Predicting Bad Lending Club Loans for Fixed Loan Grades with Multiple Different Models

Bill Anderson (william.david.anderson@gmail.com)

17 May 2016

Introduction and Executive Summary

This document presents an analysis of lending club data for loans issued between June 2007 and December 2011, with the goal of predicting which loans will go "bad" (i.e., the borrower misses a payment or defaults). This analysis is done with the loan grade held constant (e.g., analysis for all A loans, analysis for all B loans, etc.), which can be useful; for example, if we could identify all the grade D loans that would not go bad, we would have the best of both worlds: high interest rates, but no risk of loss from default. For this study, loans with grade A, B, C, and D were considered (not enough data for grade E loans). Also, this study used five different model types: logistic regression, random forest, gradient boost, support vector machines, and neural networks. The results from the different models were similar, although the some models performed slightly better than others.

For the grade C and D loans (the ones with the most defaults), we can correctly identify over 65% of the loans that will go bad. Also for these same loan grades, all four of the models identified the same two predictors that were most important in predicting which loans will go bad: FICO score and the number of credit inquiries in the past six months.

Details on these and other results are shown below.

Data Ingest and Initialization Steps

```
# read in the lending club data
setwd("/Users/andersnb/lending-club/my-analysis")
loans <- read.csv("../data/LoanStats3a_securev1.csv")
str(loans)
```

```
## 'data.frame':    42536 obs. of  115 variables:
##  $ id                         : Factor w/ 42536 levels "1000007","1000030",..: 4388 4387 4386 4385
##  $ member_id                  : int  1296599 1314167 1313524 1277178 1311748 1311441 1304742 12886
##  $ loan_amnt                  : int  5000 2500 2400 10000 3000 5000 7000 3000 5600 5375 ...
##  $ funded_amnt                : int  5000 2500 2400 10000 3000 5000 7000 3000 5600 5375 ...
##  $ funded_amnt_inv            : num  4975 2500 2400 10000 3000 ...
##  $ term                       : Factor w/ 3 levels ""," 36 months",..: 2 3 2 2 3 2 3 2 3 3 ...
##  $ int_rate                   : Factor w/ 395 levels "","  5.42%","  5.79%",..: 80 223 241 162 13
##  $ installment                : num  162.9 59.8 84.3 339.3 67.8 ...
##  $ grade                      : Factor w/ 8 levels "","A","B","C",..: 3 4 4 4 3 2 4 6 7 3 ...
##  $ sub_grade                  : Factor w/ 36 levels "","A1","A2","A3",..: 8 15 16 12 11 5 16 22 28
##  $ emp_title                  : Factor w/ 30661 levels "","  old palm inc",..: 1 22922 1 791 28234
##  $ emp_length                 : Factor w/ 13 levels "","< 1 year",..: 4 2 4 4 3 6 11 12 7 2 ...
##  $ home_ownership             : Factor w/ 6 levels "","MORTGAGE",..: 6 6 6 6 6 6 6 6 5 6 ...
##  $ annual_inc                 : num  24000 30000 12252 49200 80000 ...
##  $ verification_status        : Factor w/ 4 levels "","Not Verified",..: 4 3 2 3 3 3 2 3 3 4 ...
```

```
##  $ issue_d                    : Factor w/ 56 levels "","Apr-2008",..: 15 15 15 15 15 15 15 15 15 1
##  $ loan_status                : Factor w/ 10 levels "","Charged Off",..: 7 2 7 7 3 7 3 7 2 2 ...
##  $ pymnt_plan                 : Factor w/ 3 levels "","n","y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ url                        : Factor w/ 42536 levels "","https://www.lendingclub.com/browse/loan
##  $ desc                       : Factor w/ 28965 levels "","\t Member# 809768, loan description. (
##  $ purpose                    : Factor w/ 15 levels "","car","credit_card",..: 3 2 13 11 11 15 4 1
##  $ title                      : Factor w/ 21267 levels "","\tdebt_consolidation",..: 3687 1869 172
##  $ zip_code                   : Factor w/ 838 levels "","007xx","010xx",..: 728 282 514 765 814 72
##  $ addr_state                 : Factor w/ 51 levels "","AK","AL","AR",..: 5 12 16 6 38 5 29 6 6 4
##  $ dti                        : num  27.65 1 8.72 20 17.94 ...
##  $ delinq_2yrs                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ earliest_cr_line           : Factor w/ 531 levels "","Apr-1964",..: 194 36 431 163 205 434 256
##  $ fico_range_low             : int  735 740 735 690 695 730 690 660 675 725 ...
##  $ fico_range_high            : int  739 744 739 694 699 734 694 664 679 729 ...
##  $ inq_last_6mths             : int  1 5 2 1 0 3 1 2 2 0 ...
##  $ mths_since_last_delinq     : int  NA NA NA 35 38 NA NA NA NA NA ...
##  $ mths_since_last_record     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ open_acc                   : int  3 3 2 10 15 9 7 4 11 2 ...
##  $ pub_rec                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ revol_bal                  : int  13648 1687 2956 5598 27783 7963 17726 8221 5210 9279 ...
##  $ revol_util                 : Factor w/ 1120 levels "","0.01%","0.03%",..: 943 1012 1105 204 59
##  $ total_acc                  : int  9 4 10 37 38 12 11 4 13 3 ...
##  $ initial_list_status        : Factor w/ 2 levels "","f": 2 2 2 2 2 2 2 2 2 2 ...
##  $ out_prncp                  : num  0 0 0 0 707 ...
##  $ out_prncp_inv              : num  0 0 0 0 707 ...
##  $ total_pymnt                : num  5863 1009 3006 12232 3310 ...
##  $ total_pymnt_inv            : num  5834 1009 3006 12232 3310 ...
##  $ total_rec_prncp            : num  5000 456 2400 10000 2293 ...
##  $ total_rec_int              : num  863 435 606 2215 1017 ...
##  $ total_rec_late_fee         : num  0 0 0 17 0 ...
##  $ recoveries                 : num  0 117 0 0 0 ...
##  $ collection_recovery_fee    : num  0 1.11 0 0 0 0 0 0 2.09 2.52 ...
##  $ last_pymnt_d               : Factor w/ 100 levels "","Apr-2008",..: 43 7 59 43 35 43 35 43 6 8
##  $ last_pymnt_amnt            : num  171.6 119.7 649.9 357.5 67.8 ...
##  $ next_pymnt_d               : Factor w/ 102 levels "","Apr-2008",..: 1 1 1 1 70 1 10 1 1 1 ...
##  $ last_credit_pull_d         : Factor w/ 105 levels "","Apr-2009",..: 35 103 35 43 35 44 35 25 1
##  $ last_fico_range_high       : int  719 534 679 579 674 679 644 689 499 499 ...
##  $ last_fico_range_low        : int  715 530 675 575 670 675 640 685 0 0 ...
##  $ collections_12_mths_ex_med : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mths_since_last_major_derog : logi  NA NA NA NA NA NA ...
##  $ policy_code                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ application_type           : Factor w/ 2 levels "","INDIVIDUAL": 2 2 2 2 2 2 2 2 2 2 ...
##  $ annual_inc_joint           : logi  NA NA NA NA NA NA ...
##  $ dti_joint                  : logi  NA NA NA NA NA NA ...
##  $ verification_status_joint  : logi  NA NA NA NA NA NA ...
##  $ acc_now_delinq             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ tot_coll_amt               : logi  NA NA NA NA NA NA ...
##  $ tot_cur_bal                : logi  NA NA NA NA NA NA ...
##  $ open_acc_6m                : logi  NA NA NA NA NA NA ...
##  $ open_il_6m                 : logi  NA NA NA NA NA NA ...
##  $ open_il_12m                : logi  NA NA NA NA NA NA ...
##  $ open_il_24m                : logi  NA NA NA NA NA NA ...
##  $ mths_since_rcnt_il         : logi  NA NA NA NA NA NA ...
##  $ total_bal_il               : logi  NA NA NA NA NA NA ...
```

```
##  $ il_util                       : logi  NA NA NA NA NA NA ...
##  $ open_rv_12m                   : logi  NA NA NA NA NA NA ...
##  $ open_rv_24m                   : logi  NA NA NA NA NA NA ...
##  $ max_bal_bc                    : logi  NA NA NA NA NA NA ...
##  $ all_util                      : logi  NA NA NA NA NA NA ...
##  $ total_rev_hi_lim              : logi  NA NA NA NA NA NA ...
##  $ inq_fi                        : logi  NA NA NA NA NA NA ...
##  $ total_cu_tl                   : logi  NA NA NA NA NA NA ...
##  $ inq_last_12m                  : logi  NA NA NA NA NA NA ...
##  $ acc_open_past_24mths          : logi  NA NA NA NA NA NA ...
##  $ avg_cur_bal                   : logi  NA NA NA NA NA NA ...
##  $ bc_open_to_buy                : logi  NA NA NA NA NA NA ...
##  $ bc_util                       : logi  NA NA NA NA NA NA ...
##  $ chargeoff_within_12_mths      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ delinq_amnt                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mo_sin_old_il_acct            : logi  NA NA NA NA NA NA ...
##  $ mo_sin_old_rev_tl_op          : logi  NA NA NA NA NA NA ...
##  $ mo_sin_rcnt_rev_tl_op         : logi  NA NA NA NA NA NA ...
##  $ mo_sin_rcnt_tl                : logi  NA NA NA NA NA NA ...
##  $ mort_acc                      : logi  NA NA NA NA NA NA ...
##  $ mths_since_recent_bc          : logi  NA NA NA NA NA NA ...
##  $ mths_since_recent_bc_dlq      : logi  NA NA NA NA NA NA ...
##  $ mths_since_recent_inq         : logi  NA NA NA NA NA NA ...
##  $ mths_since_recent_revol_delinq: logi  NA NA NA NA NA NA ...
##  $ num_accts_ever_120_pd         : logi  NA NA NA NA NA NA ...
##  $ num_actv_bc_tl                : logi  NA NA NA NA NA NA ...
##  $ num_actv_rev_tl               : logi  NA NA NA NA NA NA ...
##  $ num_bc_sats                   : logi  NA NA NA NA NA NA ...
##  $ num_bc_tl                     : logi  NA NA NA NA NA NA ...
##  $ num_il_tl                     : logi  NA NA NA NA NA NA ...
##   [list output truncated]
```

```r
# initialize random number generator
set.seed(1)
```

Data Cleaning

In this section, we convert data types, get rid of unneeded data, etc.

```r
#
# Loans in the dataset were issued at different times and have terms of 3 or 5 years.
# We want all loans to have the same chance to reach maturity or the results could be
# misleading. Consider an extreme case where a loan is issued the month before the end
# of when data is collected. The loan is less likely to be in default after just one
# month than if it's been outstanding for 3 (or 5) years and such loans could result in
# misleading interpretations.  Thus, since this dataset ends at Feb 2016, we should only
# consider loans that were issued 5 years or more ago, or that were issued Feb 2011 or
# earlier.
#
loans <- filter(loans, issue_d != "")
loans$issue_d <- factor(loans$issue_d)
loans$issue_d <- parse_date_time(paste("01-", loans$issue_d), "%d-%b-%Y")
```

```r
loans <- filter(loans, issue_d  <= "2011-02-01")

#
# convert to a date type
#
loans <- filter(loans, last_pymnt_d != "")
loans$last_pymnt_d <- factor(loans$last_pymnt_d)
loans$last_pymnt_d <- parse_date_time(paste("01-", loans$last_pymnt_d), "%d-%b-%Y")

#
# convert to a date type
#
loans <- filter(loans, earliest_cr_line != "")
loans$earliest_cr_line <- factor(loans$earliest_cr_line)
loans$earliest_cr_line <- parse_date_time(paste("01-", loans$earliest_cr_line), "%d-%b-%Y")

#
# convert to a date type
#
loans <- filter(loans, last_credit_pull_d != "")
loans$last_credit_pull_d <- factor(loans$last_credit_pull_d)
loans$last_credit_pull_d <- parse_date_time(paste("01-", loans$last_credit_pull_d), "%d-%b-%Y")

# get rid of empty factor
loans <- filter(loans, term != "")
loans$term <- factor(loans$term)

# convert interest rate from string to float
loans$int_rate <- gsub("%", "", loans$int_rate)
loans$int_rate <- gsub(" ", "", loans$int_rate)
loans$int_rate <- as.numeric(loans$int_rate)

# get rid of empty factor
loans <- filter(loans, grade != "")
loans$grade <- factor(loans$grade)

# get rid of empty factor
loans <- filter(loans, sub_grade != "")
loans$sub_grade <- factor(loans$sub_grade)

# get rid of empty factor
loans <- filter(loans, emp_length != "")
loans$emp_length <- factor(loans$emp_length)

# get rid of empty factor
loans <- filter(loans, home_ownership != "")
loans$home_ownership <- factor(loans$home_ownership)

# get rid of empty factor
loans <- filter(loans, verification_status != "")
loans$verification_status <- factor(loans$verification_status)

# get rid of empty factor
```

```r
loans <- filter(loans, pymnt_plan != "")
loans$pymnt_plan <- factor(loans$pymnt_plan)

# create a variable that's true if the desc is empty, else false
loans <- mutate(loans, desc_empty = as.factor(ifelse(desc == "", 1, 0)))

# get rid of empty factor
loans <- filter(loans, purpose != "")
loans$purpose <- factor(loans$purpose)

# get rid of empty factor
loans <- filter(loans, zip_code != "")
loans$zip_code <- factor(loans$zip_code)

# get rid of empty factor
loans <- filter(loans, addr_state != "")
loans$addr_state <- factor(loans$addr_state)

# convert revolv_util from a factor to a numeric variable
loans$revol_util <- as.numeric(gsub("%", "", loans$revol_util))

# get rid of empty factor
loans <- filter(loans, initial_list_status != "")
loans$initial_list_status <- factor(loans$initial_list_status)

#
# the following columns are deemed not useful (for the following reasons) so we exclude them:
# mths_since_last_major_derog  (all NAs)
# annual_inc_joint             (all NAs)
# dti_joint                    (all NAs)
# verification_status_joint    (all NAs)
# tot_coll_amt                 (all NAs)
# tot_cur_bal                  (all NAs)
# open_acc_6m                  (all NAs)
# open_il_6m                   (all NAs)
# open_il_12m                  (all NAs)
# open_il_24m                  (all NAs)
# mths_since_rcnt_il           (all NAs)
# total_bal_il                 (all NAs)
# il_util                      (all NAs)
# open_rv_12m                  (all NAs)
# open_rv_24m                  (all NAs)
# max_bal_bc                   (all NAs)
# all_util                     (all NAs)
# total_rev_hi_lim             (all NAs)
# inq_fi                       (all NAs)
# total_cu_tl                  (all NAs)
# inq_last_12m                 (all NAs)
# acc_open_past_24mths         (all NAs)
# avg_cur_bal                  (all NAs)
# bc_open_to_buy               (all NAs)
# bc_util                      (all NAs)
# mo_sin_old_il_acct           (all NAs)
```

```
# mo_sin_old_rev_tl_op         (all NAs)
# mo_sin_rcnt_rev_tl_op        (all NAs)
# mo_sin_rcnt_tl               (all NAs)
# mort_acc                     (all NAs)
# mths_since_recent_bc         (all NAs)
# mths_since_recent_bc_dlq     (all NAs)
# mths_since_recent_inq        (all NAs)
# mths_since_recent_revol_delinq (all NAs)
# num_accts_ever_120_pd        (all NAs)
# num_actv_bc_tl               (all NAs)
# num_actv_rev_tl              (all NAs)
# num_bc_sats                  (all NAs)
# num_bc_tl                    (all NAs)
# num_il_tl                    (all NAs)
# num_op_rev_tl                (all NAs)
# num_rev_accts                (all NAs)
# num_rev_tl_bal_gt_0          (all NAs)
# num_sats                     (all NAs)
# num_tl_120dpd_2m             (all NAs)
# num_tl_30dpd                 (all NAs)
# num_tl_90g_dpd_24m           (all NAs)
# num_tl_op_past_12m           (all NAs)
# pct_tl_nvr_dlq               (all NAs)
# percent_bc_gt_75             (all NAs)
# tot_hi_cred_lim              (all NAs)
# total_bal_ex_mort            (all NAs)
# total_bc_limit               (all NAs)
# total_il_high_credit_limit   (all NAs)
# next_pymnt_d                 (doesn't seem relevant to loan status and contained a lot of missing dat
# mths_since_last_delinq       (a very large number of NAs)
# mths_since_last_record       (a very large number of NAs)
# id                           (not relevant to loan status)
# member_id                    (not relevant to loan status)
# url                          (url for the loan details; not relevant to loan status)
# desc                         (it's possible the information contained in the desc. could be useful; f
# title                        (it's possible the information contained in the title could be useful, f
# emp_title                    (it's possible the information contained in emp_title could be useful; f
#
```

```
loans <- subset(loans, select = -c(mths_since_last_major_derog,
    annual_inc_joint, dti_joint, verification_status_joint, tot_coll_amt,
    tot_cur_bal, open_acc_6m, open_il_6m, open_il_12m, open_il_24m,
    mths_since_rcnt_il, total_bal_il, il_util, open_rv_12m, open_rv_24m,
    max_bal_bc, all_util, total_rev_hi_lim, inq_fi, total_cu_tl,
    inq_last_12m, acc_open_past_24mths, avg_cur_bal, bc_open_to_buy,
    bc_util, mo_sin_old_il_acct, mo_sin_old_rev_tl_op, mo_sin_rcnt_rev_tl_op,
    mo_sin_rcnt_tl, mort_acc, mths_since_recent_bc, mths_since_recent_bc_dlq,
    mths_since_recent_inq, mths_since_recent_revol_delinq, num_accts_ever_120_pd,
    num_actv_bc_tl, num_actv_rev_tl, num_bc_sats, num_bc_tl,
    num_il_tl, num_op_rev_tl, num_rev_accts, num_rev_tl_bal_gt_0,
    num_sats, num_tl_120dpd_2m, num_tl_30dpd, num_tl_90g_dpd_24m,
    num_tl_op_past_12m, pct_tl_nvr_dlq, percent_bc_gt_75, tot_hi_cred_lim,
    total_bal_ex_mort, total_bc_limit, total_il_high_credit_limit,
```

```
    next_pymnt_d, mths_since_last_delinq, mths_since_last_record,
    id, member_id, url, desc, title, emp_title))

# create binary status variable; note: I define as 'bad' any
# loan that is not current or not fully paid
loans <- mutate(loans, status = factor(ifelse(loan_status ==
    "Current" | loan_status == "Fully Paid", "good", "bad"),
    levels = c("good", "bad")))
```

Exploratory Plots

In this section, we create exploratory plots and/or tables for each variable to help determine which variables are likely to have an effect on the loan status and, thus, should be used in the subsequent models. Note: to generate the various plots, set the explPlots and/or the collScatterPlots variables at the beginning of the R markdown document to TRUE.

```
# create exploratory plots
createExplPlots <- function(dft) {
    for (i in 1:ncol(dft)) {
        varname = names(dft)[i]
        print(paste(varname, ":"))

        if (varname == "annual_inc") {
            # annual income requires a limit of 200000 since there are
            # some outliers that make the plots hard to understand or
            # visualize
            p <- ggplot(aes_string(x = varname, group = "status",
                colour = "status"), data = dft)
            p <- p + geom_density() + xlab(varname)
            print(p)

            p <- ggplot(dft, aes_string(x = "status", y = varname)) +
                geom_boxplot() + ylab(varname) + ylim(0, 2e+05)
            print(p)

        } else if (varname == "delinq_2yrs") {
            # delinq_2yrs requires a limit of 5 since there are some
            # outliers that make the plots hard to understand
            p <- ggplot(aes_string(x = varname, group = "status",
                colour = "status"), data = dft)
            p <- p + geom_density() + xlab(varname)
            print(p)

            p <- ggplot(dft, aes_string(x = "status", y = varname)) +
                geom_boxplot() + ylab(varname) + ylim(0, 5)
            print(p)

        } else {
            # create plots that don't require special limits
            p <- ggplot(aes_string(x = varname, group = "status",
                colour = "status"), data = dft)
            p <- p + geom_density() + xlab(varname)
```

```r
            print(p)

            if (class(dft[[i]]) == "numeric" || class(dft[[i]]) ==
                "integer") {
                p <- ggplot(dft, aes_string(x = "status", y = varname)) +
                  geom_boxplot() + ylab(names(dft)[i])
                print(p)

            } else {
                print(table(dft[[i]], dft$status))
                print(prop.table(table(dft[[i]], dft$status),
                  1))

            }
        }
        cat("\n")
    }
}
# subset data by loan grade
a_loans <- loans[loans$grade == "A", ]
b_loans <- loans[loans$grade == "B", ]
c_loans <- loans[loans$grade == "C", ]
d_loans <- loans[loans$grade == "D", ]

# create exploratory plots by loan grade
if (explPlots == TRUE) {
    createExplPlots(a_loans)
    createExplPlots(b_loans)
    createExplPlots(c_loans)
    createExplPlots(d_loans)
}


# select predictors that have an effect on response and get
# rid of rows with NAs
a_loans <- select(a_loans, c(status, term, verification_status,
    purpose, fico_range_low, fico_range_high, inq_last_6mths,
    revol_util, last_fico_range_low, last_fico_range_high, desc_empty,
    dti))
b_loans <- select(b_loans, c(status, term, verification_status,
    purpose, fico_range_low, fico_range_high, inq_last_6mths,
    revol_util, last_fico_range_low, last_fico_range_high, desc_empty,
    dti))
c_loans <- select(c_loans, c(status, term, verification_status,
    purpose, fico_range_low, fico_range_high, inq_last_6mths,
    revol_util, last_fico_range_low, last_fico_range_high, desc_empty,
    dti))
d_loans <- select(d_loans, c(status, term, verification_status,
    purpose, fico_range_low, fico_range_high, inq_last_6mths,
    revol_util, last_fico_range_low, last_fico_range_high, desc_empty,
    dti))

a_loans <- na.omit(a_loans)
```

```r
b_loans <- na.omit(b_loans)
c_loans <- na.omit(c_loans)
d_loans <- na.omit(d_loans)

# now check for collinearity
checkForColl <- function(l) {
    pairs(~term + verification_status + purpose + fico_range_low +
        fico_range_high + inq_last_6mths + revol_util + last_fico_range_low +
        last_fico_range_high + desc_empty + dti, data = l)
}

if (collScatterPlots == TRUE) {
    checkForColl(a_loans)
    checkForColl(b_loans)
    checkForColl(c_loans)
    checkForColl(d_loans)
}


# the collinearity scatterplots suggest that there's is a
# correlation between fico_range_high/fico_range_low and
# between last_fico_range_low/last_fico_range high;
# therefore, I won't use fico_range_low or
# last_fico_range_log in the models to avoid collinearity
a_loans <- select(a_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, last_fico_range_high,
    desc_empty, dti))
b_loans <- select(b_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, last_fico_range_high,
    desc_empty, dti))
c_loans <- select(c_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, last_fico_range_high,
    desc_empty, dti))
d_loans <- select(d_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, last_fico_range_high,
    desc_empty, dti))
```

