.
## Growing trees.. Progress: 68%. Estimated remaining time: 1 minute, 15 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 44 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 12 seconds.
## - Fold2: mtry=5, splitrule=gini, min.node.size=5
## + Fold3: mtry=5, splitrule=gini, min.node.size=5
## Growing trees.. Progress: 13%. Estimated remaining time: 3 minutes, 20 seconds.
## Growing trees.. Progress: 27%. Estimated remaining time: 2 minutes, 49 seconds.
## Growing trees.. Progress: 40%. Estimated remaining time: 2 minutes, 19 seconds.
## Growing trees.. Progress: 54%. Estimated remaining time: 1 minute, 47 seconds.
## Growing trees.. Progress: 54%. Estimated remaining time: 2 minutes, 51 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 1 minute, 53 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 1 minute, 2 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 17 seconds.
## - Fold3: mtry=5, splitrule=gini, min.node.size=5
## Aggregating results
## Fitting final model on full training set
## Growing trees.. Progress: 8%. Estimated remaining time: 6 minutes, 6 seconds.
## Growing trees.. Progress: 16%. Estimated remaining time: 5 minutes, 21 seconds.
## Growing trees.. Progress: 25%. Estimated remaining time: 4 minutes, 45 seconds.
## Growing trees.. Progress: 33%. Estimated remaining time: 4 minutes, 11 seconds.
## Growing trees.. Progress: 41%. Estimated remaining time: 3 minutes, 45 seconds.
## Growing trees.. Progress: 49%. Estimated remaining time: 3 minutes, 19 seconds.
## Growing trees.. Progress: 56%. Estimated remaining time: 2 minutes, 50 seconds.
## Growing trees.. Progress: 64%. Estimated remaining time: 2 minutes, 19 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 1 minute, 52 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 1 minute, 18 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 45 seconds.
## Growing trees.. Progress: 96%. Estimated remaining time: 14 seconds.
## Computing prediction error.. Progress: 98%. Estimated remaining time: 0 seconds.
```

```r
## Predictions based on training subset ##
sub_ensemb_lc_pred_logit1 <- predict(sub_ensemb_lc_train_logit, sub_ensemb_lc_test, type = "response")
sub_ensemb_lc_pred_ranger1 <- predict(sub_ensemb_lc_train_mranger, sub_ensemb_lc_test, type = "prob")[,

## ROCs ##
roc.logit <- roc(response = sub_ensemb_lc_test$default, predictor = sub_ensemb_lc_pred_logit1)
```

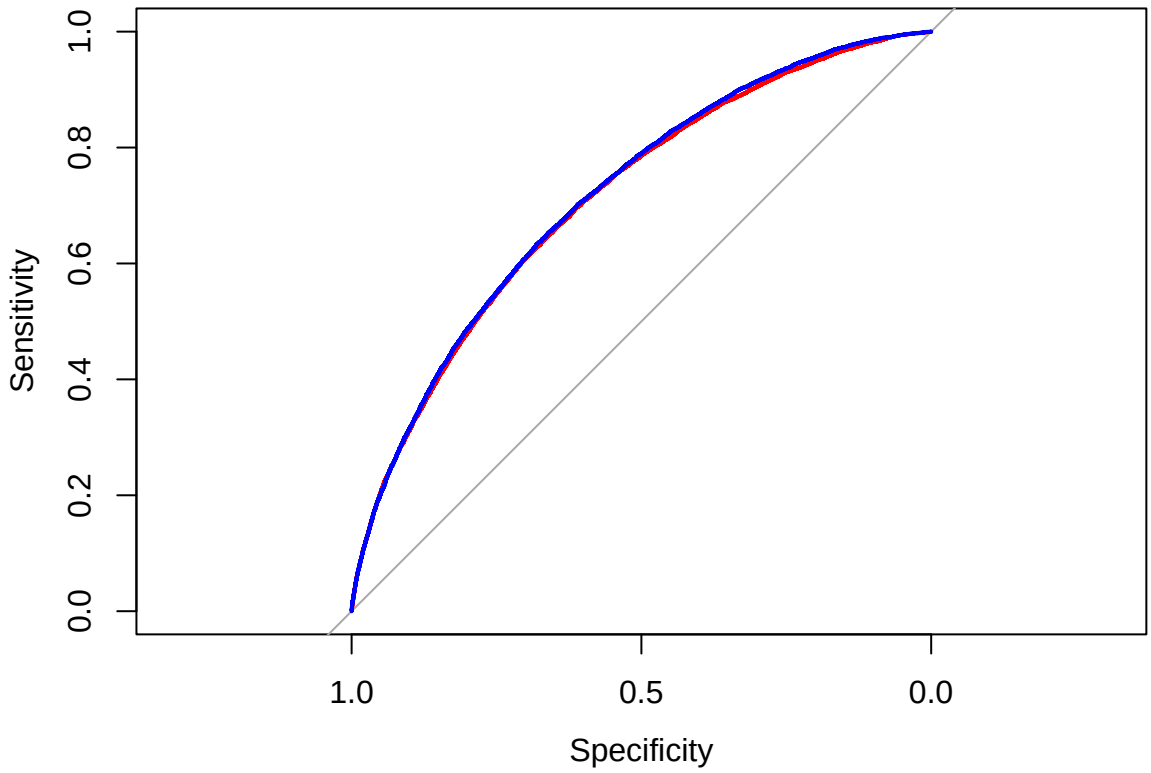
```
## Setting levels: control = no, case = default
```