Model Construction and Execution

The next section builds several model types (logistic, random forest, gradient boost, support vector machine (SVM), and neural network), makes predictions and identifies the important variables in each model.

```r
createDataForInput <- function(dft) {
    # partition the data into a training portion and test portion
    inTraining <- createDataPartition(dft$status, p = 0.75, list = FALSE)
    dft_orig <- dft
    dft_train <- dft_orig[inTraining, ]
    dft_test <- dft_orig[-inTraining, ]

    return(list(dft_train = dft_train, dft_test = dft_test))
}
```

```r
# function to create logistic regression model
logRegModel <- function(dft_train, dft_test) {
    modLogReg <- train(status ~ ., data = dft_train, method = "glm")
    print(modLogReg)
    print(summary(modLogReg))
    print(varImp(modLogReg))

    testPred <- predict(modLogReg, dft_test)
    print(confusionMatrix(testPred, dft_test$status, positive = "bad"))

    testProbs <- predict(modLogReg, dft_test, type = "prob")
    rocObj <- roc(dft_test$status, testProbs[, "bad"])
    plot(rocObj, type = "S", print.thres = 0.5)

}

# function to create random forest model
rfModel <- function(dft_train, dft_test) {
    modRandFor <- train(status ~ ., data = dft_train, method = "rf")
    print(modRandFor)
    print(summary(modRandFor))
    print(varImp(modRandFor))

    testPred <- predict(modRandFor, dft_test)
    print(confusionMatrix(testPred, dft_test$status, positive = "bad"))

    testProbs <- predict(modRandFor, dft_test, type = "prob")
    rocObj <- roc(dft_test$status, testProbs[, "bad"])
    plot(rocObj, type = "S", print.thres = 0.5)

}

# function to create a gradient boost model
gbModel <- function(dft_train, dft_test) {
    modGradBoost <- train(status ~ ., data = dft_train, method = "gbm", verbose = FALSE)
    print(modGradBoost)
    print(summary(modGradBoost))
    print(varImp(modGradBoost))

    testPred <- predict(modGradBoost, dft_test)
    print(confusionMatrix(testPred, dft_test$status, positive = "bad"))

    testProbs <- predict(modGradBoost, dft_test, type = "prob")
    rocObj <- roc(dft_test$status, testProbs[, "bad"])
    plot(rocObj, type = "S", print.thres = 0.5)

}

# function to create SVM Gaussian kernel model note: I use the 'cv' method
# for resampling because the default boot method results in a lot of warning
# messages about duplicate row names and the 'cv' method yields results that
# are as accurate as the 'boot' method
svmModel <- function(dft_train, dft_test) {
```

```
    modSvm <- train(status ~ ., data = dft_train, method = "svmRadial", preProc = c("center",
        "scale"), trControl = trainControl(classProbs = TRUE, method = "cv"))

    print(modSvm)
    print(summary(modSvm))
    print(varImp(modSvm))

    testPred <- predict(modSvm, dft_test)
    print(confusionMatrix(testPred, dft_test$status, positive = "bad"))

    testProbs <- predict(modSvm, dft_test, type = "prob")
    rocObj <- roc(dft_test$status, testProbs[, "bad"])
    plot(rocObj, type = "S", print.thres = 0.5)

}

# function to create neural network model note: I use one hidden layer, but,
# via the tuneLength paramter to train, specify that it try 7 different
# parameter values (higher than for the other model types)
nnetModel <- function(dft_train, dft_test) {
    modNnet <- train(status ~ ., data = dft_train, method = "nnet", tuneLength = 7,
        trace = FALSE)

    print(modNnet)
    print(summary(modNnet))
    print(varImp(modNnet))

    testPred <- predict(modNnet, dft_test)
    print(confusionMatrix(testPred, dft_test$status, positive = "bad"))

    testProbs <- predict(modNnet, dft_test, type = "prob")
    rocObj <- roc(dft_test$status, testProbs[, "bad"])
    plot(rocObj, type = "S", print.thres = 0.5)

}
```

Results for Grade A Loans

Only a small percentage (~7%) of the Grade A loans go bad, making it somewhat challenging to identify those loans, but, since there are so few, it's also less important. The results show that the five models had sensitivities (i.e., ability to correctly predict the bad loans) ranging from 0% to 26%. This predictive ability is based on a 50% probability classification cutoff. As the ROC curves show, it's possible to predict the bad loans with a higher probability, but, of course, with a higher false positive rate. The FICO range and the number of inquiries in the past 6 months were important predictors.

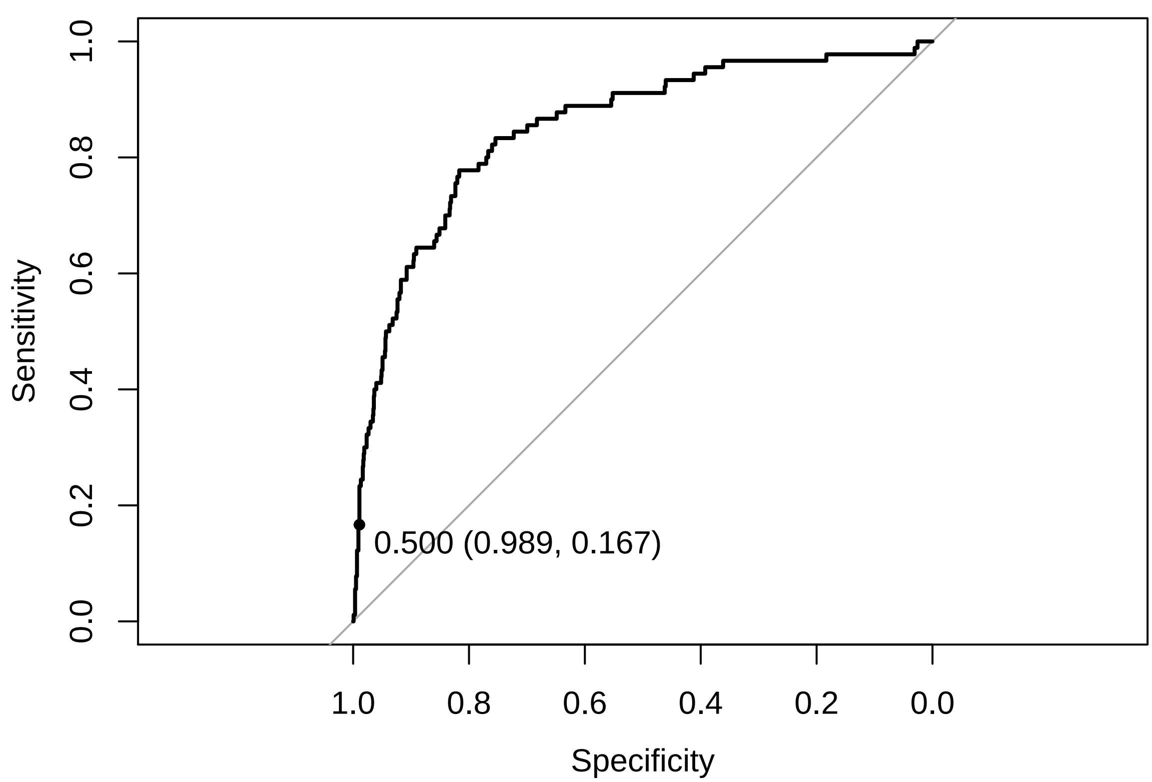
Logistic Regression Model

```
## Generalized Linear Model
##
## 3879 samples
##    9 predictor
##    2 classes: 'good', 'bad'
```

```
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3879, 3879, 3879, 3879, 3879, 3879, ...
##
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.9335321  0.2797235  0.003836129  0.03338435
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1505  -0.3149  -0.2099  -0.1442   3.2303
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                          13.7986678  2.9075901   4.746
## `term 60 months`                      0.6508329  0.2844911   2.288
## `verification_statusSource Verified` -0.3583547  0.2000624  -1.791
## verification_statusVerified          -0.1064885  0.1794718  -0.593
## purposecredit_card                   -0.5959759  0.3486052  -1.710
## purposedebt_consolidation            -0.3283520  0.2963099  -1.108
## purposeeducational                    0.4707276  0.4926210   0.956
## purposehome_improvement              -0.4675126  0.3742106  -1.249
## purposehouse                         -0.3180473  0.7039476  -0.452
## purposemajor_purchase                -0.7079184  0.3973222  -1.782
## purposemedical                       -0.1293572  0.5060129  -0.256
## purposemoving                        -0.9334389  0.5546842  -1.683
## purposeother                         -0.5040392  0.3337273  -1.510
## purposerenewable_energy               0.4920301  1.3039656   0.377
## purposesmall_business                -0.0894251  0.4298221  -0.208
## purposevacation                       0.1535385  0.5752085   0.267
## purposewedding                       -1.0387206  0.6924558  -1.500
## fico_range_high                      -0.0067497  0.0038009  -1.776
## inq_last_6mths                        0.2858123  0.0565287   5.056
## revol_util                           -0.0011562  0.0037267  -0.310
## last_fico_range_high                 -0.0162113  0.0008758 -18.511
## desc_empty1                          -0.1684563  0.1697134  -0.993
## dti                                   0.0054574  0.0114910   0.475
##                                      Pr(>|z|)
## (Intercept)                          2.08e-06 ***
## `term 60 months`                       0.0222 *
## `verification_statusSource Verified`   0.0733 .
## verification_statusVerified            0.5530
## purposecredit_card                     0.0873 .
## purposedebt_consolidation              0.2678
## purposeeducational                     0.3393
## purposehome_improvement                0.2115
```

```
## purposehouse                          0.6514
## purposemajor_purchase                 0.0748 .
## purposemedical                        0.7982
## purposemoving                         0.0924 .
## purposeother                          0.1310
## purposerenewable_energy               0.7059
## purposesmall_business                 0.8352
## purposevacation                       0.7895
## purposewedding                        0.1336
## fico_range_high                       0.0758 .
## inq_last_6mths                      4.28e-07 ***
## revol_util                            0.7564
## last_fico_range_high                 < 2e-16 ***
## desc_empty1                           0.3209
## dti                                   0.6348
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1975.3  on 3878  degrees of freedom
## Residual deviance: 1429.0  on 3856  degrees of freedom
## AIC: 1475
##
## Number of Fisher Scoring iterations: 6
##
## glm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                     Overall
## last_fico_range_high               100.0000
## inq_last_6mths                      26.4873
## `term 60 months`                    11.3623
## `verification_statusSource Verified`   8.6497
## purposemajor_purchase                8.5978
## fico_range_high                      8.5655
## purposecredit_card                   8.2038
## purposemoving                        8.0575
## purposeother                         7.1151
## purposewedding                       7.0589
## purposehome_improvement              5.6891
## purposedebt_consolidation            4.9177
## desc_empty1                          4.2864
## purposeeducational                   4.0840
## verification_statusVerified          2.1051
## dti                                  1.4581
## purposehouse                         1.3318
## purposerenewable_energy              0.9249
## revol_util                           0.5584
## purposevacation                      0.3217
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction good  bad
##       good 1188   75
##       bad    13   15
##
##                Accuracy : 0.9318
##                  95% CI : (0.9167, 0.945)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.4409
##
##                   Kappa : 0.2287
##  Mcnemar's Test P-Value : 7.893e-11
##
##             Sensitivity : 0.16667
##             Specificity : 0.98918
##          Pos Pred Value : 0.53571
##          Neg Pred Value : 0.94062
##              Prevalence : 0.06971
##          Detection Rate : 0.01162
##    Detection Prevalence : 0.02169
##       Balanced Accuracy : 0.57792
##
##        'Positive' Class : bad
##
```



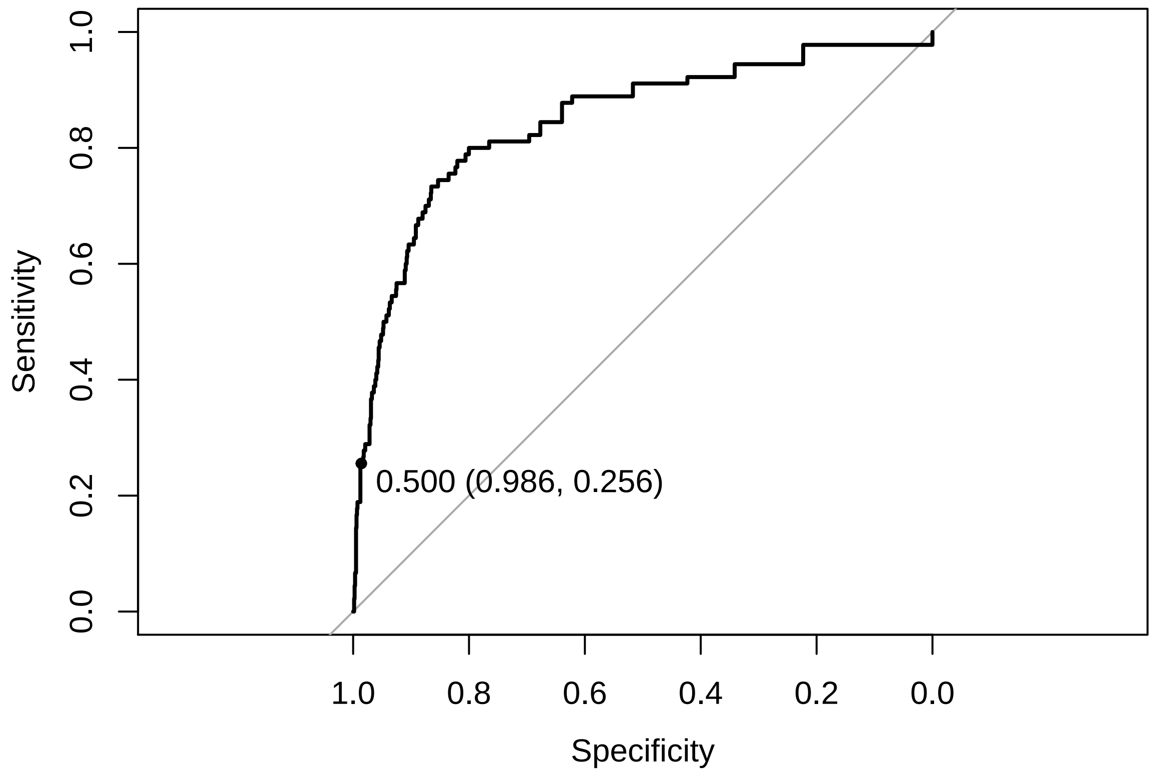

```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
```

```
## Area under the curve: 0.8509
```

Random Forest Model

```
## Random Forest
##
## 3879 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3879, 3879, 3879, 3879, 3879, 3879, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa        Accuracy SD  Kappa SD
##    2    0.9286334  0.002294058  0.005948366  0.006716373
##   12    0.9294704  0.320081829  0.005578916  0.036346900
##   22    0.9257695  0.315731616  0.005629033  0.025405788
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 12.
##                 Length Class      Mode
## call               4   -none-     call
## type               1   -none-     character
## predicted       3879   factor     numeric
## err.rate        1500   -none-     numeric
## confusion          6   -none-     numeric
## votes           7758   matrix     numeric
## oob.times       3879   -none-     numeric
## classes            2   -none-     character
## importance        22   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               3879   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            22   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
## rf variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                    Overall
## last_fico_range_high              100.0000
```

```
## dti                                 48.6456
## revol_util                          45.6345
## fico_range_high                     33.4729
## inq_last_6mths                      19.1502
## purposedebt_consolidation            5.4778
## verification_statusVerified          4.9122
## purposeother                         4.2978
## verification_statusSource Verified   3.9842
## desc_empty1                          3.9830
## purposecredit_card                   3.5645
## term 60 months                       3.2203
## purposehome_improvement              2.5997
## purposemajor_purchase                2.3445
## purposesmall_business                1.5694
## purposemedical                       1.4680
## purposeeducational                   1.2634
## purposevacation                      1.0744
## purposemoving                        0.8869
## purposehouse                         0.6622
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1185   67
##       bad    16   23
##
##                Accuracy : 0.9357
##                  95% CI : (0.9209, 0.9485)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.2412
##
##                   Kappa : 0.3283
##  Mcnemar's Test P-Value : 4.06e-08
##
##             Sensitivity : 0.25556
##             Specificity : 0.98668
##          Pos Pred Value : 0.58974
##          Neg Pred Value : 0.94649
##              Prevalence : 0.06971
##          Detection Rate : 0.01782
##    Detection Prevalence : 0.03021
##       Balanced Accuracy : 0.62112
##
##        'Positive' Class : bad
##
```

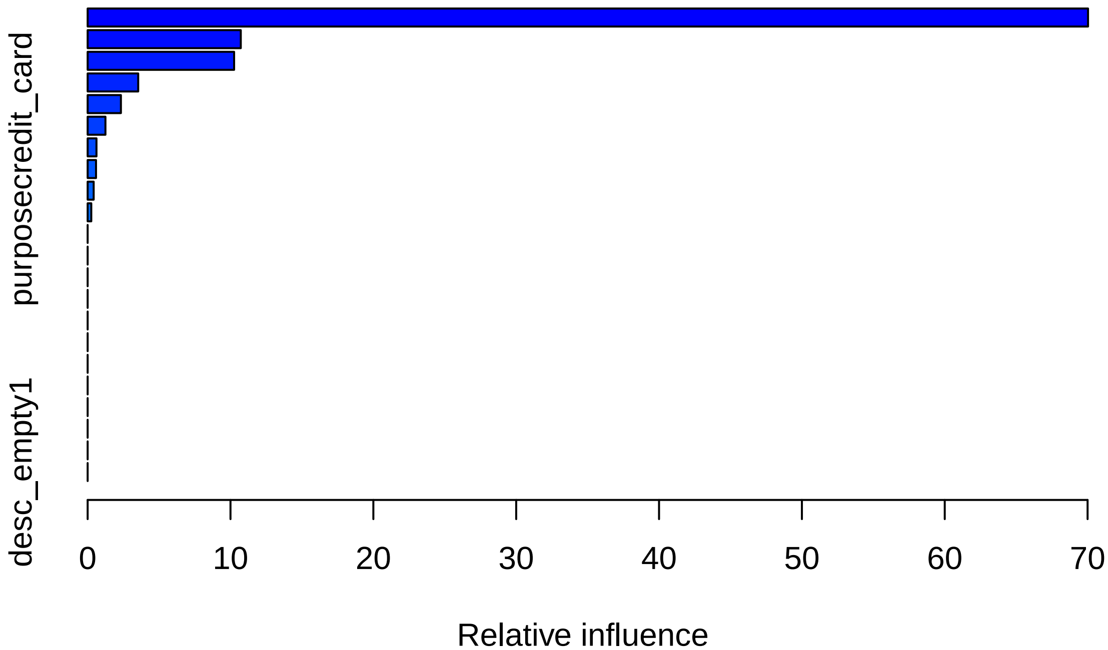



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.8529
```

Gradient Boost Model

```
## Loading required package: plyr

## Warning: package 'plyr' was built under R version 3.1.3

## -------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## -------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:reshape':
##
##     rename, round_any
##
## The following object is masked from 'package:lubridate':
##
```

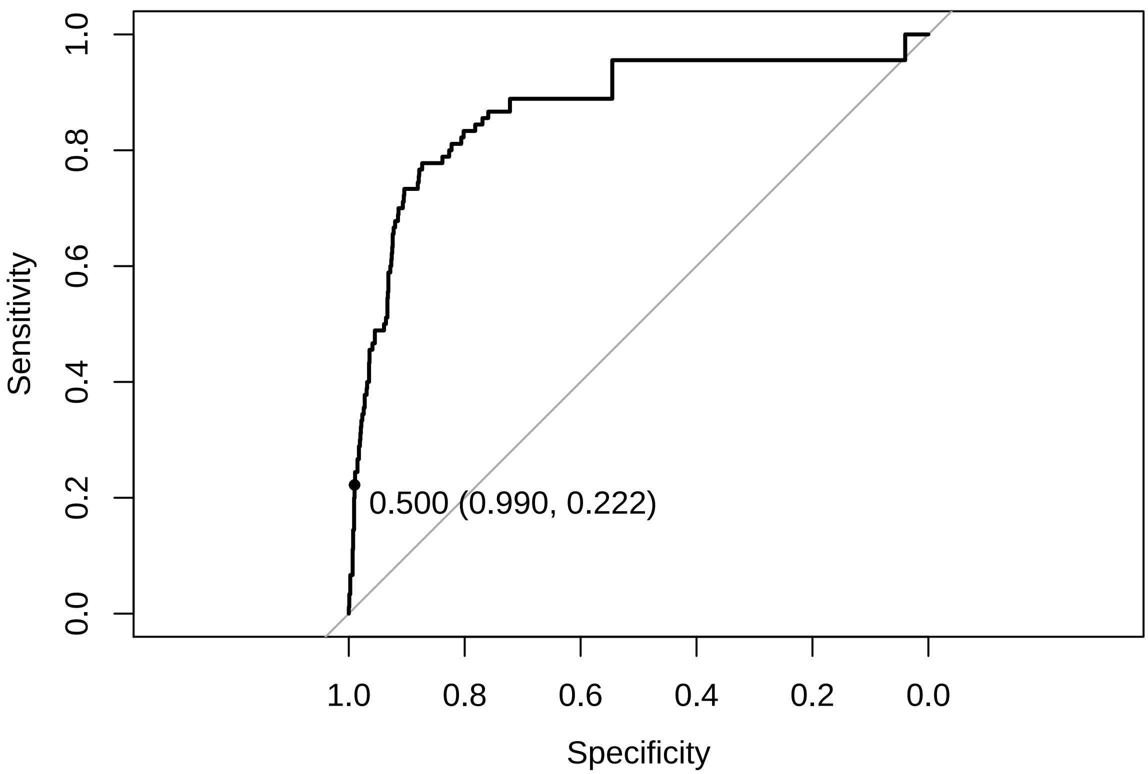
```
##     here
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## Stochastic Gradient Boosting
##
## 3879 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3879, 3879, 3879, 3879, 3879, 3879, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.9315846  0.2746793  0.004460033
##   1                  100      0.9305221  0.2731812  0.004811056
##   1                  150      0.9302392  0.2781188  0.005544541
##   2                   50      0.9332636  0.3065577  0.004945600
##   2                  100      0.9317234  0.3080410  0.005155197
##   2                  150      0.9308576  0.3097073  0.005087139
##   3                   50      0.9325115  0.3126589  0.005051484
##   3                  100      0.9317023  0.3194381  0.004382830
##   3                  150      0.9295464  0.3060641  0.004869173
##   Kappa SD
##   0.06294179
##   0.06290319
##   0.05216209
##   0.04617534
##   0.04941934
##   0.04880348
##   0.04759473
##   0.04865336
##   0.04405054
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 50, interaction.depth
##  = 2, shrinkage = 0.1 and n.minobsinnode = 10.
```




```
##                                                                 var
## last_fico_range_high                           last_fico_range_high
## fico_range_high                                     fico_range_high
## inq_last_6mths                                       inq_last_6mths
## revol_util                                               revol_util
## dti                                                             dti
## purposeother                                           purposeother
## term 60 months                                       term 60 months
## purposecredit_card                               purposecredit_card
## purposedebt_consolidation                 purposedebt_consolidation
## purposeeducational                               purposeeducational
## verification_statusSource Verified verification_statusSource Verified
## verification_statusVerified             verification_statusVerified
## purposehome_improvement                     purposehome_improvement
## purposehouse                                           purposehouse
## purposemajor_purchase                         purposemajor_purchase
## purposemedical                                       purposemedical
## purposemoving                                         purposemoving
## purposerenewable_energy                     purposerenewable_energy
## purposesmall_business                         purposesmall_business
## purposevacation                                     purposevacation
## purposewedding                                       purposewedding
## desc_empty1                                             desc_empty1
##                                       rel.inf
## last_fico_range_high               70.0311765
## fico_range_high                    10.7205994
## inq_last_6mths                     10.2527697
## revol_util                          3.5395880
## dti                                 2.3295672
## purposeother                        1.2455554
## term 60 months                      0.6185364
## purposecredit_card                  0.5857127
## purposedebt_consolidation           0.4216319
```

```
## purposeeducational                 0.2548628
## verification_statusSource Verified  0.0000000
## verification_statusVerified         0.0000000
## purposehome_improvement             0.0000000
## purposehouse                        0.0000000
## purposemajor_purchase               0.0000000
## purposemedical                      0.0000000
## purposemoving                       0.0000000
## purposerenewable_energy             0.0000000
## purposesmall_business               0.0000000
## purposevacation                     0.0000000
## purposewedding                      0.0000000
## desc_empty1                         0.0000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                    Overall
## last_fico_range_high              100.0000
## fico_range_high                    15.3083
## inq_last_6mths                     14.6403
## revol_util                          5.0543
## dti                                 3.3265
## purposeother                        1.7786
## term 60 months                      0.8832
## purposecredit_card                  0.8364
## purposedebt_consolidation           0.6021
## purposeeducational                  0.3639
## purposevacation                     0.0000
## desc_empty1                         0.0000
## purposemoving                       0.0000
## purposewedding                      0.0000
## purposehome_improvement             0.0000
## purposemedical                      0.0000
## verification_statusVerified         0.0000
## purposesmall_business               0.0000
## purposehouse                        0.0000
## verification_statusSource Verified   0.0000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1189   70
##       bad    12   20
##
##                Accuracy : 0.9365
##                  95% CI : (0.9218, 0.9492)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.2076
##
##                   Kappa : 0.3024
##  Mcnemar's Test P-Value : 3.082e-10
##
##             Sensitivity : 0.22222
```

```
##             Specificity : 0.99001
##          Pos Pred Value : 0.62500
##          Neg Pred Value : 0.94440
##              Prevalence : 0.06971
##          Detection Rate : 0.01549
##    Detection Prevalence : 0.02479
##       Balanced Accuracy : 0.60612
##
##        'Positive' Class : bad
##
```

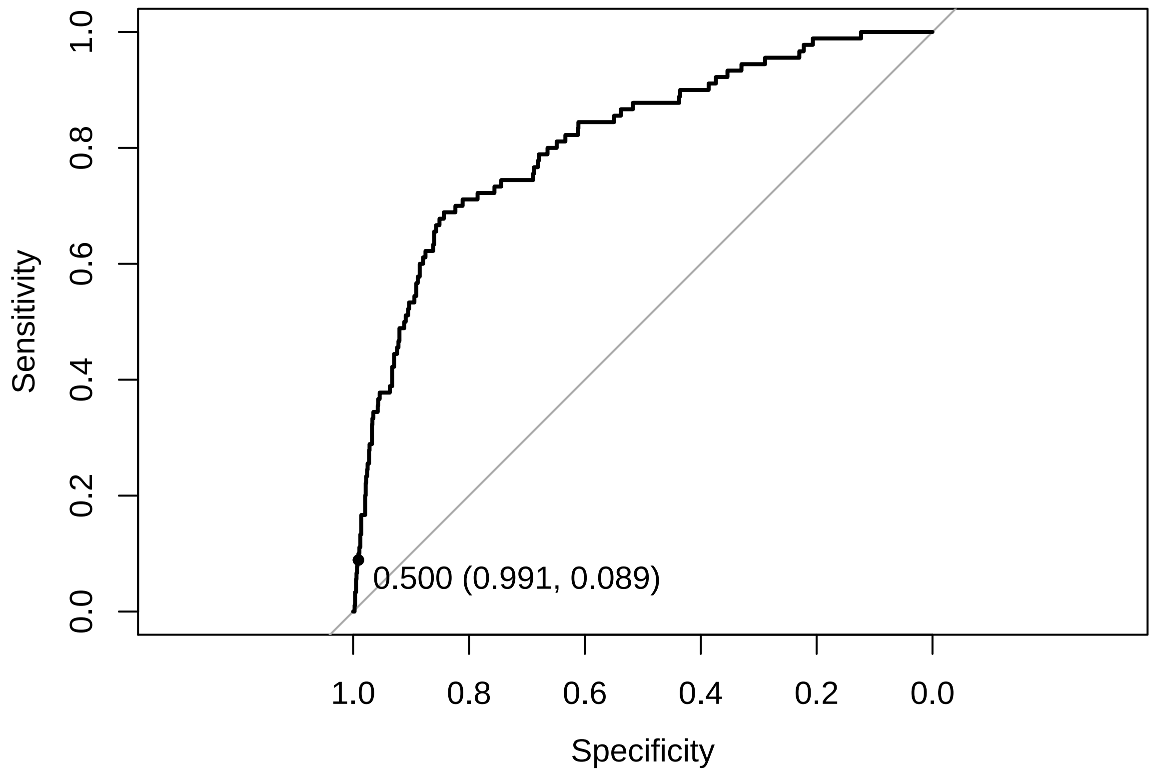



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.888
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 3879 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
```

```
##
## Summary of sample sizes: 3491, 3491, 3491, 3491, 3492, 3491, ...
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.9283411  0.1374781  0.008142182  0.10385356
##   0.50  0.9283398  0.1378456  0.007593441  0.09922288
##   1.00  0.9293707  0.1402285  0.007239328  0.10303976
##
## Tuning parameter 'sigma' was held constant at a value of 0.04150303
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.04150303 and C = 1.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                      Importance
## last_fico_range_high    100.000
## fico_range_high          38.782
## inq_last_6mths           28.886
## revol_util               18.958
## dti                      13.235
## purpose                  11.062
## term                      6.388
## verification_status       2.016
## desc_empty                0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1190   82
##       bad    11    8
##
##                Accuracy : 0.928
##                  95% CI : (0.9125, 0.9415)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.6539
##
##                   Kappa : 0.1255
##  Mcnemar's Test P-Value : 3.909e-13
##
##             Sensitivity : 0.088889
##             Specificity : 0.990841
##          Pos Pred Value : 0.421053
##          Neg Pred Value : 0.935535
##              Prevalence : 0.069713
##          Detection Rate : 0.006197
##    Detection Prevalence : 0.014717
##       Balanced Accuracy : 0.539865
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.8141
```

Neural Net Model

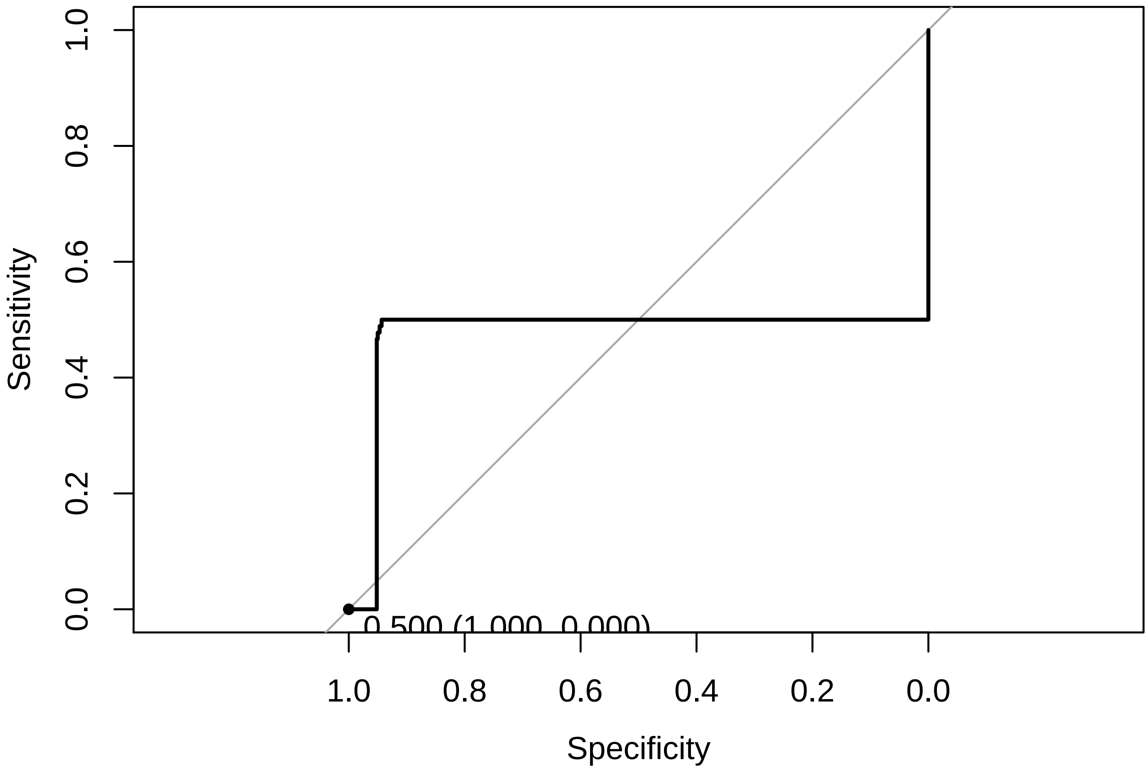
```
## Loading required package: nnet
```

```
## Neural Network
##
## 3879 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3879, 3879, 3879, 3879, 3879, 3879, ...
##
## Resampling results across tuning parameters:
##
##   size  decay         Accuracy   Kappa         Accuracy SD  Kappa SD
##    1    0.0000000000  0.9299527   0.0000000000  0.003994777  0.000000000
##    1    0.0001000000  0.9299527   0.0000000000  0.003994777  0.000000000
##    1    0.0003981072  0.9299527   0.0000000000  0.003994777  0.000000000
```

```
##    1    0.0015848932  0.9306794   0.0495578909  0.003815131  0.111351765
##    1    0.0063095734  0.9296965   0.0260068960  0.003641712  0.090327518
##    1    0.0251188643  0.9316946   0.1506353046  0.004488492  0.175584693
##    1    0.1000000000  0.9316384   0.1640536138  0.004468982  0.167797990
##    3    0.0000000000  0.9299527   0.0000000000  0.003994777  0.000000000
##    3    0.0001000000  0.9299527   0.0000000000  0.003994777  0.000000000
##    3    0.0003981072  0.9298654   0.0523789135  0.003796314  0.123172152
##    3    0.0015848932  0.9307909   0.0746254217  0.004490080  0.141790845
##    3    0.0063095734  0.9309808   0.1645813744  0.004887842  0.167901920
##    3    0.0251188643  0.9303914   0.2066172980  0.004107474  0.159968056
##    3    0.1000000000  0.9298198   0.2542104919  0.004905477  0.113258230
##    5    0.0000000000  0.9302378   0.0169923249  0.004353623  0.084961625
##    5    0.0001000000  0.9300087   0.0130180807  0.004016422  0.065090403
##    5    0.0003981072  0.9278842   0.0782727242  0.006786760  0.144100364
##    5    0.0015848932  0.9304548   0.1158054330  0.005077771  0.158836049
##    5    0.0063095734  0.9277099   0.1619095405  0.006057252  0.150578220
##    5    0.0251188643  0.9289388   0.2064899707  0.005964008  0.150903384
##    5    0.1000000000  0.9297105   0.2473133643  0.005206319  0.115582571
##    7    0.0000000000  0.9299527   0.0000000000  0.003994777  0.000000000
##    7    0.0001000000  0.9301225   0.0156044387  0.004235629  0.078022193
##    7    0.0003981072  0.9298968   0.0255572216  0.004083608  0.089292177
##    7    0.0015848932  0.9293139   0.1179278114  0.005540049  0.143934145
##    7    0.0063095734  0.9299806   0.1442885961  0.006227651  0.164467518
##    7    0.0251188643  0.9295211   0.1857392973  0.004378884  0.131471812
##    7    0.1000000000  0.9299561   0.1519550518  0.004963629  0.117277641
##    9    0.0000000000  0.9300383   0.0056216489  0.003990265  0.028108245
##    9    0.0001000000  0.9299527   0.0000000000  0.003994777  0.000000000
##    9    0.0003981072  0.9287082   0.0709179521  0.004792766  0.132747814
##    9    0.0015848932  0.9296738   0.1306185765  0.004273384  0.161810313
##    9    0.0063095734  0.9294436   0.1782066688  0.004805152  0.146991983
##    9    0.0251188643  0.9292612   0.2359785117  0.004707002  0.121889892
##    9    0.1000000000  0.9294139   0.2256015279  0.004508062  0.114273247
##   11    0.0000000000  0.9298409  -0.0002164249  0.004089062  0.001082125
##   11    0.0001000000  0.9298130   0.0032505439  0.004124154  0.016252720
##   11    0.0003981072  0.9299012   0.0489556177  0.004391611  0.114151712
##   11    0.0015848932  0.9285298   0.1276925189  0.005745291  0.165178579
##   11    0.0063095734  0.9278534   0.1677767712  0.005402512  0.161259544
##   11    0.0251188643  0.9300100   0.2241935982  0.004474518  0.113020474
##   11    0.1000000000  0.9286202   0.2370919783  0.004668513  0.099341890
##   13    0.0000000000  0.9301740   0.0106109274  0.003795388  0.053054637
##   13    0.0001000000  0.9294021   0.0268089985  0.004166887  0.095154068
##   13    0.0003981072  0.9286022   0.0820467489  0.003309702  0.138018679
##   13    0.0015848932  0.9286347   0.1416452585  0.005451765  0.171584358
##   13    0.0063095734  0.9279208   0.1730308427  0.004878645  0.146932888
##   13    0.0251188643  0.9303203   0.2278603730  0.004009130  0.138609045
##   13    0.1000000000  0.9291628   0.2168813275  0.004243674  0.114940276
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 1 and decay = 0.02511886.
## a 22-1-1 network with 25 weights
## options were - entropy fitting  decay=0.02511886
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##    0.20    0.59    0.93   -0.68    1.46    2.98    0.26   -1.20    0.00
##  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
```

```
##    0.19    0.00    0.00   -0.15    0.00    0.43    2.23    0.92   -1.81
## i18->h1 i19->h1 i20->h1 i21->h1 i22->h1
##   -0.72   -0.21    2.17   -1.55    0.21
##  b->o h1->o
## -0.34 -3.06
## nnet variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                     Overall
## purposedebt_consolidation         1.000e+02
## purposevacation                   7.487e+01
## last_fico_range_high              7.291e+01
## fico_range_high                   6.067e+01
## desc_empty1                       5.219e+01
## purposecredit_card                4.910e+01
## purposehome_improvement           4.023e+01
## verification_statusSource Verified 3.136e+01
## purposewedding                    3.096e+01
## inq_last_6mths                    2.404e+01
## verification_statusVerified       2.284e+01
## term 60 months                    1.965e+01
## purposesmall_business             1.440e+01
## purposeeducational                8.787e+00
## revol_util                        7.195e+00
## dti                               6.926e+00
## purposemajor_purchase             6.515e+00
## purposeother                      5.050e+00
## purposehouse                      2.730e-02
## purposemoving                     5.682e-04
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1201   90
##       bad     0    0
##
##                Accuracy : 0.9303
##                  95% CI : (0.915, 0.9436)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.528
##
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.93029
##              Prevalence : 0.06971
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
```

```
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.7213
```

Results for Grade B Loans

Approximately 16% of the Grade B loans in this dataset went bad. With the four models, we were able to predict between 48% and 59% of the bad loans. This predictive ability is based on a 50% probability classification cutoff. As the ROC curves show, it's possible to predict the bad loans with a higher probability, of course, with a higher false positive rate, though. The FICO range and the number of inquiries in the past 6 months were also important predictors for this loan grade.