```
## Setting direction: controls < cases
```

```r
roc.ranger <- roc(response = sub_ensemb_lc_test$default, predictor = sub_ensemb_lc_pred_ranger1)
```

```
## Setting levels: control = no, case = default
## Setting direction: controls < cases
```

```r
auc.logit <- round(auc(roc.logit), 4)
auc.ranger <- round(auc(roc.ranger), 4)

plot(roc.logit, col = "red")
plot(roc.ranger, col = "blue", add = TRUE)
```



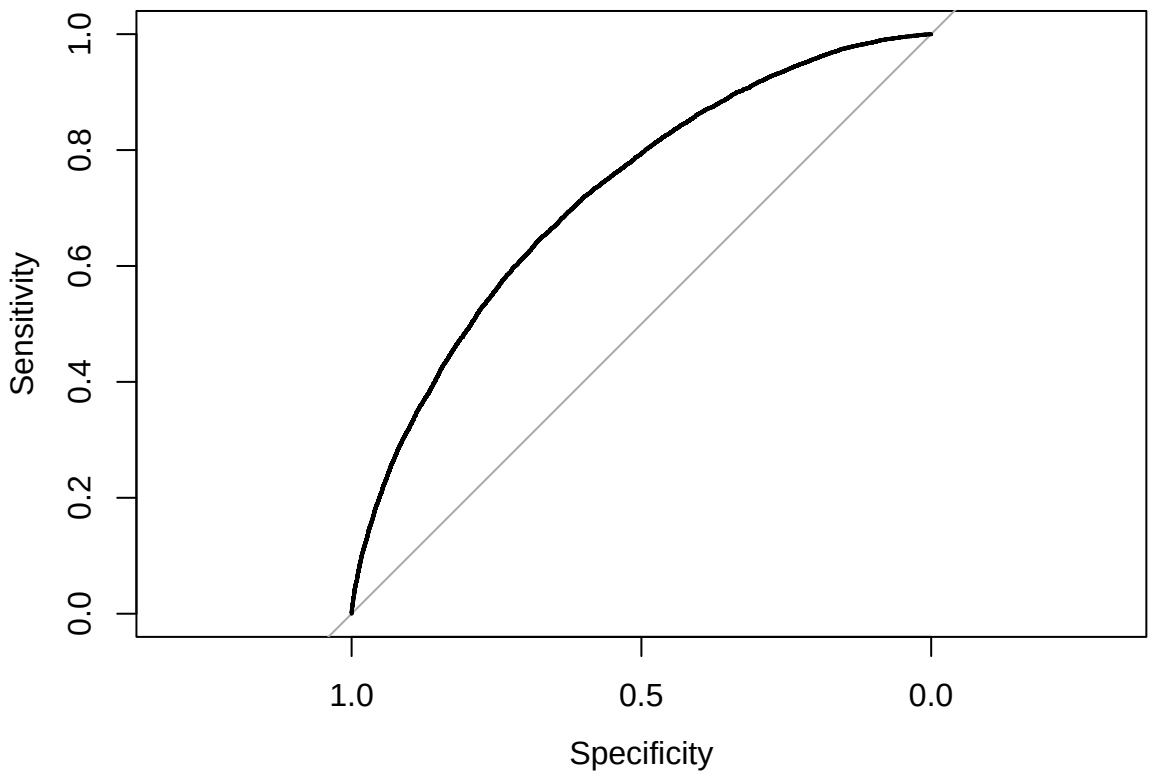
```
## Second stage ensemble ##
sub_ensemb_lc_test$pred_logit <- sub_ensemb_lc_pred_logit1
sub_ensemb_lc_test$pred_ranger <- sub_ensemb_lc_pred_ranger1