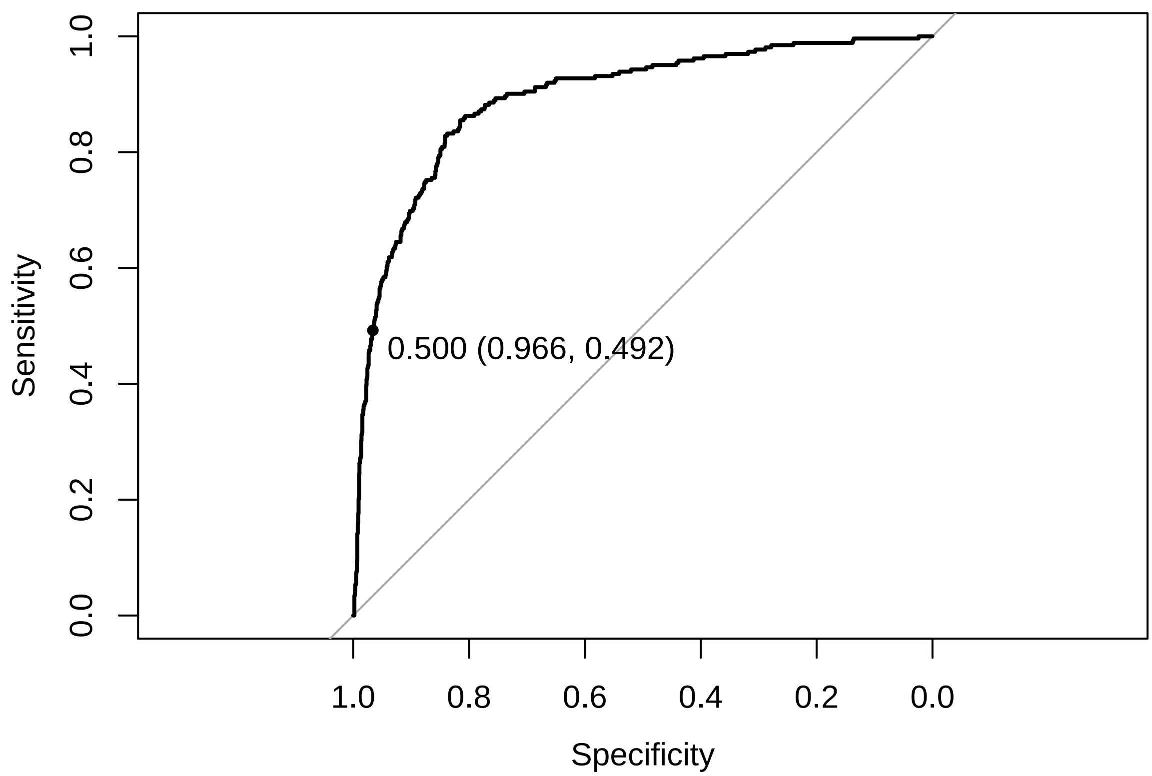
Logistic Regression Model

```
## Generalized Linear Model
##
## 4929 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
```

```
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 4929, 4929, 4929, 4929, 4929, 4929, ...
##
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.8746598  0.4483682  0.007394032  0.0236198
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6196  -0.4533  -0.2716  -0.1563   3.1072
##
## Coefficients:
##                                        Estimate Std. Error z value
## (Intercept)                           7.0767926  1.7203404   4.114
## `term 60 months`                      0.7165132  0.1374132   5.214
## `verification_statusSource Verified` -0.1320350  0.1340081  -0.985
## verification_statusVerified          -0.0237590  0.1175609  -0.202
## purposecredit_card                    0.4031126  0.2968169   1.358
## purposedebt_consolidation             0.2420080  0.2721458   0.889
## purposeeducational                    1.0127752  0.3995905   2.535
## purposehome_improvement              -0.0697073  0.3160148  -0.221
## purposehouse                          0.2662726  0.5214064   0.511
## purposemajor_purchase                 0.1106961  0.3341139   0.331
## purposemedical                        0.3305896  0.4305379   0.768
## purposemoving                         0.6034460  0.4348902   1.388
## purposeother                          0.3971603  0.2895945   1.371
## purposerenewable_energy               0.5478014  0.7642567   0.717
## purposesmall_business                 0.8652322  0.3283205   2.635
## purposevacation                      -0.3286216  0.5387682  -0.610
## purposewedding                       -0.3028271  0.4895150  -0.619
## fico_range_high                       0.0025532  0.0023032   1.109
## inq_last_6mths                        0.5757365  0.0346238  16.628
## revol_util                            0.0021635  0.0021146   1.023
## last_fico_range_high                 -0.0178601  0.0006632 -26.932
## desc_empty1                          -0.1831735  0.1204618  -1.521
## dti                                   0.0030663  0.0076509   0.401
##                                      Pr(>|z|)
## (Intercept)                          3.90e-05 ***
## `term 60 months`                     1.85e-07 ***
## `verification_statusSource Verified`  0.32449
## verification_statusVerified           0.83984
## purposecredit_card                    0.17443
## purposedebt_consolidation             0.37386
## purposeeducational                    0.01126 *
## purposehome_improvement               0.82542
## purposehouse                          0.60957
## purposemajor_purchase                 0.74041
```

```
## purposemedical                        0.44257
## purposemoving                         0.16526
## purposeother                          0.17024
## purposerenewable_energy               0.47351
## purposesmall_business                 0.00841 **
## purposevacation                       0.54190
## purposewedding                        0.53616
## fico_range_high                       0.26764
## inq_last_6mths                        < 2e-16 ***
## revol_util                            0.30627
## last_fico_range_high                  < 2e-16 ***
## desc_empty1                           0.12836
## dti                                   0.68859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4325.5  on 4928  degrees of freedom
## Residual deviance: 2929.2  on 4906  degrees of freedom
## AIC: 2975.2
##
## Number of Fisher Scoring iterations: 6
##
## glm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                       Overall
## last_fico_range_high                 100.0000
## inq_last_6mths                        61.4529
## `term 60 months`                      18.7513
## purposesmall_business                  9.1031
## purposeeducational                     8.7260
## desc_empty1                            4.9327
## purposemoving                          4.4351
## purposeother                           4.3747
## purposecredit_card                     4.3248
## fico_range_high                        3.3911
## revol_util                             3.0714
## `verification_statusSource Verified`   2.9300
## purposedebt_consolidation              2.5708
## purposemedical                         2.1166
## purposerenewable_energy                1.9255
## purposewedding                         1.5583
## purposevacation                        1.5258
## purposehouse                           1.1544
## dti                                    0.7433
## purposemajor_purchase                  0.4834
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1333  133
```

```
##       bad    47  129
##
##                Accuracy : 0.8904
##                  95% CI : (0.8743, 0.9051)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 4.351e-09
##
##                   Kappa : 0.5286
##  Mcnemar's Test P-Value : 2.365e-10
##
##             Sensitivity : 0.49237
##             Specificity : 0.96594
##          Pos Pred Value : 0.73295
##          Neg Pred Value : 0.90928
##              Prevalence : 0.15956
##          Detection Rate : 0.07856
##    Detection Prevalence : 0.10719
##       Balanced Accuracy : 0.72915
##
##        'Positive' Class : bad
##
```

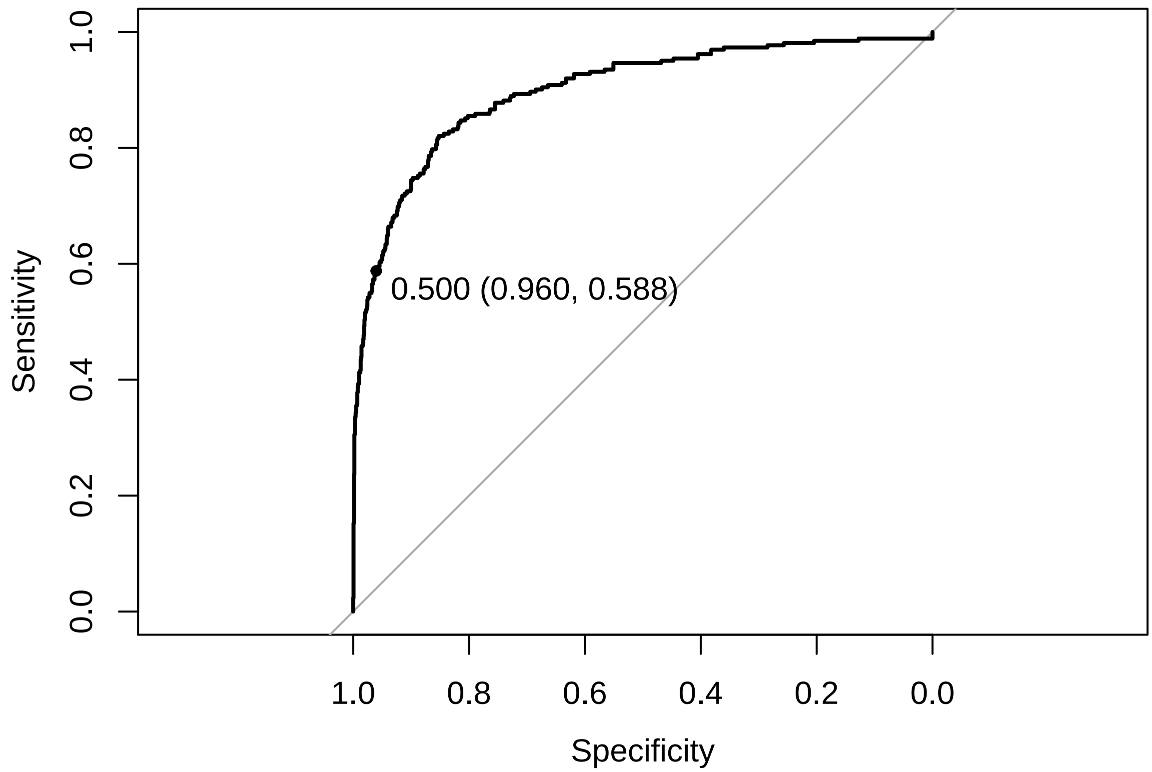



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.8933
```

Random Forest Model

```
## Random Forest
##
## 4929 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 4929, 4929, 4929, 4929, 4929, 4929, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.8666526  0.2513198  0.005839638  0.05013582
##   12    0.8929782  0.5490899  0.005669010  0.02148163
##   22    0.8896533  0.5393636  0.006499728  0.02407075
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 12.
##                 Length Class      Mode
## call               4   -none-     call
## type               1   -none-     character
## predicted       4929   factor     numeric
## err.rate        1500   -none-     numeric
## confusion          6   -none-     numeric
## votes           9858   matrix     numeric
## oob.times       4929   -none-     numeric
## classes            2   -none-     character
## importance        22   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               4929   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            22   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
## rf variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                    Overall
## last_fico_range_high              100.0000
## inq_last_6mths                     46.9503
## fico_range_high                    40.4123
## dti                                39.9121
```

```
## revol_util                          38.5455
## term 60 months                       5.1736
## verification_statusVerified          3.7508
## purposedebt_consolidation            3.3592
## desc_empty1                          3.3367
## verification_statusSource Verified   2.7632
## purposeother                         2.7450
## purposesmall_business                2.2455
## purposecredit_card                   2.2215
## purposehome_improvement              1.9474
## purposemajor_purchase                1.5279
## purposeeducational                   1.0514
## purposemedical                       0.9867
## purposemoving                        0.7148
## purposehouse                         0.2799
## purposewedding                       0.1050
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1325  108
##       bad    55  154
##
##                Accuracy : 0.9007
##                  95% CI : (0.8852, 0.9148)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 9.628e-13
##
##                   Kappa : 0.5968
##  Mcnemar's Test P-Value : 4.642e-05
##
##             Sensitivity : 0.58779
##             Specificity : 0.96014
##          Pos Pred Value : 0.73684
##          Neg Pred Value : 0.92463
##              Prevalence : 0.15956
##          Detection Rate : 0.09379
##    Detection Prevalence : 0.12728
##       Balanced Accuracy : 0.77397
##
##        'Positive' Class : bad
##
```

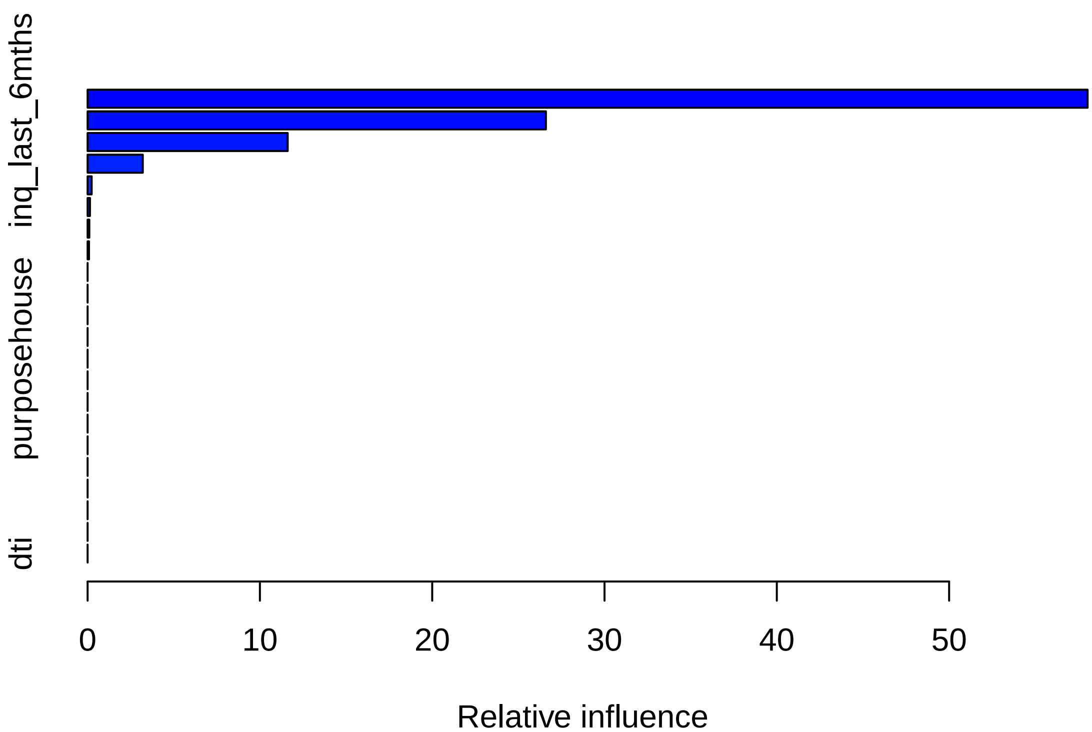



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.9009
```

Gradient Boost Model

```
## Stochastic Gradient Boosting
##
## 4929 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 4929, 4929, 4929, 4929, 4929, 4929, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.8795876  0.4587024  0.006395612
##   1                  100      0.8841563  0.4954586  0.006734249
##   1                  150      0.8852226  0.5052724  0.006948547
##   2                   50      0.9005761  0.5773318  0.004658106
##   2                  100      0.8993579  0.5761511  0.004655376
```

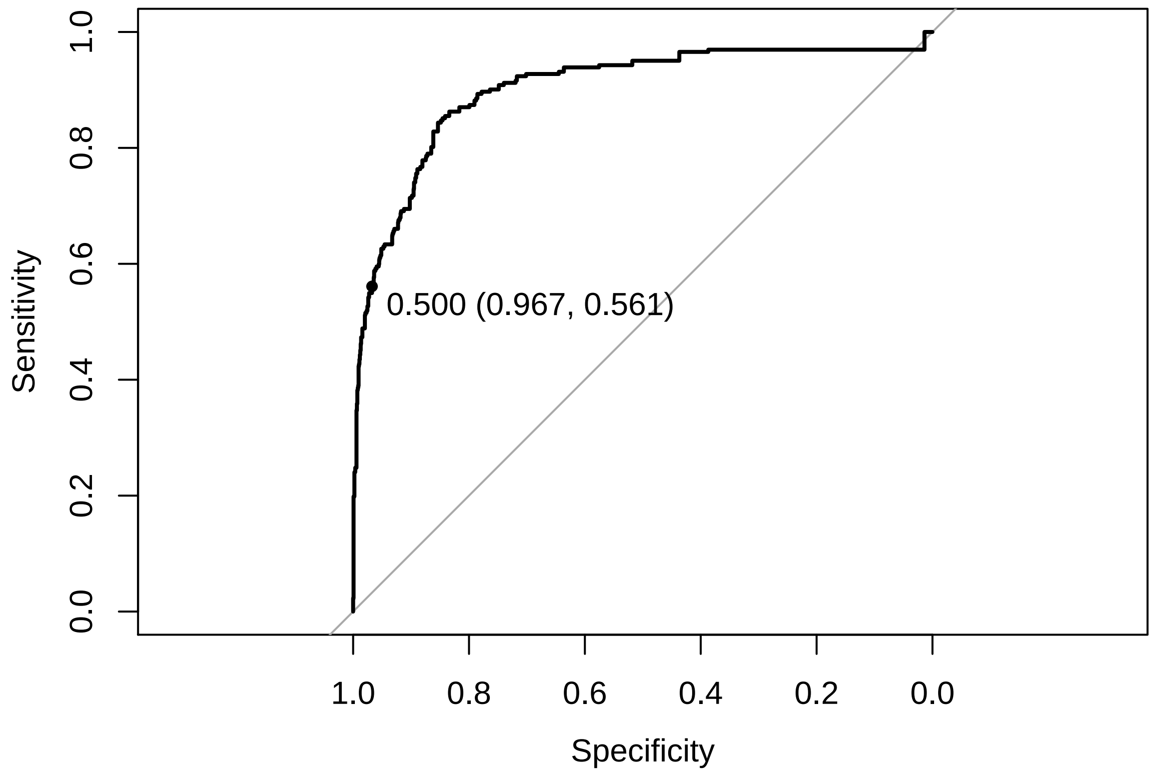
```
##   2                  150      0.8988032  0.5743232  0.004606226
##   3                   50      0.8996448  0.5742893  0.005470957
##   3                  100      0.8989181  0.5746370  0.004932349
##   3                  150      0.8970182  0.5690928  0.004457498
##   Kappa SD
##   0.02657461
##   0.02828961
##   0.02932224
##   0.01719115
##   0.01901795
##   0.01992406
##   0.02252652
##   0.01990045
##   0.02020754
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 50, interaction.depth
##  = 2, shrinkage = 0.1 and n.minobsinnode = 10.
```




```
##                                                                   var
## last_fico_range_high                             last_fico_range_high
## inq_last_6mths                                         inq_last_6mths
## fico_range_high                                       fico_range_high
## term 60 months                                         term 60 months
## purposesmall_business                           purposesmall_business
## purposehome_improvement                       purposehome_improvement
## revol_util                                                 revol_util
```

```
## purposeeducational                                         purposeeducational
## verification_statusSource Verified verification_statusSource Verified
## verification_statusVerified                 verification_statusVerified
## purposecredit_card                                          purposecredit_card
## purposedebt_consolidation                         purposedebt_consolidation
## purposehouse                                                      purposehouse
## purposemajor_purchase                                 purposemajor_purchase
## purposemedical                                                  purposemedical
## purposemoving                                                    purposemoving
## purposeother                                                      purposeother
## purposerenewable_energy                             purposerenewable_energy
## purposevacation                                                purposevacation
## purposewedding                                                  purposewedding
## desc_empty1                                                        desc_empty1
## dti                                                                        dti
##                                         rel.inf
## last_fico_range_high                58.03586689
## inq_last_6mths                      26.59840721
## fico_range_high                     11.61250471
## term 60 months                       3.20172544
## purposesmall_business                0.23550668
## purposehome_improvement              0.13559383
## revol_util                           0.09839214
## purposeeducational                   0.08200310
## verification_statusSource Verified  0.00000000
## verification_statusVerified          0.00000000
## purposecredit_card                   0.00000000
## purposedebt_consolidation            0.00000000
## purposehouse                         0.00000000
## purposemajor_purchase                0.00000000
## purposemedical                       0.00000000
## purposemoving                        0.00000000
## purposeother                         0.00000000
## purposerenewable_energy              0.00000000
## purposevacation                      0.00000000
## purposewedding                       0.00000000
## desc_empty1                          0.00000000
## dti                                  0.00000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                      Overall
## last_fico_range_high                100.0000
## inq_last_6mths                       45.8310
## fico_range_high                      20.0092
## term 60 months                        5.5168
## purposesmall_business                 0.4058
## purposehome_improvement               0.2336
## revol_util                            0.1695
## purposeeducational                    0.1413
## desc_empty1                           0.0000
## purposeother                          0.0000
## purposedebt_consolidation             0.0000
```

```
## verification_statusVerified          0.0000
## purposemajor_purchase                0.0000
## purposecredit_card                   0.0000
## purposemedical                       0.0000
## purposevacation                      0.0000
## purposehouse                         0.0000
## purposerenewable_energy              0.0000
## purposewedding                       0.0000
## verification_statusSource Verified   0.0000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1335  115
##       bad    45  147
##
##                Accuracy : 0.9026
##                  95% CI : (0.8872, 0.9165)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 1.793e-13
##
##                   Kappa : 0.5926
##  Mcnemar's Test P-Value : 4.899e-08
##
##             Sensitivity : 0.56107
##             Specificity : 0.96739
##          Pos Pred Value : 0.76562
##          Neg Pred Value : 0.92069
##              Prevalence : 0.15956
##          Detection Rate : 0.08952
##    Detection Prevalence : 0.11693
##       Balanced Accuracy : 0.76423
##
##        'Positive' Class : bad
##
```

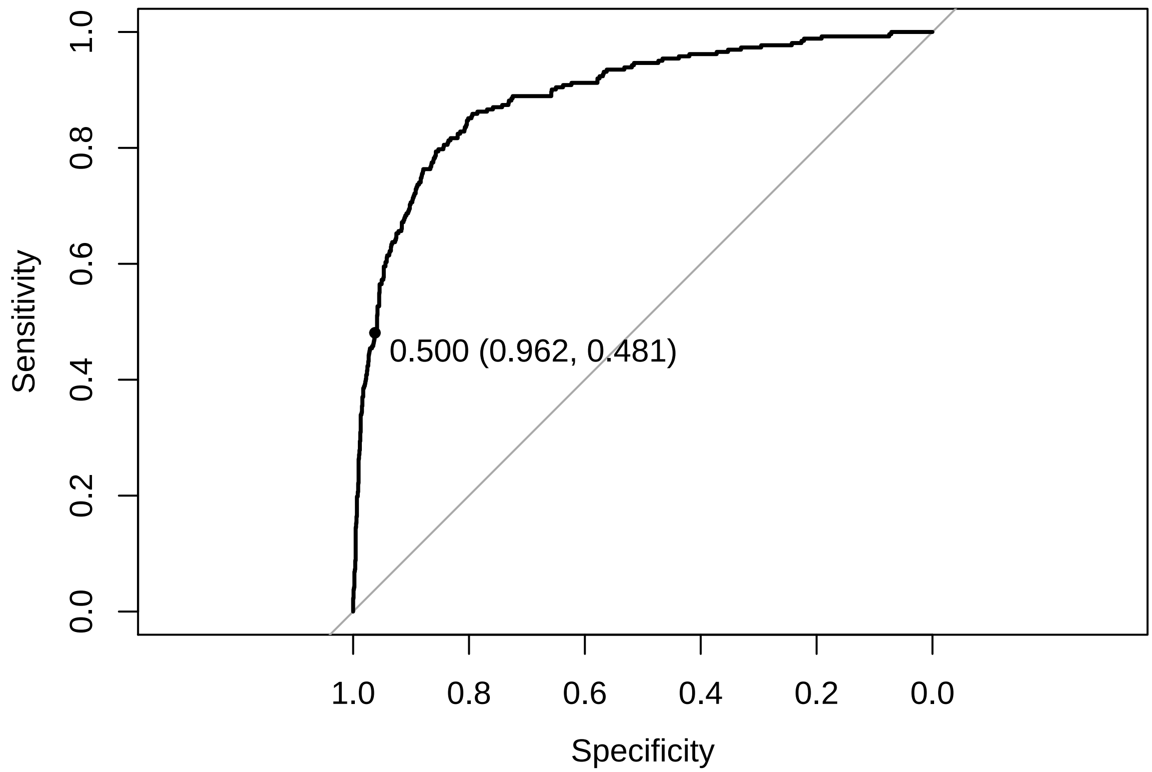



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.9088
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 4929 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 4437, 4436, 4435, 4435, 4436, 4437, ...
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.8768501  0.4787776  0.006651991  0.03118540
##   0.50  0.8784712  0.4616365  0.006365928  0.03458948
##   1.00  0.8788793  0.4574807  0.006141753  0.03471678
##
## Tuning parameter 'sigma' was held constant at a value of 0.04261215
```