# second stage logit regression
m.stack <- glm(default ~ sub_ensemb_lc_pred_logit1 + sub_ensemb_lc_pred_ranger1,
               data = sub_ensemb_lc_test, family = binomial)

# second stage prediction based on logit estimates
p.stack <- predict(m.stack, sub_ensemb_lc_test, type = "response")

# second stage combined ROC
roc.stack <- roc(response = sub_ensemb_lc_test$default, predictor = p.stack)
```

```
## Setting levels: control = no, case = default
## Setting direction: controls < cases
```

```
auc(roc.stack)
```

```
## Area under the curve: 0.7215
```

```
plot(roc.stack)
```



For our ensemble, we combined a separately run Random Forest with a logistic regression model. These would then be blended in a second step. As far as the test and training dataset are concerned, we only

utilized the original "lending club train" dataset since creating holdouts based on this set only would allow us to perform our own testing and validation steps. Though it should be added that this approach came at the expense of the training dataset size.

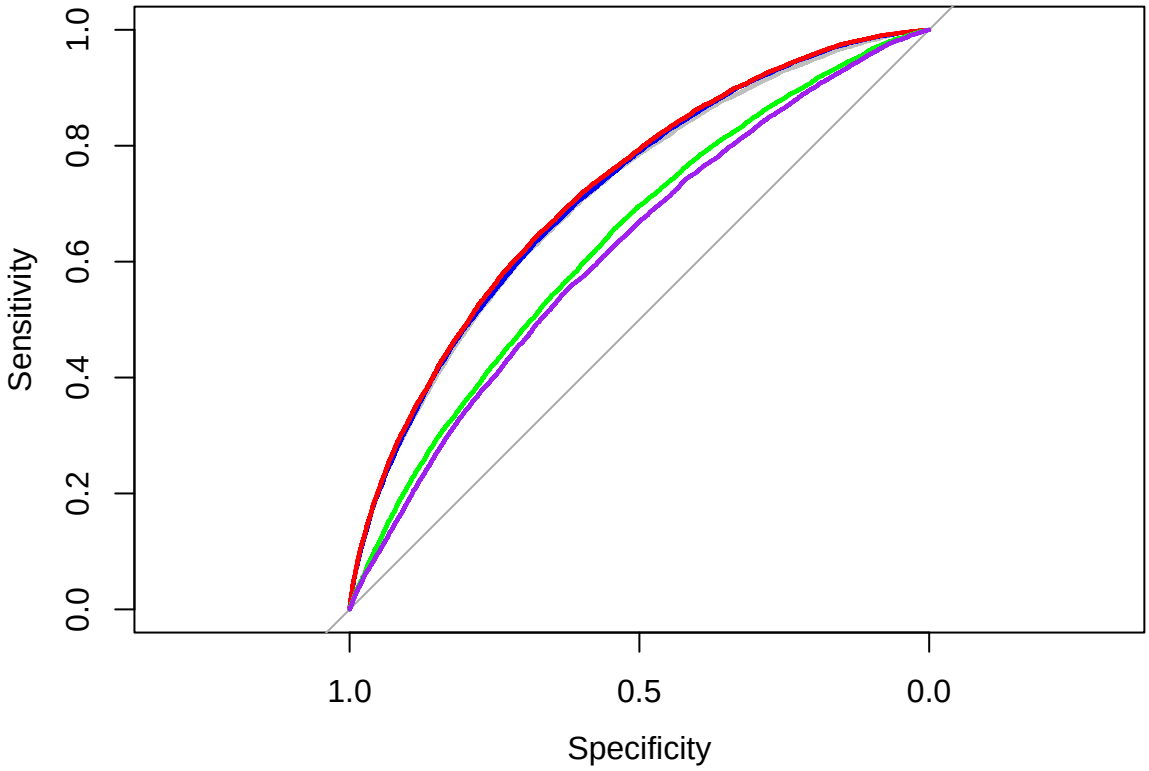
Applying the datasplit function in the rsample package, we ensured the creation of balanced datasets not only regarding the outcome variable but also regarding the other features.

After seeing disappointing results initially, we centered our efforts around the following strategies: modifying the model parameters (e.g., number of folds or node size) and varying the set of features (selective versus all-in). The features turned out to be the most effective route. Even adding features with a high share of missing variables which we had originally dismissed for this reason, resulted in a steady increase in terms of AUC. We ultimately ended up including almost 50 features plus 50 MVIs, respectively. Through this approach, we gradually managed to drive AUC to 0.7215

Despite the fact that this result was obtained with a lot of effort, fine-tuning and training new models over an over, the economic significance should be taken with a grain of salt. We doubt that our model would be ready for production stage of Lending Club even though we are unaware how good their loan decisions perform in practice using other methods. Given a 30% probability to miss the right decision in our model, the value at risk seems too high.

Comparing best Models | Logistic Regression vs Random Forest vs Ensemble