```
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.04261215 and C = 1.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                      Importance
## last_fico_range_high    100.000
## inq_last_6mths           48.718
## purpose                  22.739
## term                     13.076
## revol_util               10.743
## dti                       6.760
## fico_range_high           4.398
## verification_status       1.999
## desc_empty                0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1328  136
##       bad    52  126
##
##                Accuracy : 0.8855
##                  95% CI : (0.8691, 0.9005)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 1.228e-07
##
##                   Kappa : 0.5094
##  Mcnemar's Test P-Value : 1.418e-09
##
##             Sensitivity : 0.48092
##             Specificity : 0.96232
##          Pos Pred Value : 0.70787
##          Neg Pred Value : 0.90710
##              Prevalence : 0.15956
##          Detection Rate : 0.07674
##    Detection Prevalence : 0.10840
##       Balanced Accuracy : 0.72162
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.8897
```

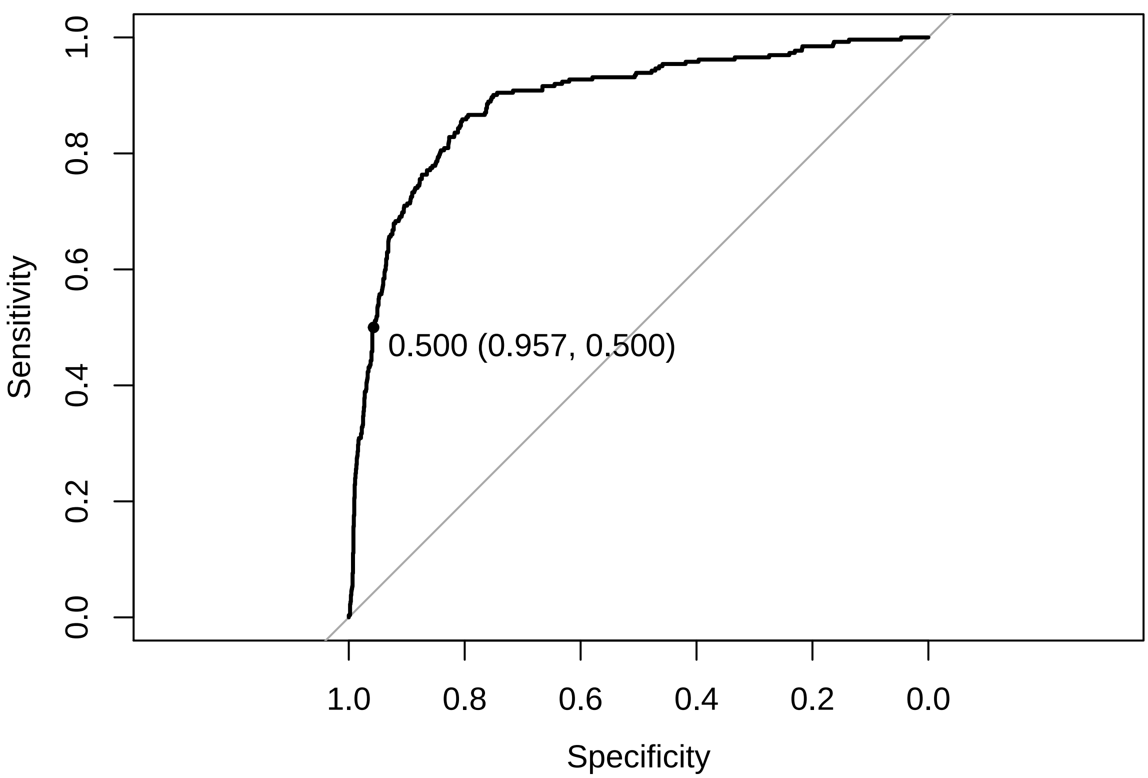
Neural Net Model

```
## Neural Network
##
## 4929 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 4929, 4929, 4929, 4929, 4929, 4929, ...
##
## Resampling results across tuning parameters:
##
##   size  decay         Accuracy   Kappa       Accuracy SD  Kappa SD
##    1    0.0000000000  0.8427527  0.00000000  0.005813760  0.00000000
##    1    0.0001000000  0.8427527  0.00000000  0.005813760  0.00000000
##    1    0.0003981072  0.8427527  0.00000000  0.005813760  0.00000000
##    1    0.0015848932  0.8500107  0.14339705  0.014890668  0.21391380
##    1    0.0063095734  0.8568336  0.28068998  0.013872048  0.21589130
```

```
##    1    0.0251188643  0.8584150  0.29009911  0.014771125  0.22308373
##    1    0.1000000000  0.8656052  0.39505359  0.012069989  0.15068973
##    3    0.0000000000  0.8461520  0.05410244  0.011107530  0.14956907
##    3    0.0001000000  0.8431679  0.03410103  0.008941837  0.11823251
##    3    0.0003981072  0.8419974  0.03711674  0.006103621  0.12846988
##    3    0.0015848932  0.8577950  0.26366398  0.015148330  0.22195219
##    3    0.0063095734  0.8615733  0.38149309  0.012549611  0.15148574
##    3    0.0251188643  0.8616931  0.40617229  0.015794601  0.09766932
##    3    0.1000000000  0.8727060  0.44938188  0.007132952  0.04077842
##    5    0.0000000000  0.8462769  0.07233671  0.011334083  0.16930620
##    5    0.0001000000  0.8428200  0.06883600  0.011022700  0.16132501
##    5    0.0003981072  0.8444160  0.04855569  0.006752677  0.13421102
##    5    0.0015848932  0.8509986  0.27271289  0.014618377  0.21208710
##    5    0.0063095734  0.8626243  0.38932949  0.014526727  0.12636263
##    5    0.0251188643  0.8682163  0.43133058  0.008638671  0.06042921
##    5    0.1000000000  0.8706676  0.44064566  0.009101136  0.04802515
##    7    0.0000000000  0.8451332  0.03553677  0.010512362  0.12310937
##    7    0.0001000000  0.8425657  0.05304369  0.007697716  0.14660722
##    7    0.0003981072  0.8554333  0.23648805  0.012886042  0.22140580
##    7    0.0015848932  0.8590382  0.33814699  0.013606836  0.17543000
##    7    0.0063095734  0.8632543  0.39863028  0.013965420  0.13026544
##    7    0.0251188643  0.8693183  0.43323136  0.008530796  0.05413403
##    7    0.1000000000  0.8719036  0.42969957  0.007749452  0.04870954
##    9    0.0000000000  0.8428385  0.04981440  0.007770439  0.13902471
##    9    0.0001000000  0.8450211  0.03463544  0.010402702  0.12022497
##    9    0.0003981072  0.8527428  0.25041947  0.014197990  0.22755221
##    9    0.0015848932  0.8581241  0.30641424  0.014620641  0.19751307
##    9    0.0063095734  0.8646812  0.38589837  0.010744674  0.12271868
##    9    0.0251188643  0.8666306  0.42583636  0.009794696  0.05199239
##    9    0.1000000000  0.8708260  0.42626915  0.008006621  0.05727133
##   11    0.0000000000  0.8458256  0.05092983  0.010770219  0.14126475
##   11    0.0001000000  0.8447387  0.08817024  0.011049479  0.18022411
##   11    0.0003981072  0.8551664  0.28754005  0.012717357  0.20343670
##   11    0.0015848932  0.8600755  0.38434534  0.015537611  0.12847500
##   11    0.0063095734  0.8666186  0.41437058  0.010034134  0.06309706
##   11    0.0251188643  0.8698524  0.44424793  0.009857013  0.04682370
##   11    0.1000000000  0.8684467  0.42660757  0.006439968  0.05389186
##   13    0.0000000000  0.8472021  0.15453293  0.013471493  0.21167081
##   13    0.0001000000  0.8464549  0.09209396  0.010283026  0.18805249
##   13    0.0003981072  0.8593603  0.33479253  0.012069677  0.17368209
##   13    0.0015848932  0.8588446  0.37867479  0.014447981  0.14506045
##   13    0.0063095734  0.8614133  0.42751666  0.011003684  0.04798877
##   13    0.0251188643  0.8684749  0.42738352  0.008621341  0.04965990
##   13    0.1000000000  0.8702371  0.42971129  0.006928472  0.04027583
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 3 and decay = 0.1.
## a 22-3-1 network with 73 weights
## options were - entropy fitting  decay=0.1
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##    0.00   -0.03    0.01   -0.25   -0.03    0.16   -0.11   -0.09    0.23
##  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
##   -0.08   -0.03    0.00   -0.04    0.00   -0.01   -0.01    0.00    0.25
## i18->h1 i19->h1 i20->h1 i21->h1 i22->h1
```

```
##   -0.23   -0.03   -0.23    0.13    0.37
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##   -4.69   -0.46    0.13    0.06   -0.31   -0.25   -0.84    0.03   -0.18
##  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##   -0.16   -0.37   -0.61   -0.34   -0.12   -0.95    0.42    0.24   -0.01
## i18->h2 i19->h2 i20->h2 i21->h2 i22->h2
##   -0.53    0.00    0.01    0.21    0.00
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##    0.04    0.48    0.79    0.06    0.98   -1.06   -0.02    0.47   -0.50
##  i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##   -0.10   -0.09    0.01    0.79    0.05   -0.49    0.00   -0.01    0.81
## i18->h3 i19->h3 i20->h3 i21->h3 i22->h3
##    0.75    0.77   -1.12    0.45   -0.20
##  b->o h1->o h2->o h3->o
## -0.64  1.19 -6.09  0.99
## nnet variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                   Overall
## inq_last_6mths                     100.00
## purposedebt_consolidation           81.68
## last_fico_range_high                80.64
## purposesmall_business               78.62
## fico_range_high                     71.20
## purposeeducational                  67.68
## dti                                 66.08
## purposehouse                        65.05
## purposecredit_card                  56.92
## purposeother                        54.92
## desc_empty1                         47.56
## term 60 months                      46.58
## verification_statusVerified         41.45
## verification_statusSource Verified  32.92
## purposemoving                       31.87
## purposehome_improvement             28.01
## revol_util                          27.44
## purposemedical                      23.05
## purposevacation                     19.30
## purposemajor_purchase               18.54
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1321  131
##       bad    59  131
##
##                Accuracy : 0.8843
##                  95% CI : (0.8678, 0.8994)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 2.669e-07
##
##                   Kappa : 0.5145
##  Mcnemar's Test P-Value : 2.593e-07
```

```
##
##             Sensitivity : 0.50000
##             Specificity : 0.95725
##          Pos Pred Value : 0.68947
##          Neg Pred Value : 0.90978
##              Prevalence : 0.15956
##          Detection Rate : 0.07978
##    Detection Prevalence : 0.11571
##       Balanced Accuracy : 0.72862
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.8884
```

Results for Grade C Loans

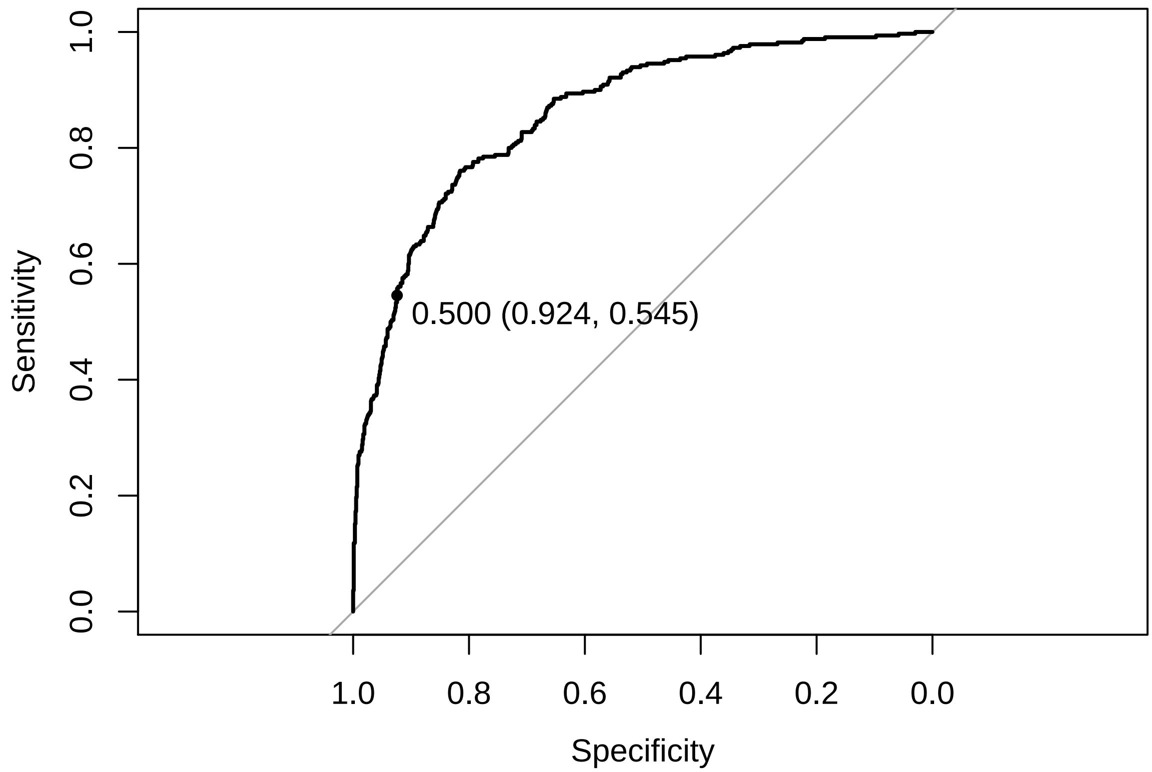
Approximately 25% of the Grade C loans in this dataset went bad. With the four models, we were able to correctly predict between 55% and 65% of the bad loans. This predictive ability is based on a 50% probability classification cutoff. As the ROC curves show, it's possible to predict the bad loans with a higher probability, of course, with a higher false positive rate, though. The FICO range and the number of inquiries in the past 6 months were also important predictors for this loan grade.

Logistic Regression Model

```
## Generalized Linear Model
##
## 3919 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3919, 3919, 3919, 3919, 3919, 3919, ...
##
## Resampling results
##
##   Accuracy  Kappa      Accuracy SD  Kappa SD
##   0.837957  0.5339636  0.01015      0.03099186
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4385  -0.5398  -0.3037   0.0547   3.1294
##
## Coefficients:
##                                      Estimate Std. Error z value
## (Intercept)                         4.8282981  1.8071389   2.672
## `term 60 months`                    0.3522714  0.1451884   2.426
## `verification_statusSource Verified` -0.5407083  0.1372003  -3.941
## verification_statusVerified        -0.2711942  0.1156667  -2.345
## purposecredit_card                  0.3834916  0.3405326   1.126
## purposedebt_consolidation           0.3598467  0.3174281   1.134
## purposeeducational                  0.2937829  0.4435413   0.662
## purposehome_improvement             0.1935797  0.3581491   0.541
## purposehouse                        0.6694480  0.5779723   1.158
## purposemajor_purchase               0.1052773  0.3747967   0.281
## purposemedical                      0.6581225  0.4442072   1.482
## purposemoving                      -0.3485626  0.4905935  -0.710
## purposeother                        0.2777299  0.3330013   0.834
## purposerenewable_energy             1.7078222  1.1839431   1.442
## purposesmall_business               0.8138861  0.3778411   2.154
## purposevacation                     0.8100797  0.5669364   1.429
## purposewedding                      0.4536172  0.4343918   1.044
## fico_range_high                     0.0046105  0.0025764   1.789
## inq_last_6mths                      0.7431495  0.0335607  22.143
## revol_util                         -0.0016015  0.0019099  -0.839
## last_fico_range_high               -0.0166819  0.0006801 -24.530
## desc_empty1                        -0.1510430  0.1200550  -1.258
## dti                                 0.0251223  0.0077898   3.225
##                                    Pr(>|z|)
## (Intercept)                         0.00754 **
```

```
## `term 60 months`                       0.01525 *
## `verification_statusSource Verified` 8.11e-05 ***
## verification_statusVerified           0.01905 *
## purposecredit_card                    0.26010
## purposedebt_consolidation             0.25695
## purposeeducational                    0.50774
## purposehome_improvement               0.58885
## purposehouse                          0.24675
## purposemajor_purchase                 0.77879
## purposemedical                        0.13846
## purposemoving                         0.47740
## purposeother                          0.40427
## purposerenewable_energy               0.14917
## purposesmall_business                 0.03124 *
## purposevacation                       0.15304
## purposewedding                        0.29637
## fico_range_high                       0.07354 .
## inq_last_6mths                        < 2e-16 ***
## revol_util                            0.40173
## last_fico_range_high                  < 2e-16 ***
## desc_empty1                           0.20835
## dti                                   0.00126 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4430.0  on 3918  degrees of freedom
## Residual deviance: 2838.6  on 3896  degrees of freedom
## AIC: 2884.6
##
## Number of Fisher Scoring iterations: 5
##
## glm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                      Overall
## last_fico_range_high                 100.000
## inq_last_6mths                        90.157
## `verification_statusSource Verified`  15.094
## dti                                   12.141
## `term 60 months`                       8.847
## verification_statusVerified            8.510
## purposesmall_business                  7.725
## fico_range_high                        6.221
## purposemedical                         4.951
## purposerenewable_energy                4.790
## purposevacation                        4.734
## desc_empty1                            4.030
## purposehouse                           3.618
## purposedebt_consolidation              3.517
## purposecredit_card                     3.486
## purposewedding                         3.148
```

```
## revol_util                             2.300
## purposeother                           2.281
## purposemoving                          1.772
## purposeeducational                     1.573
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  902 150
##       bad    74 180
##
##                Accuracy : 0.8285
##                  95% CI : (0.8069, 0.8486)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : 1.301e-12
##
##                   Kappa : 0.5084
##  Mcnemar's Test P-Value : 5.411e-07
##
##             Sensitivity : 0.5455
##             Specificity : 0.9242
##          Pos Pred Value : 0.7087
##          Neg Pred Value : 0.8574
##              Prevalence : 0.2527
##          Detection Rate : 0.1378
##    Detection Prevalence : 0.1945
##       Balanced Accuracy : 0.7348
##
##        'Positive' Class : bad
##
```



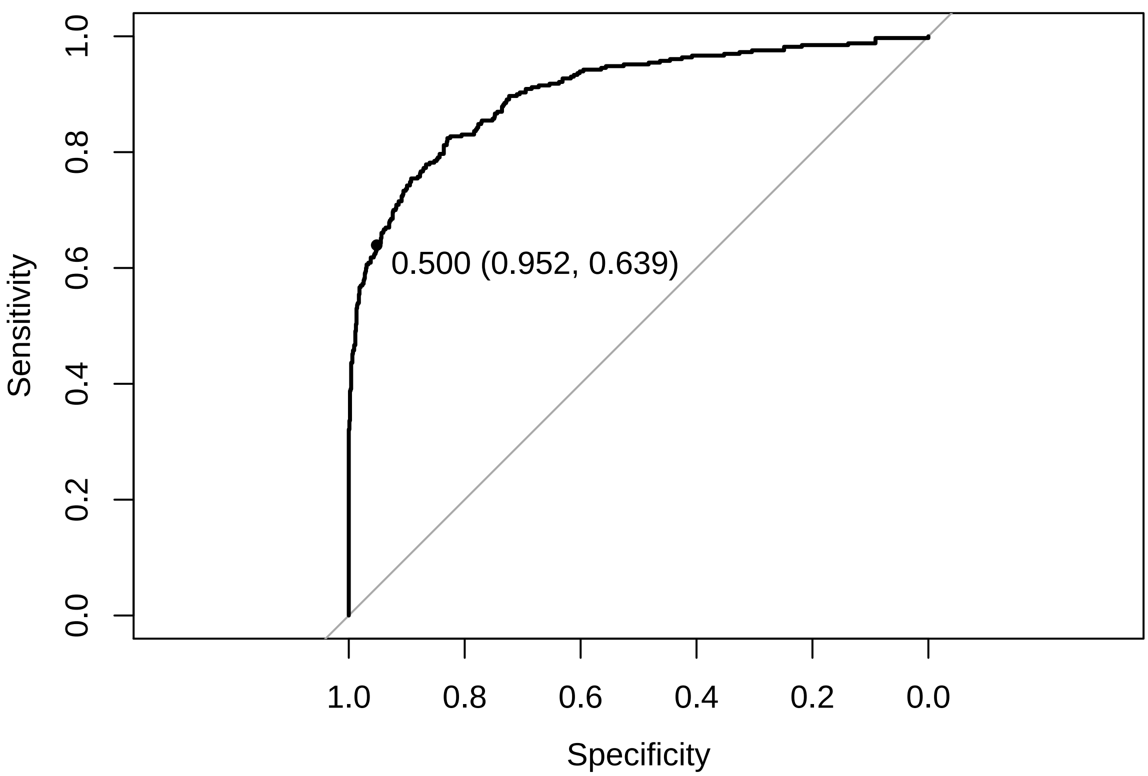

```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.859
```

Random Forest Model

```
## Random Forest
##
## 3919 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3919, 3919, 3919, 3919, 3919, 3919, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.8484946  0.5116558  0.009391680  0.03249618
##   12    0.8713681  0.6395364  0.007385996  0.01930579
##   22    0.8652840  0.6236837  0.005709080  0.01538639
##
## Accuracy was used to select the optimal model using  the largest value.
```

```
## The final value used for the model was mtry = 12.
##                 Length Class      Mode
## call               4   -none-     call
## type               1   -none-     character
## predicted       3919   factor     numeric
## err.rate        1500   -none-     numeric
## confusion          6   -none-     numeric
## votes           7838   matrix     numeric
## oob.times       3919   -none-     numeric
## classes            2   -none-     character
## importance        22   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               3919   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            22   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
## rf variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                      Overall
## inq_last_6mths                      100.0000
## last_fico_range_high                 90.9519
## dti                                  36.4097
## revol_util                           34.9793
## fico_range_high                      27.3696
## term 60 months                        5.0469
## verification_statusVerified           3.5482
## purposedebt_consolidation             3.3888
## desc_empty1                           3.2381
## verification_statusSource Verified    2.6966
## purposecredit_card                    2.4399
## purposeother                          2.3728
## purposehome_improvement               1.9858
## purposesmall_business                 1.8100
## purposemoving                         0.9619
## purposemedical                        0.9563
## purposemajor_purchase                 0.8799
## purposeeducational                    0.7209
## purposewedding                        0.2392
## purposevacation                       0.2283
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  929 120
```

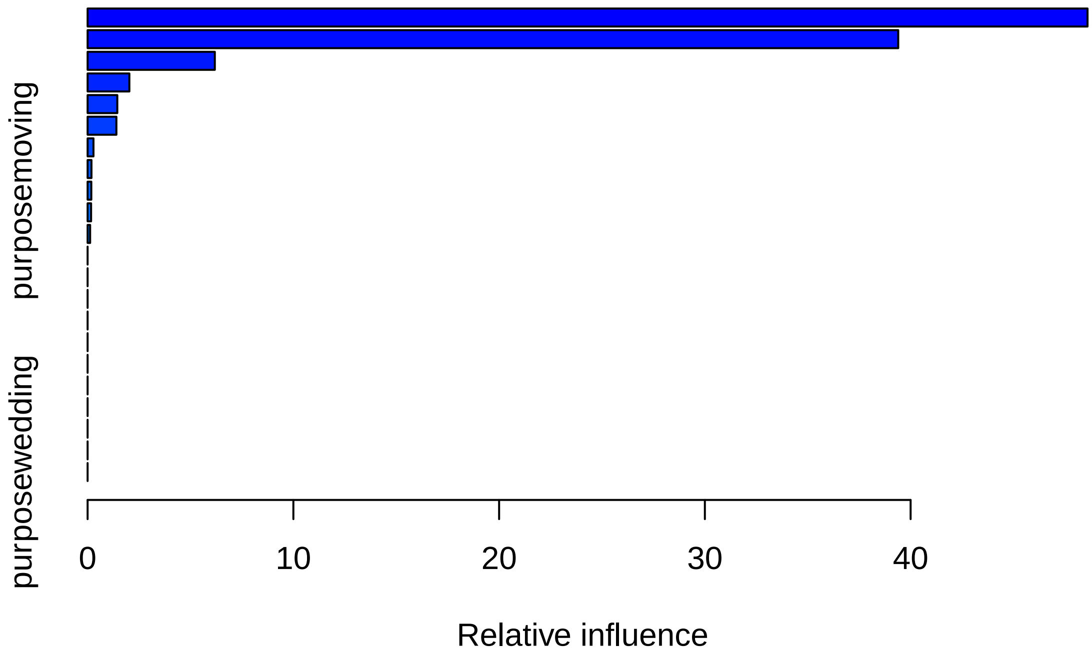
```
##       bad    47 210
##
##                Accuracy : 0.8721
##                  95% CI : (0.8528, 0.8898)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6347
##  Mcnemar's Test P-Value : 2.525e-08
##
##             Sensitivity : 0.6364
##             Specificity : 0.9518
##          Pos Pred Value : 0.8171
##          Neg Pred Value : 0.8856
##              Prevalence : 0.2527
##          Detection Rate : 0.1608
##    Detection Prevalence : 0.1968
##       Balanced Accuracy : 0.7941
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.9041
```

Gradient Boost Model

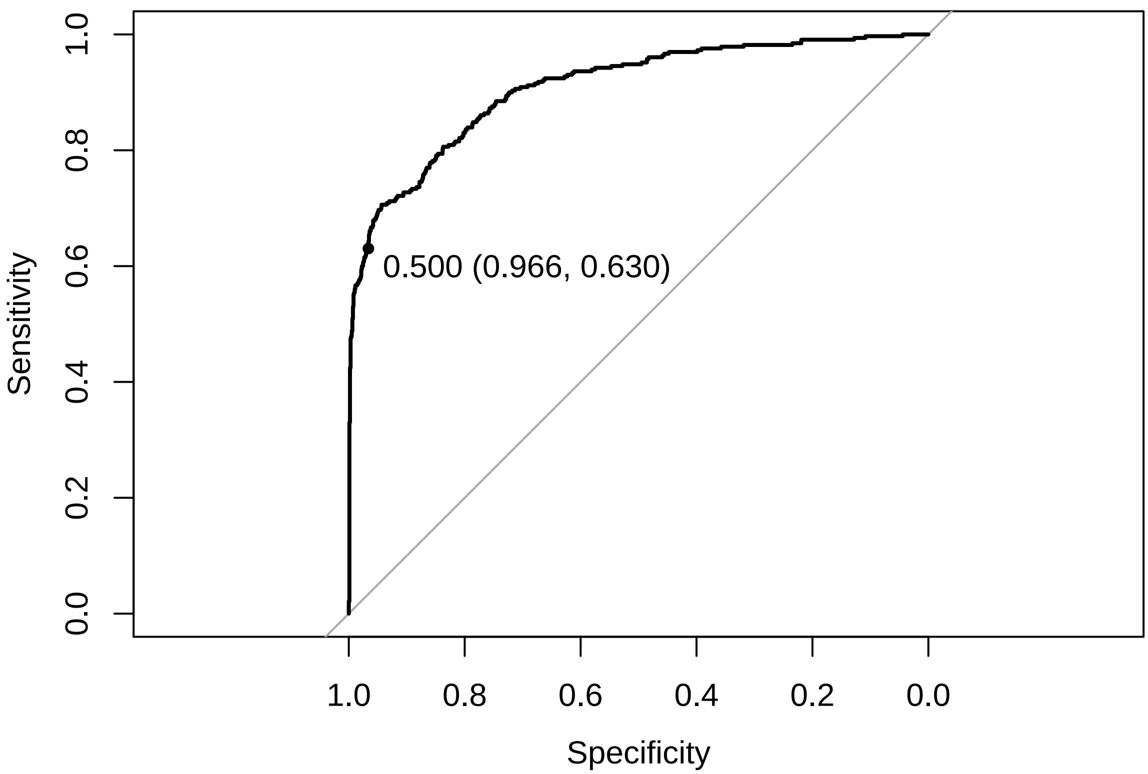
```
## Stochastic Gradient Boosting
##
## 3919 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3919, 3919, 3919, 3919, 3919, 3919, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.8703165  0.6236488  0.007396589
##   1                  100      0.8712055  0.6308410  0.007688464
##   1                  150      0.8734992  0.6395033  0.007476500
##   2                   50      0.8765195  0.6457668  0.007359885
##   2                  100      0.8779044  0.6526400  0.006998734
##   2                  150      0.8772667  0.6515767  0.006685624
##   3                   50      0.8783938  0.6527329  0.007163479
##   3                  100      0.8771426  0.6516696  0.007538935
##   3                  150      0.8752912  0.6475722  0.006568296
##   Kappa SD
##   0.01920238
##   0.01988459
##   0.01906593
##   0.01991299
##   0.01871444
##   0.01812448
##   0.01829630
##   0.02016125
##   0.01808832
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 50, interaction.depth
##  = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```




```
##                                                                      var
## inq_last_6mths                                            inq_last_6mths
## last_fico_range_high                                last_fico_range_high
## fico_range_high                                          fico_range_high
## term 60 months                                            term 60 months
## dti                                                                  dti
## revol_util                                                    revol_util
## purposesmall_business                              purposesmall_business
## desc_empty1                                                  desc_empty1
## purposemoving                                              purposemoving
## purposecredit_card                                    purposecredit_card
## verification_statusSource Verified verification_statusSource Verified
## verification_statusVerified                  verification_statusVerified
## purposedebt_consolidation                      purposedebt_consolidation
## purposeeducational                                    purposeeducational
## purposehome_improvement                          purposehome_improvement
## purposehouse                                                purposehouse
## purposemajor_purchase                              purposemajor_purchase
## purposemedical                                            purposemedical
## purposeother                                                purposeother
## purposerenewable_energy                          purposerenewable_energy
## purposevacation                                          purposevacation
## purposewedding                                            purposewedding
##                                       rel.inf
## inq_last_6mths                     48.5989428
## last_fico_range_high               39.3981668
## fico_range_high                     6.1841942
## term 60 months                      2.0290404
## dti                                 1.4440477
## revol_util                          1.3993865
## purposesmall_business               0.2851919
## desc_empty1                         0.1877967
## purposemoving                       0.1803855
```

```
## purposecredit_card                  0.1694137
## verification_statusSource Verified  0.1234338
## verification_statusVerified         0.0000000
## purposedebt_consolidation           0.0000000
## purposeeducational                  0.0000000
## purposehome_improvement             0.0000000
## purposehouse                        0.0000000
## purposemajor_purchase               0.0000000
## purposemedical                      0.0000000
## purposeother                        0.0000000
## purposerenewable_energy             0.0000000
## purposevacation                     0.0000000
## purposewedding                      0.0000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                     Overall
## inq_last_6mths                     100.0000
## last_fico_range_high                81.0680
## fico_range_high                     12.7250
## term 60 months                       4.1751
## dti                                  2.9714
## revol_util                           2.8795
## purposesmall_business                0.5868
## desc_empty1                          0.3864
## purposemoving                        0.3712
## purposecredit_card                   0.3486
## verification_statusSource Verified   0.2540
## purposewedding                       0.0000
## purposeother                         0.0000
## purposedebt_consolidation            0.0000
## purposehome_improvement              0.0000
## purposemedical                       0.0000
## purposehouse                         0.0000
## purposevacation                      0.0000
## verification_statusVerified          0.0000
## purposeeducational                   0.0000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  943 122
##       bad    33 208
##
##                Accuracy : 0.8813
##                  95% CI : (0.8625, 0.8984)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6549
##  Mcnemar's Test P-Value : 1.568e-12
##
##             Sensitivity : 0.6303
```

```
##             Specificity : 0.9662
##          Pos Pred Value : 0.8631
##          Neg Pred Value : 0.8854
##              Prevalence : 0.2527
##          Detection Rate : 0.1593
##    Detection Prevalence : 0.1845
##       Balanced Accuracy : 0.7982
##
##        'Positive' Class : bad
##
```

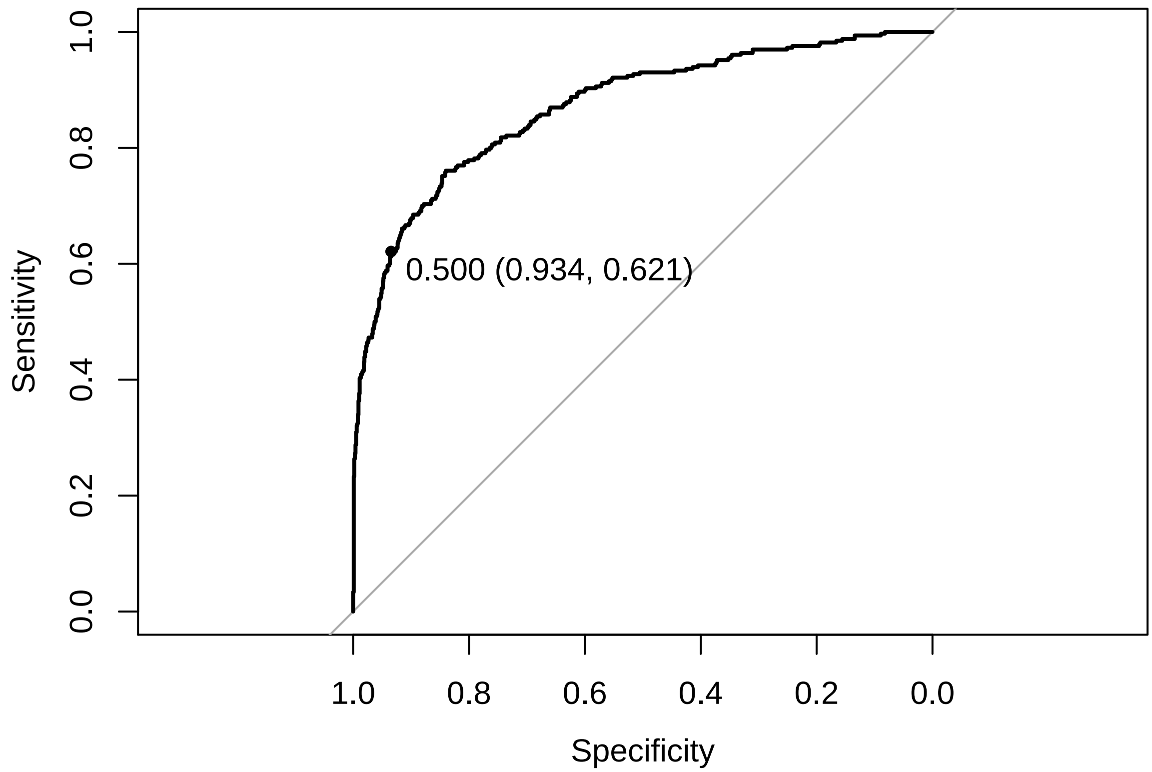



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.909
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 3919 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
```

```
##
## Summary of sample sizes: 3527, 3528, 3527, 3527, 3527, 3527, ...
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.8479253  0.5686628  0.01507753   0.04221153
##   0.50  0.8489457  0.5686387  0.01544019   0.04199727
##   1.00  0.8560905  0.5906895  0.01697615   0.04751378
##
## Tuning parameter 'sigma' was held constant at a value of 0.04429943
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.04429943 and C = 1.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                      Importance
## last_fico_range_high    100.000
## inq_last_6mths           80.437
## revol_util               32.654
## purpose                  26.695
## fico_range_high          19.644
## dti                      14.130
## term                     11.135
## desc_empty                4.003
## verification_status       0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  912 125
##       bad    64 205
##
##                Accuracy : 0.8553
##                  95% CI : (0.835, 0.8739)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5918
##  Mcnemar's Test P-Value : 1.275e-05
##
##             Sensitivity : 0.6212
##             Specificity : 0.9344
##          Pos Pred Value : 0.7621
##          Neg Pred Value : 0.8795
##              Prevalence : 0.2527
##          Detection Rate : 0.1570
##    Detection Prevalence : 0.2060
##       Balanced Accuracy : 0.7778
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.8702
```

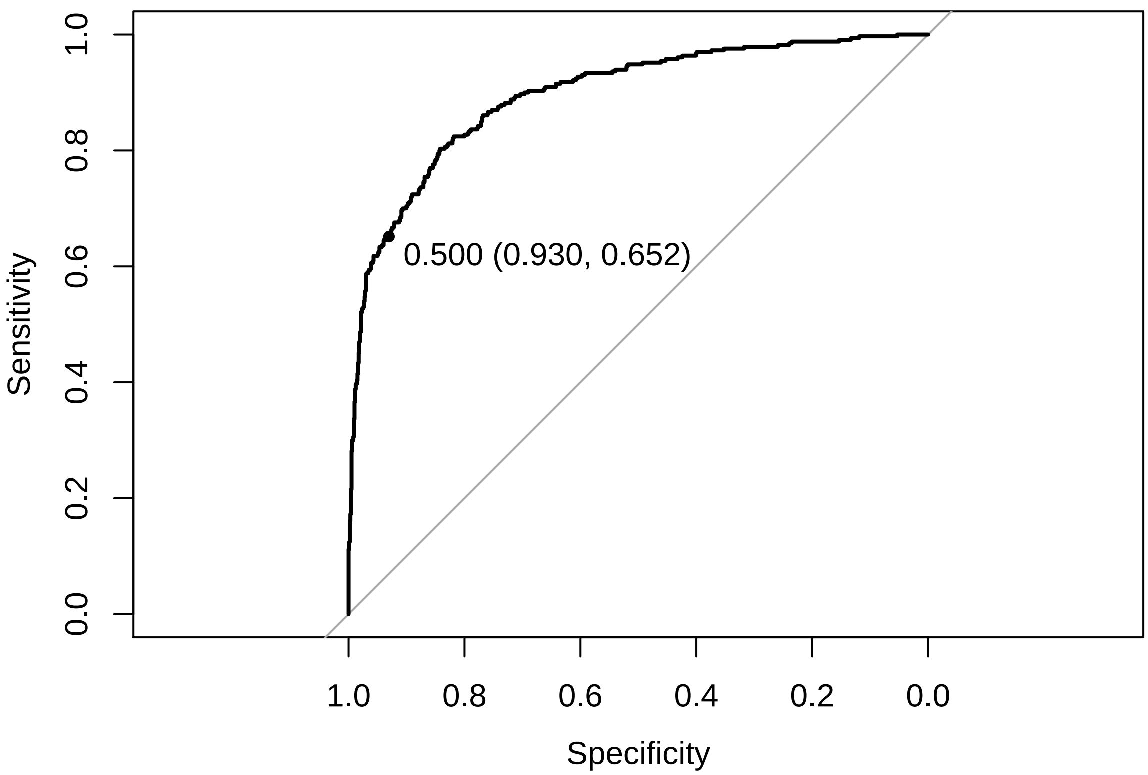
Neural Net Model

```
## Neural Network
##
## 3919 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3919, 3919, 3919, 3919, 3919, 3919, ...
##
## Resampling results across tuning parameters:
##
##   size  decay         Accuracy   Kappa       Accuracy SD  Kappa SD
##    1    0.0000000000  0.7468994  0.00000000  0.009718752  0.00000000
##    1    0.0001000000  0.7505585  0.02117868  0.018596486  0.10589340
##    1    0.0003981072  0.7535250  0.04046547  0.024159210  0.14103366
##    1    0.0015848932  0.7708928  0.17655937  0.036893717  0.24935598
##    1    0.0063095734  0.8016811  0.36621587  0.038393275  0.24279813
```

```
##    1     0.0251188643  0.7979983  0.30378061  0.045409870  0.27583948
##    1     0.1000000000  0.8100569  0.37349308  0.041895072  0.26213357
##    3     0.0000000000  0.7504758  0.03841005  0.019694190  0.13630576
##    3     0.0001000000  0.7542753  0.05715715  0.025722132  0.16056998
##    3     0.0003981072  0.7579290  0.06476973  0.030786456  0.17919834
##    3     0.0015848932  0.8115484  0.42399272  0.036509126  0.22309335
##    3     0.0063095734  0.8229521  0.50615594  0.029027276  0.12128644
##    3     0.0251188643  0.8350146  0.53204925  0.024670362  0.11563586
##    3     0.1000000000  0.8425768  0.55520864  0.016907640  0.06016095
##    5     0.0000000000  0.7660636  0.14779920  0.036289847  0.24238780
##    5     0.0001000000  0.7605017  0.08694250  0.033170388  0.20362371
##    5     0.0003981072  0.7800041  0.23345380  0.042034539  0.26953914
##    5     0.0015848932  0.8121570  0.45407157  0.031293437  0.17961686
##    5     0.0063095734  0.8260822  0.48859251  0.026846869  0.15744928
##    5     0.0251188643  0.8419205  0.55967510  0.010066607  0.03500075
##    5     0.1000000000  0.8516179  0.58572884  0.015992192  0.04699520
##    7     0.0000000000  0.7655468  0.15849514  0.037909710  0.23870327
##    7     0.0001000000  0.7750976  0.17606577  0.042690543  0.26211896
##    7     0.0003981072  0.7843051  0.28608687  0.044955436  0.26203506
##    7     0.0015848932  0.8162509  0.44796260  0.030032912  0.18554099
##    7     0.0063095734  0.8198983  0.49361864  0.030186546  0.12998579
##    7     0.0251188643  0.8367292  0.53382336  0.018819741  0.07524849
##    7     0.1000000000  0.8455410  0.56411866  0.016852333  0.05587054
##    9     0.0000000000  0.7692181  0.15974305  0.037018666  0.23858981
##    9     0.0001000000  0.7598756  0.14547485  0.029052059  0.22170235
##    9     0.0003981072  0.8017602  0.41142822  0.036796678  0.20047571
##    9     0.0015848932  0.8240404  0.52746736  0.024556477  0.05974188
##    9     0.0063095734  0.8361181  0.55103697  0.015798859  0.04792126
##    9     0.0251188643  0.8403693  0.55151368  0.013438303  0.05679280
##    9     0.1000000000  0.8475980  0.56411062  0.017209914  0.06883504
##   11     0.0000000000  0.7709229  0.14604335  0.040956955  0.24193632
##   11     0.0001000000  0.7670807  0.16705545  0.037470945  0.25281966
##   11     0.0003981072  0.8184983  0.46426784  0.033270905  0.18290362
##   11     0.0015848932  0.8151695  0.46895028  0.035971365  0.18374522
##   11     0.0063095734  0.8361373  0.54534516  0.019475178  0.06703141
##   11     0.0251188643  0.8441478  0.57350151  0.014476948  0.03952897
##   11     0.1000000000  0.8507217  0.58033681  0.013032496  0.04666353
##   13     0.0000000000  0.7780947  0.21435987  0.041336371  0.26843367
##   13     0.0001000000  0.7938230  0.28039394  0.046619305  0.27986601
##   13     0.0003981072  0.8135129  0.45524429  0.039675590  0.19552725
##   13     0.0015848932  0.8260415  0.51288942  0.029941813  0.12358539
##   13     0.0063095734  0.8337643  0.52933634  0.024345729  0.12022306
##   13     0.0251188643  0.8383652  0.55543329  0.017423934  0.04413289
##   13     0.1000000000  0.8441666  0.56077268  0.009909212  0.03364321
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 5 and decay = 0.1.
## a 22-5-1 network with 121 weights
## options were - entropy fitting  decay=0.1
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   -0.46
## i18->h1 i19->h1 i20->h1 i21->h1 i22->h1
```