```
## Plot Logistic Regression vs Random Forest vs Ensemble

plot(roc.logit, col = "grey") # weshalb so gut?
plot(roc.ranger, col = "blue", add = TRUE)
plot(roc.stack, col = "red", add = TRUE )
plot(roc_logit, col = "green", add = TRUE) # neu hinzugefügt
plot(roc_logit_2, col = "purple", add = TRUE) # neu hinzugefügt
```



We will conclude by scrutinizing the performance of the individual models in the ensemble compared to the ensemble's overall performance. Clearly, picking an ensemble was one factor contributing to the best model we were able to come up with, as the ensemble blending a Random Forest and a logit performs better than the two components individually.

This becomes evident by taking a closer look at the ROC curves since we decided to use AUC as our performance metric: with the logit yielding 0.713 and the Random Forest yielding 0.718, we can identify a delta between the two curves, though hardly distinguishable visually. The combined curve (red) exhibits the "best" coverage by taking the combined area into account. The two base curves indeed intersect at a few positions as one should expect.

It is also noteworthy that the purple and green lines in the plot represent the two initial pure logistic regression models. These models perform clearly worse compared to the Random Forest and the ensemble, as can be seen by their ROC curves being further away from the top-left corner and closer to the diagonal baseline.

Our team-internal discussions on potential reasons for this strikingly limited difference between a simple logit and a, relatively, more state-of-the-art Random Forest remained uncertain. Learning more about this aspect would be a relevant learning from this final exercise.

Qualitative Description of Model Performance

Our model performance was achieved with an ensemble model, which is a combination of a Random Forest and logistic regression. As described in the previous chapter and shown in the plot, we reached an AUC value of approximately 0.72. In our opinion, this indicates a moderate level of discriminatory power. This means the model is able to differentiate between borrowers with and without default better than random classification.

For Lending Club, however, the performance of our model would not be sufficient for direct implementation and operational use. In a real-world environment, where credit decisions involve significant financial risks, higher prediction accuracy is necessary for a company like Lending Club, as misclassifications can lead to substantial financial consequences.

Nevertheless, the current model could serve Lending Club as a valuable interim result for initial analytical insights. However, the results should be interpreted with caution, since the high error rate of approximately 28–30% represents a considerable risk of misjudgments.

To further improve the model and make it suitable for productive use, we would recommend Lending Club to pursue the following steps:

- Expand feature engineering approaches
- Explore alternative modeling approaches and ensemble combinations, such as neural networks or boosting methods
- Integrate additional external data sources or increase the dataset size

In summary, although the model has been improved throughout the course week with the use of an ensemble, it is still not ready for deployment. Further development work would be necessary to achieve a prediction quality suitable for Lending Club's operational standards. However, the fact that an AUC value above 0.7 could be achieved within a relatively short period of time shows that there is significant potential for improvement with more time and resources.

Personal Learning and Work Distribution | Beat Kessler

As described in the introductory chapter, the work was carried out in collaboration between Beat Kessler and Kilian Dinkelaker. The reason for this collaboration was the idea that better results can generally be achieved in a team and ultimately a higher model score can be reached. Additionally, the team members were able to benefit from each other and contribute different added value due to their individual academic and professional experience.

Within the scope of this report, Beat Kessler was primarily responsible for the Exploratory Data Analysis (EDA), all logistic regression models, mean imputation, and the chapter on Missing Value Indicators (MVI). His responsibilities included coding and implementation, as well as the technical development of the models. Furthermore, Beat worked on the chapters and texts for the Introduction, Data Preparation and Feature Engineering, and the Qualitative Description of Model Performance. He was also generally responsible for setting up and formatting the R Markdown document and resolving technical code issues to enable the generation of the PDF document.

Beat also invested additional time in handling the formatting of the R Markdown document and in merging the various models and structures. Additionally, Beat was responsible for code improvements, which generally contributed to better readability and understanding.

The collaboration took place through close cooperation during and after lecture hours, as well as mutual code reviews, ensuring a common understanding. Through this project, the team members were able to strengthen not only their technical skills but also their communication and project management competencies.