```
##    0.00   -0.09   -0.55    0.00   -0.01
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##   -1.06    1.71   -0.73    0.72    0.29   -0.46    0.18    0.86   -0.83
##  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##    0.08   -2.41    0.40    0.71    0.14    0.61   -0.83    0.08    0.04
## i18->h2 i19->h2 i20->h2 i21->h2 i22->h2
##   -5.93    0.01   -0.01   -0.78    0.02
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
## i18->h3 i19->h3 i20->h3 i21->h3 i22->h3
##    0.00    0.00    0.01    0.00    0.00
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4
##    0.25    1.27   -0.81    2.02   -0.87   -1.42    0.94    0.62    0.54
##  i9->h4 i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4
##    0.15    0.69    0.40    0.11    0.10   -0.92   -0.08    0.15    0.05
## i18->h4 i19->h4 i20->h4 i21->h4 i22->h4
##   -0.52    0.01   -0.06    0.28    0.34
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5
##    3.06    0.37   -0.98   -0.82    0.74    0.72    0.86    0.08    0.28
##  i9->h5 i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5
##    0.06    0.12    0.47    0.82    0.06    1.43    1.21   -0.29    0.02
## i18->h5 i19->h5 i20->h5 i21->h5 i22->h5
##    0.09    0.00   -0.02   -0.47    0.04
##  b->o h1->o h2->o h3->o h4->o h5->o
##  0.62 -0.10 -5.07  0.62  0.84  3.52
## nnet variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                    Overall
## last_fico_range_high              100.000
## fico_range_high                    58.185
## inq_last_6mths                     30.982
## verification_statusVerified        22.601
## purposesmall_business              19.692
## term 60 months                     18.295
## purposedebt_consolidation          16.286
## verification_statusSource Verified  15.692
## purposemedical                     15.439
## purposevacation                    12.989
## purposeeducational                 12.832
## purposecredit_card                 11.903
## purposeother                        9.341
## purposehouse                        8.264
## desc_empty1                         7.847
## revol_util                          7.217
## purposehome_improvement             7.181
## purposemoving                       6.918
## dti                                 1.955
## purposewedding                      1.941
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction good bad
##       good  908 115
##       bad    68 215
##
##                Accuracy : 0.8599
##                  95% CI : (0.8399, 0.8783)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6106
##  Mcnemar's Test P-Value : 0.0006728
##
##             Sensitivity : 0.6515
##             Specificity : 0.9303
##          Pos Pred Value : 0.7597
##          Neg Pred Value : 0.8876
##              Prevalence : 0.2527
##          Detection Rate : 0.1646
##    Detection Prevalence : 0.2167
##       Balanced Accuracy : 0.7909
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
```

```
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.8962
```

Results for Grade D Loans

Approximately 35% of the Grade D loans in this dataset went bad. With the four models, we were able to correctly predict between 63% and 72% of the bad loans. This predictive ability is based on a 50% probability classification cutoff. As the ROC curves show, it's possible to predict the bad loans with a higher probability, of course, with a higher false positive rate, though. The FICO range and the number of inquiries in the past 6 months were also important predictors for this loan grade.

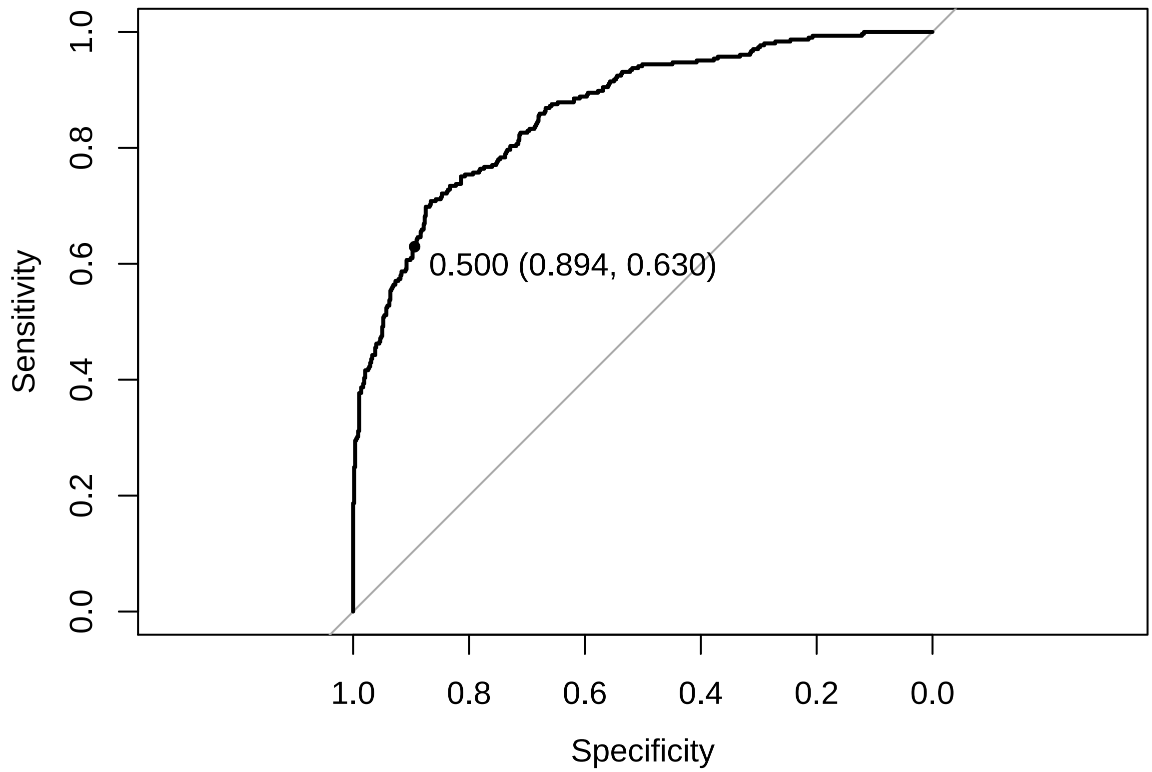
Logistic Regression Model

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
## Generalized Linear Model
##
## 2643 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 2643, 2643, 2643, 2643, 2643, 2643, ...
##
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.8159264  0.5783075  0.01108971   0.02696422
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4448  -0.6232  -0.3383   0.5243   3.2461
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            5.501408   2.048027   2.686 0.007227
## `term 60 months`                       0.031229   0.144886   0.216 0.829347
## `verification_statusSource Verified` -0.303380   0.157797  -1.923 0.054531
## verification_statusVerified          -0.234582   0.128998  -1.818 0.068988
## purposecredit_card                     1.540008   0.449611   3.425 0.000614
## purposedebt_consolidation              1.071763   0.425870   2.517 0.011848
## purposeeducational                     2.204750   0.585702   3.764 0.000167
## purposehome_improvement                0.587496   0.476104   1.234 0.217216
## purposehouse                           0.908325   0.770871   1.178 0.238673
```

```
## purposemajor_purchase                1.051006   0.476386   2.206 0.027370
## purposemedical                       1.488440   0.618274   2.407 0.016066
## purposemoving                        1.068828   0.618778   1.727 0.084110
## purposeother                         1.391355   0.448153   3.105 0.001905
## purposerenewable_energy              0.108553   1.211484   0.090 0.928602
## purposesmall_business                1.014746   0.477539   2.125 0.033591
## purposevacation                      0.295210   0.790210   0.374 0.708713
## purposewedding                       1.005743   0.536424   1.875 0.060806
## fico_range_high                      0.002314   0.002961   0.781 0.434579
## inq_last_6mths                       0.736047   0.036497  20.167  < 2e-16
## revol_util                          -0.004724   0.002204  -2.143 0.032096
## last_fico_range_high                -0.015055   0.000769 -19.577  < 2e-16
## desc_empty1                         -0.552767   0.140638  -3.930 8.48e-05
## dti                                  0.007284   0.008688   0.838 0.401778
##
## (Intercept)                         **
## `term 60 months`
## `verification_statusSource Verified` .
## verification_statusVerified         .
## purposecredit_card                  ***
## purposedebt_consolidation           *
## purposeeducational                  ***
## purposehome_improvement
## purposehouse
## purposemajor_purchase               *
## purposemedical                      *
## purposemoving                       .
## purposeother                        **
## purposerenewable_energy
## purposesmall_business               *
## purposevacation
## purposewedding                      .
## fico_range_high
## inq_last_6mths                      ***
## revol_util                          *
## last_fico_range_high                ***
## desc_empty1                         ***
## dti
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3409.8  on 2642  degrees of freedom
## Residual deviance: 2162.1  on 2620  degrees of freedom
## AIC: 2208.1
##
## Number of Fisher Scoring iterations: 5
##
## glm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                     Overall
```

```
## inq_last_6mths                        100.000
## last_fico_range_high                   97.061
## desc_empty1                            19.130
## purposeeducational                     18.302
## purposecredit_card                     16.613
## purposeother                           15.017
## purposedebt_consolidation              12.088
## purposemedical                         11.544
## purposemajor_purchase                  10.542
## revol_util                             10.228
## purposesmall_business                  10.137
## `verification_statusSource Verified`    9.129
## purposewedding                          8.892
## verification_statusVerified             8.611
## purposemoving                           8.157
## purposehome_improvement                 5.700
## purposehouse                            5.422
## dti                                     3.730
## fico_range_high                         3.446
## purposevacation                         1.414
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  514 113
##       bad    61 192
##
##                Accuracy : 0.8023
##                  95% CI : (0.7744, 0.8281)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5452
##  Mcnemar's Test P-Value : 0.0001105
##
##             Sensitivity : 0.6295
##             Specificity : 0.8939
##          Pos Pred Value : 0.7589
##          Neg Pred Value : 0.8198
##              Prevalence : 0.3466
##          Detection Rate : 0.2182
##    Detection Prevalence : 0.2875
##       Balanced Accuracy : 0.7617
##
##        'Positive' Class : bad
##
```



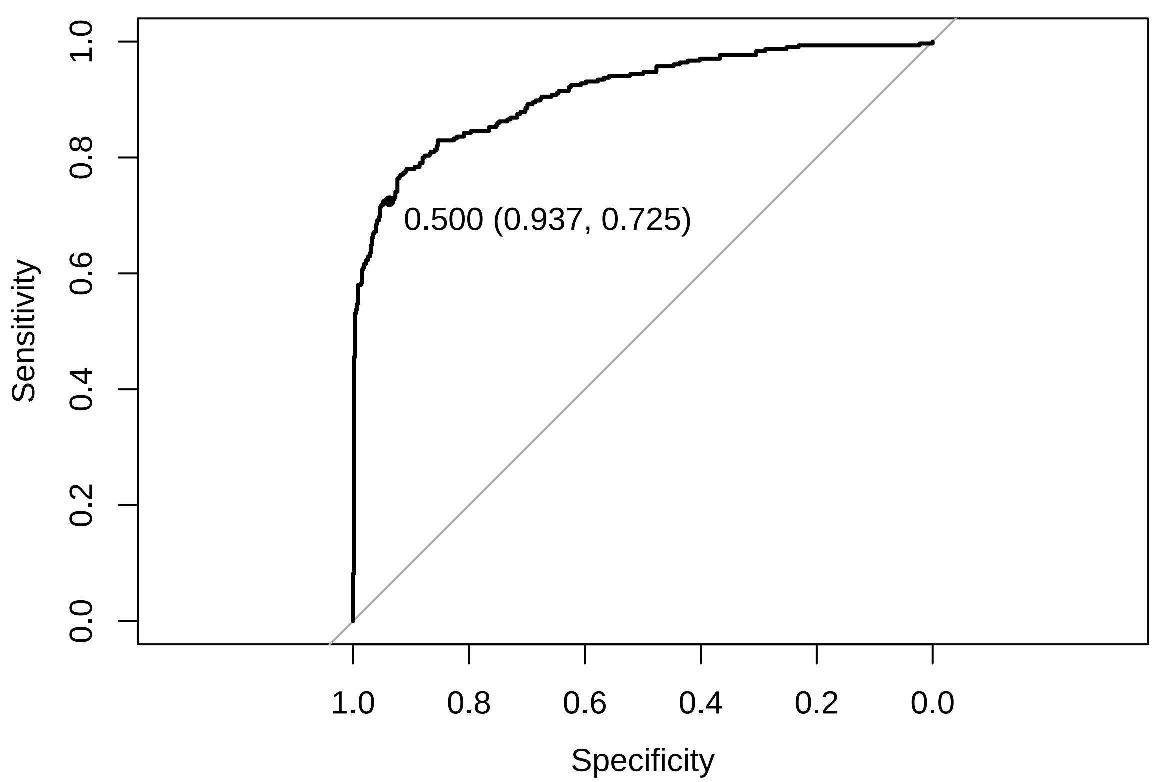

```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.8635
```

Random Forest Model

```
## Random Forest
##
## 2643 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 2643, 2643, 2643, 2643, 2643, 2643, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.8374058  0.6045573  0.011991157  0.02918795
##   12    0.8612697  0.6835699  0.009242641  0.02288577
##   22    0.8562889  0.6729565  0.009451313  0.02312815
##
## Accuracy was used to select the optimal model using  the largest value.
```

```
## The final value used for the model was mtry = 12.
##                 Length Class      Mode
## call               4   -none-     call
## type               1   -none-     character
## predicted       2643   factor     numeric
## err.rate        1500   -none-     numeric
## confusion          6   -none-     numeric
## votes           5286   matrix     numeric
## oob.times       2643   -none-     numeric
## classes            2   -none-     character
## importance        22   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               2643   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            22   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
## rf variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                    Overall
## inq_last_6mths                    100.0000
## last_fico_range_high               76.9173
## fico_range_high                    37.0313
## dti                                33.6802
## revol_util                         29.6818
## term 60 months                      4.1432
## verification_statusVerified         3.4264
## purposedebt_consolidation           3.2923
## desc_empty1                         3.0581
## verification_statusSource Verified   2.8529
## purposecredit_card                  2.4508
## purposeother                        2.1273
## purposesmall_business               1.7122
## purposemajor_purchase               1.1920
## purposehome_improvement             1.1585
## purposeeducational                  1.0301
## purposehouse                        0.8826
## purposewedding                      0.7868
## purposemedical                      0.7425
## purposemoving                       0.7135
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  540  84
```

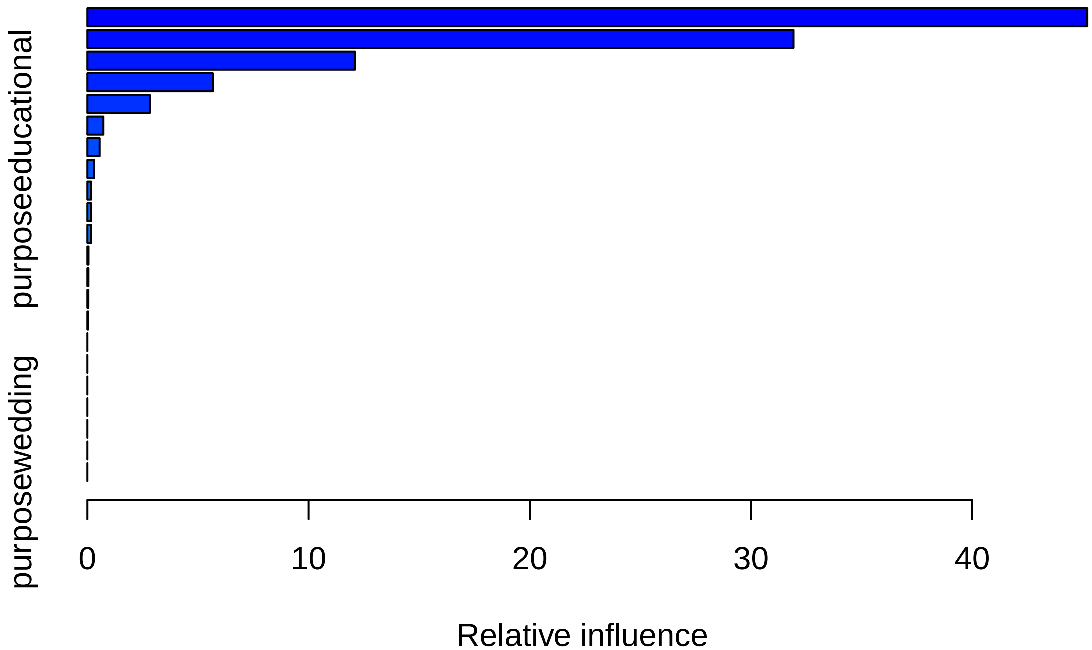
```
##       bad    35 221
##
##                Accuracy : 0.8648
##                  95% CI : (0.8404, 0.8867)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6897
##  Mcnemar's Test P-Value : 1.082e-05
##
##             Sensitivity : 0.7246
##             Specificity : 0.9391
##          Pos Pred Value : 0.8633
##          Neg Pred Value : 0.8654
##              Prevalence : 0.3466
##          Detection Rate : 0.2511
##    Detection Prevalence : 0.2909
##       Balanced Accuracy : 0.8319
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.9118
```

Gradient Boost Model

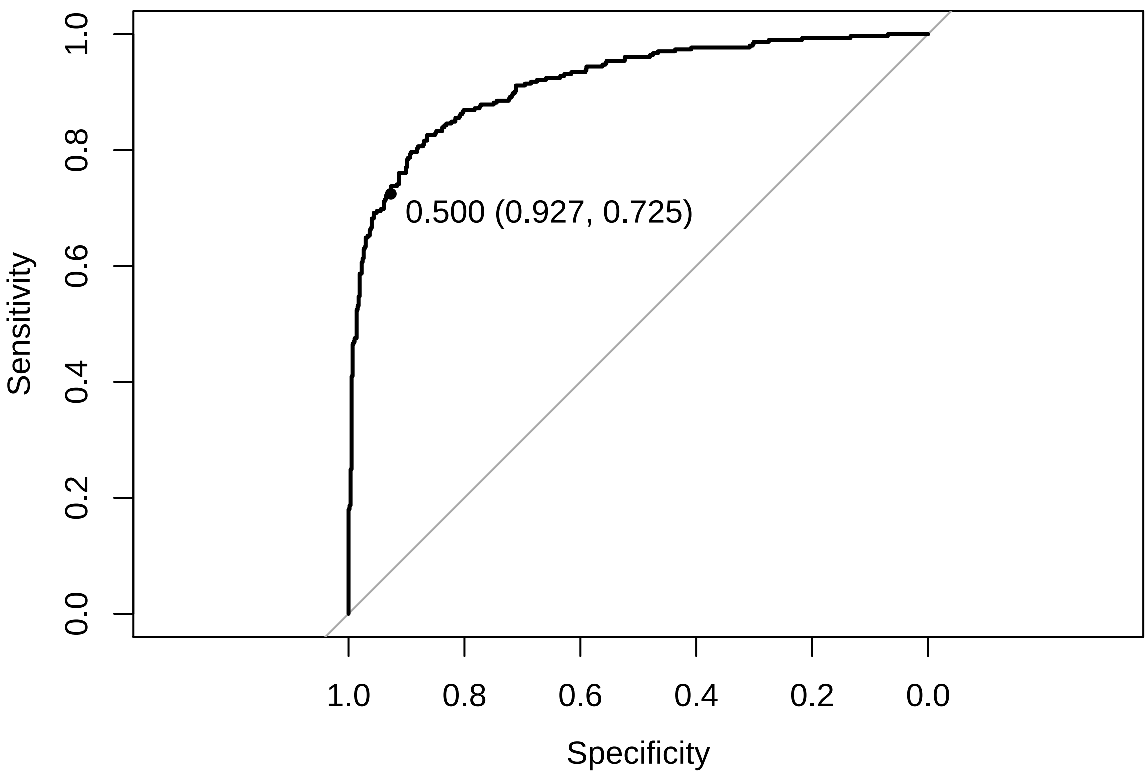
```
## Stochastic Gradient Boosting
##
## 2643 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 2643, 2643, 2643, 2643, 2643, 2643, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.8581440  0.6702251  0.009157259
##   1                  100      0.8650549  0.6893528  0.008899009
##   1                  150      0.8653912  0.6904454  0.010640733
##   2                   50      0.8646848  0.6884431  0.009255246
##   2                  100      0.8661950  0.6925453  0.011711520
##   2                  150      0.8655130  0.6910937  0.009151581
##   3                   50      0.8658846  0.6918635  0.010099111
##   3                  100      0.8672241  0.6954752  0.009976149
##   3                  150      0.8667525  0.6942425  0.009698905
##   Kappa SD
##   0.02081322
##   0.02021083
##   0.02399457
##   0.02089844
##   0.02605261
##   0.02129935
##   0.02230646
##   0.02231161
##   0.02159582
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 100,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```




```
##                                                                  var
## inq_last_6mths                                        inq_last_6mths
## last_fico_range_high                            last_fico_range_high
## fico_range_high                                      fico_range_high
## dti                                                              dti
## revol_util                                                revol_util
## desc_empty1                                              desc_empty1
## term 60 months                                        term 60 months
## purposeeducational                                purposeeducational
## purposehome_improvement                      purposehome_improvement
## verification_statusSource Verified verification_statusSource Verified
## verification_statusVerified              verification_statusVerified
## purposemajor_purchase                          purposemajor_purchase
## purposehouse                                            purposehouse
## purposeother                                            purposeother
## purposemoving                                          purposemoving
## purposecredit_card                                purposecredit_card
## purposedebt_consolidation                  purposedebt_consolidation
## purposemedical                                        purposemedical
## purposerenewable_energy                      purposerenewable_energy
## purposesmall_business                          purposesmall_business
## purposevacation                                      purposevacation
## purposewedding                                        purposewedding
##                                         rel.inf
## inq_last_6mths                      45.20451882
## last_fico_range_high                31.92307071
## fico_range_high                     12.10111351
## dti                                  5.67049714
## revol_util                           2.82244497
## desc_empty1                          0.72368753
## term 60 months                       0.55404397
## purposeeducational                   0.30679976
## purposehome_improvement              0.17216340
```

```
## verification_statusSource Verified  0.17006458
## verification_statusVerified         0.16968559
## purposemajor_purchase               0.05011704
## purposehouse                        0.04708456
## purposeother                        0.04368752
## purposemoving                       0.04102090
## purposecredit_card                  0.00000000
## purposedebt_consolidation           0.00000000
## purposemedical                      0.00000000
## purposerenewable_energy             0.00000000
## purposesmall_business               0.00000000
## purposevacation                     0.00000000
## purposewedding                      0.00000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                      Overall
## inq_last_6mths                     100.00000
## last_fico_range_high                70.61920
## fico_range_high                     26.76970
## dti                                 12.54409
## revol_util                           6.24372
## desc_empty1                          1.60092
## term 60 months                       1.22564
## purposeeducational                   0.67869
## purposehome_improvement              0.38085
## verification_statusSource Verified   0.37621
## verification_statusVerified          0.37537
## purposemajor_purchase                0.11087
## purposehouse                         0.10416
## purposeother                         0.09664
## purposemoving                        0.09075
## purposewedding                       0.00000
## purposedebt_consolidation            0.00000
## purposerenewable_energy              0.00000
## purposevacation                      0.00000
## purposecredit_card                   0.00000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  533  84
##       bad    42 221
##
##                Accuracy : 0.8568
##                  95% CI : (0.8319, 0.8793)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6733
##  Mcnemar's Test P-Value : 0.0002596
##
##             Sensitivity : 0.7246
```

```
##             Specificity : 0.9270
##          Pos Pred Value : 0.8403
##          Neg Pred Value : 0.8639
##              Prevalence : 0.3466
##          Detection Rate : 0.2511
##    Detection Prevalence : 0.2989
##       Balanced Accuracy : 0.8258
##
##        'Positive' Class : bad
##
```

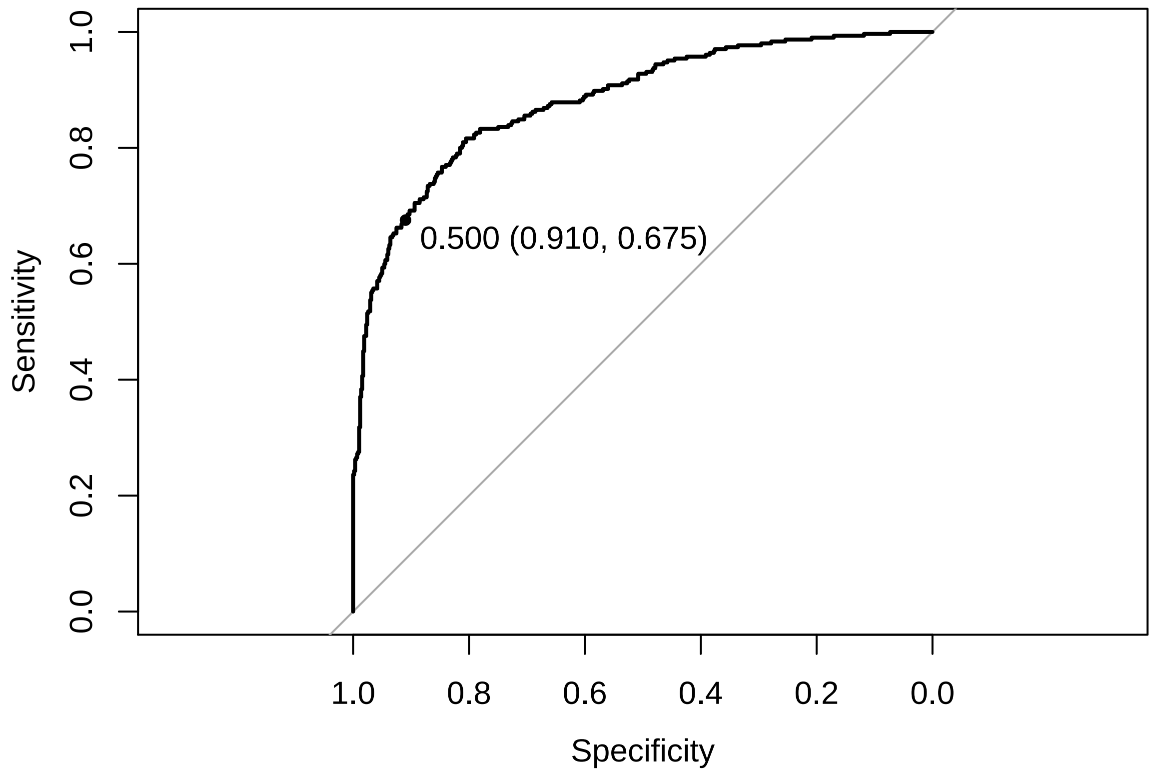



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.9161
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 2643 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
```

```
##
## Summary of sample sizes: 2378, 2379, 2379, 2378, 2379, 2378, ...
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.8191562  0.5922916  0.01362757   0.03096520
##   0.50  0.8244549  0.6018029  0.01293071   0.02812489
##   1.00  0.8289947  0.6101319  0.01460652   0.03098270
##
## Tuning parameter 'sigma' was held constant at a value of 0.05131291
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.05131291 and C = 1.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                      Importance
## last_fico_range_high    100.000
## inq_last_6mths           94.764
## verification_status      36.507
## revol_util               35.149
## fico_range_high          29.454
## purpose                  27.960
## dti                      15.469
## term                      5.221
## desc_empty                0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  523  99
##       bad    52 206
##
##                Accuracy : 0.8284
##                  95% CI : (0.8018, 0.8528)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6069
##  Mcnemar's Test P-Value : 0.0001815
##
##             Sensitivity : 0.6754
##             Specificity : 0.9096
##          Pos Pred Value : 0.7984
##          Neg Pred Value : 0.8408
##              Prevalence : 0.3466
##          Detection Rate : 0.2341
##    Detection Prevalence : 0.2932
##       Balanced Accuracy : 0.7925
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.8819
```

Neural Net Model

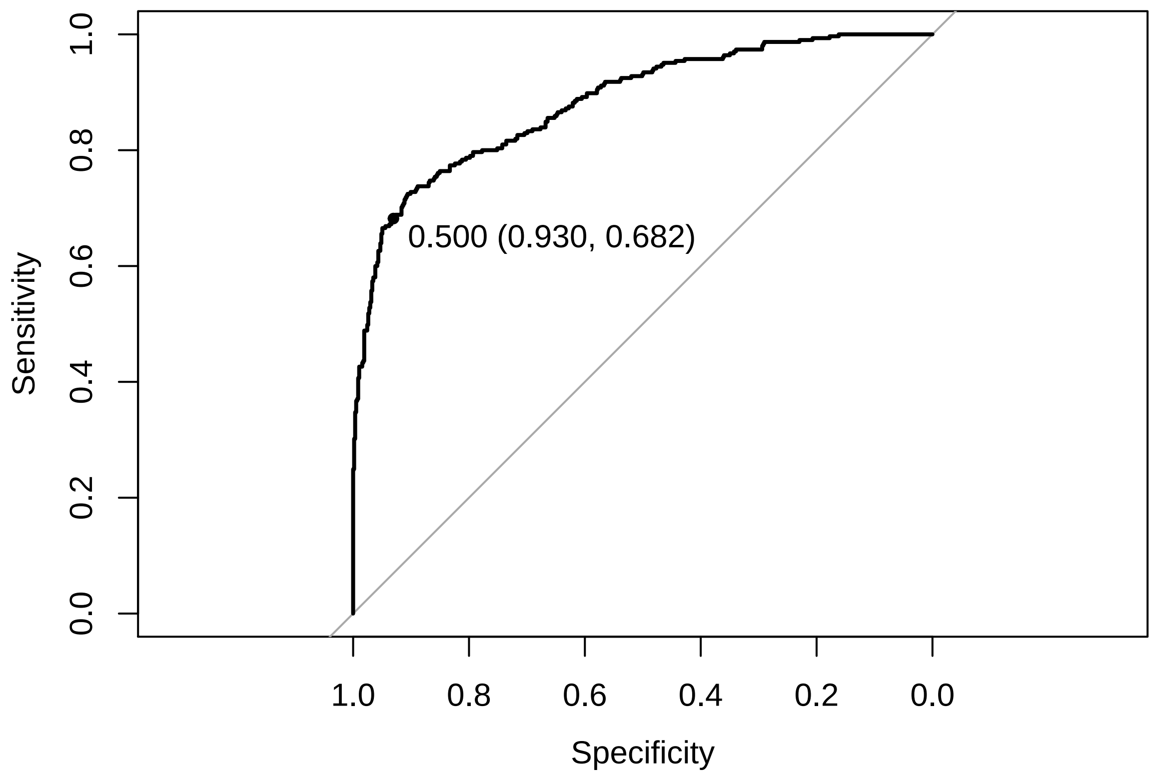
```
## Neural Network
##
## 2643 samples
##    9 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 2643, 2643, 2643, 2643, 2643, 2643, ...
##
## Resampling results across tuning parameters:
##
##   size  decay         Accuracy   Kappa       Accuracy SD  Kappa SD
##    1    0.0000000000  0.6608212  0.02297872  0.033131709  0.11489362
##    1    0.0001000000  0.6542089  0.00000000  0.010237951  0.00000000
##    1    0.0003981072  0.6607991  0.02315191  0.032055731  0.11575956
##    1    0.0015848932  0.7351593  0.31230430  0.078216664  0.27293579
##    1    0.0063095734  0.7810797  0.45546454  0.066117091  0.24075815
```

```
##    1    0.0251188643  0.7963797  0.51140522  0.057220564  0.19387184
##    1    0.1000000000  0.8176125  0.58250881  0.010982622  0.02613322
##    3    0.0000000000  0.6843452  0.11179434  0.061749842  0.22835146
##    3    0.0001000000  0.6718943  0.06229193  0.048945151  0.17622914
##    3    0.0003981072  0.6735273  0.07144137  0.055814410  0.19752026
##    3    0.0015848932  0.8054942  0.54817665  0.035274080  0.12042671
##    3    0.0063095734  0.8226783  0.59785586  0.008817661  0.01863438
##    3    0.0251188643  0.8156144  0.57803881  0.018203943  0.04858236
##    3    0.1000000000  0.8255058  0.59942086  0.013722020  0.03501555
##    5    0.0000000000  0.6726095  0.06887764  0.055663007  0.19072274
##    5    0.0001000000  0.6938557  0.13888331  0.071625723  0.25379043
##    5    0.0003981072  0.7551729  0.36655555  0.079935142  0.28425724
##    5    0.0015848932  0.8099518  0.56608612  0.021127331  0.05508887
##    5    0.0063095734  0.8116609  0.57434781  0.019267615  0.04200101
##    5    0.0251188643  0.8237850  0.59666986  0.015138442  0.03916782
##    5    0.1000000000  0.8299409  0.61017424  0.015504732  0.03415607
##    7    0.0000000000  0.7116283  0.20908311  0.079428483  0.28472185
##    7    0.0001000000  0.7024222  0.19863874  0.069737195  0.25701083
##    7    0.0003981072  0.8009813  0.53354214  0.037907846  0.13207528
##    7    0.0015848932  0.8217919  0.59553709  0.015546729  0.03324877
##    7    0.0063095734  0.8184624  0.58652757  0.011553725  0.03371530
##    7    0.0251188643  0.8254082  0.60055145  0.013519639  0.03079503
##    7    0.1000000000  0.8253754  0.60017387  0.011485701  0.02806951
##    9    0.0000000000  0.6906818  0.13570090  0.065975984  0.23541930
##    9    0.0001000000  0.7061964  0.19637434  0.069785477  0.25251584
##    9    0.0003981072  0.8000311  0.53852348  0.038190640  0.10190894
##    9    0.0015848932  0.8120850  0.56987160  0.017638307  0.05788853
##    9    0.0063095734  0.8170792  0.58458108  0.011968506  0.03404369
##    9    0.0251188643  0.8198963  0.58800971  0.014103969  0.03940512
##    9    0.1000000000  0.8279361  0.60576428  0.011862111  0.02474368
##   11    0.0000000000  0.7267474  0.26680225  0.074721733  0.28613589
##   11    0.0001000000  0.7272073  0.27741540  0.077902577  0.27619485
##   11    0.0003981072  0.8104718  0.55403182  0.037584472  0.12921956
##   11    0.0015848932  0.8154144  0.58373251  0.013140144  0.02823739
##   11    0.0063095734  0.8147245  0.56910863  0.036896362  0.12243163
##   11    0.0251188643  0.8259268  0.60308682  0.012611759  0.02466108
##   11    0.1000000000  0.8316676  0.61453003  0.014475660  0.03272543
##   13    0.0000000000  0.7410831  0.32170991  0.076978553  0.27850489
##   13    0.0001000000  0.7115532  0.22384341  0.072747696  0.26963896
##   13    0.0003981072  0.8098317  0.56564720  0.024981739  0.07303986
##   13    0.0015848932  0.8177232  0.58097845  0.020155846  0.05882777
##   13    0.0063095734  0.8173901  0.58028145  0.022980099  0.06096097
##   13    0.0251188643  0.8229170  0.59985656  0.024128700  0.04437668
##   13    0.1000000000  0.8295168  0.60878769  0.016410802  0.03607164
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 11 and decay = 0.1.
## a 22-11-1 network with 265 weights
## options were - entropy fitting  decay=0.1
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##    0.00    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.01
##  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
##    0.01    0.00    0.01    0.00   -0.01    0.00    0.00    0.00    0.05
## i18->h1 i19->h1 i20->h1 i21->h1 i22->h1
```

```
##    0.01   -0.01    0.11   -0.01   -0.01
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##   -0.29    0.85   -0.31   -0.74   -0.33   -0.23    0.44   -0.74    0.36
##  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##    1.21   -0.18   -0.06   -1.67   -0.05    0.68    0.05    0.02    0.03
## i18->h2 i19->h2 i20->h2 i21->h2 i22->h2
##   -3.89    0.02    0.00    0.81   -0.12
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##    0.01   -0.01    0.01    0.00    0.01    0.01   -0.01    0.00    0.00
##  i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##    0.00   -0.01    0.00    0.00    0.00    0.00   -0.01    0.00   -0.68
## i18->h3 i19->h3 i20->h3 i21->h3 i22->h3
##   -0.01   -0.07   -0.55    0.00   -0.01
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4
##    0.00    0.00    0.01    0.01   -0.01    0.00   -0.01   -0.01    0.00
##  i9->h4 i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4
##    0.00    0.01    0.01   -0.01    0.01    0.01   -0.01    0.00    0.01
## i18->h4 i19->h4 i20->h4 i21->h4 i22->h4
##   -0.01    0.00   -0.01    0.01    0.00
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5
##    0.02   -0.24   -0.49    0.38    0.25   -0.28   -0.05   -0.09    0.03
##  i9->h5 i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5
##   -0.23    0.02    0.04    0.68    0.01   -0.15   -0.02   -0.08   -0.01
## i18->h5 i19->h5 i20->h5 i21->h5 i22->h5
##   -1.13   -0.05    0.03   -0.18    0.15
##   b->h6  i1->h6  i2->h6  i3->h6  i4->h6  i5->h6  i6->h6  i7->h6  i8->h6
##    2.85    0.37   -0.84   -0.55    0.94    1.06    2.46    0.50    1.78
##  i9->h6 i10->h6 i11->h6 i12->h6 i13->h6 i14->h6 i15->h6 i16->h6 i17->h6
##    0.05    1.44    1.32    1.50   -0.95    0.29   -0.11    1.21    0.01
## i18->h6 i19->h6 i20->h6 i21->h6 i22->h6
##    0.70    0.00   -0.02   -1.10   -0.05
##   b->h7  i1->h7  i2->h7  i3->h7  i4->h7  i5->h7  i6->h7  i7->h7  i8->h7
##    0.00   -0.01    0.00    0.01    0.00    0.01    0.00    0.00   -0.01
##  i9->h7 i10->h7 i11->h7 i12->h7 i13->h7 i14->h7 i15->h7 i16->h7 i17->h7
##    0.00   -0.01    0.00    0.00   -0.01   -0.01    0.00    0.00   -0.14
## i18->h7 i19->h7 i20->h7 i21->h7 i22->h7
##    0.00   -0.02   -0.13    0.00    0.00
##   b->h8  i1->h8  i2->h8  i3->h8  i4->h8  i5->h8  i6->h8  i7->h8  i8->h8
##    0.00    0.00    0.00    0.00    0.00    0.00    0.01    0.00    0.00
##  i9->h8 i10->h8 i11->h8 i12->h8 i13->h8 i14->h8 i15->h8 i16->h8 i17->h8
##    0.01    0.00    0.00    0.01   -0.01    0.00   -0.01   -0.01    0.49
## i18->h8 i19->h8 i20->h8 i21->h8 i22->h8
##    0.00    0.06    0.41    0.00    0.01
##   b->h9  i1->h9  i2->h9  i3->h9  i4->h9  i5->h9  i6->h9  i7->h9  i8->h9
##    0.07   -3.73   -1.23    1.17   -1.81    0.82   -0.11    2.16    0.05
##  i9->h9 i10->h9 i11->h9 i12->h9 i13->h9 i14->h9 i15->h9 i16->h9 i17->h9
##    0.81   -0.44   -0.59   -0.28    0.08   -0.09   -0.16   -0.50   -0.05
## i18->h9 i19->h9 i20->h9 i21->h9 i22->h9
##    2.33    0.20    0.07    0.90   -0.91
##   b->h10  i1->h10  i2->h10  i3->h10  i4->h10  i5->h10  i6->h10  i7->h10
##     0.01     0.00     0.00     0.00     0.00     0.00     0.00    -0.01
##  i8->h10  i9->h10 i10->h10 i11->h10 i12->h10 i13->h10 i14->h10 i15->h10
##     0.00     0.01     0.00     0.00     0.01     0.00     0.01     0.00
## i16->h10 i17->h10 i18->h10 i19->h10 i20->h10 i21->h10 i22->h10
```

```
##     0.01     0.28    -0.01     0.05     0.26     0.00     0.01
##   b->h11  i1->h11  i2->h11  i3->h11  i4->h11  i5->h11  i6->h11  i7->h11
##     0.03    -0.35     0.31    -0.93    -0.11     0.52    -0.08    -0.12
##  i8->h11  i9->h11 i10->h11 i11->h11 i12->h11 i13->h11 i14->h11 i15->h11
##     0.07     0.03    -0.03    -0.30     0.22     0.00    -0.25     0.05
## i16->h11 i17->h11 i18->h11 i19->h11 i20->h11 i21->h11 i22->h11
##     0.02    -1.38    -1.86    -0.65     1.83    -0.17    -0.45
##   b->o  h1->o  h2->o  h3->o  h4->o  h5->o  h6->o  h7->o  h8->o  h9->o
##   1.15   0.15  -3.58  -0.19   1.15  -0.40   3.01  -0.50   0.15  -1.45
## h10->o h11->o
##   1.14  -0.82
## nnet variable importance
##
##   only 20 most important variables shown (out of 22)
##
##                                    Overall
## fico_range_high                    100.000
## last_fico_range_high               100.000
## inq_last_6mths                      43.816
## purposeother                        18.177
## verification_statusVerified         16.753
## term 60 months                      16.009
## revol_util                          12.319
## verification_statusSource Verified  11.409
## desc_empty1                         10.715
## purposehome_improvement             10.039
## purposecredit_card                   9.580
## purposeeducational                   9.503
## purposedebt_consolidation            8.991
## purposemajor_purchase                7.764
## purposesmall_business                7.394
## purposemoving                        7.089
## purposemedical                       6.681
## purposehouse                         6.047
## dti                                  6.027
## purposerenewable_energy              4.280
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  535  97
##       bad    40 208
##
##                Accuracy : 0.8443
##                  95% CI : (0.8186, 0.8677)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6405
##  Mcnemar's Test P-Value : 1.715e-06
##
##             Sensitivity : 0.6820
##             Specificity : 0.9304
##          Pos Pred Value : 0.8387
```

```
##          Neg Pred Value : 0.8465
##              Prevalence : 0.3466
##          Detection Rate : 0.2364
##    Detection Prevalence : 0.2818
##       Balanced Accuracy : 0.8062
##
##        'Positive' Class : bad
##
```




```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.8833
```