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 approximately 60% of the loans that will go bad. Also for these same loan grades, all five of the models identified the number of credit inquiries in the past six months as one of the top two predictors that were most important in predicting which loans will go bad; FICO score was also identified as important in many models.

Details on these and other results are shown below.

Data Ingest and Initialization Steps

```r
# 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 438
##  $ member_id                  : int  1296599 1314167 1313524 1277178 1311748 1311441 1304742 1288
##  $ 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 2
##  $ emp_title                  : Factor w/ 30661 levels "","  old palm inc",..: 1 22922 1 791 2823
##  $ 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/loar
##  $ 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 17
##  $ zip_code                   : Factor w/ 838 levels "","007xx","010xx",..: 728 282 514 765 814 7
##  $ 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
# last_fico_range_high         (this data is not available at time of loan origination so can't be used
# last_fico_range_low          (this data is not available at time of loan origination so can't be used
```

```
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, last_fico_range_high,
    last_fico_range_low))

# 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",
```

```r
                colour = "status"), data = dft)
            p <- p + geom_density() + xlab(varname)
            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)
}


# Based on exploratory plots, 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, 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, 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, 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, desc_empty, dti))

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

```r
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 + 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
# therefore, I won't use fico_range_low 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, desc_empty,
    dti))
b_loans <- select(b_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, desc_empty,
    dti))
c_loans <- select(c_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, desc_empty,
    dti))
d_loans <- select(d_loans, c(status, term, verification_status,
    purpose, fico_range_high, inq_last_6mths, revol_util, 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. We use the default probability threshold of 0.5 for the classification threshold used for the confusion matrix, but we use a probability of threshold of 0.25 for the annotated point on the ROC curves.

```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))
}

# function to create logistic regression model
```

```r
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 = 0.25)

}

# 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 = 0.25)

}

# 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 = 0.25)

}

# 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",
```

```r
        "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 = 0.25)

}

# 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 = 0.25)

}
```

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 4%. 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 with several of the models.

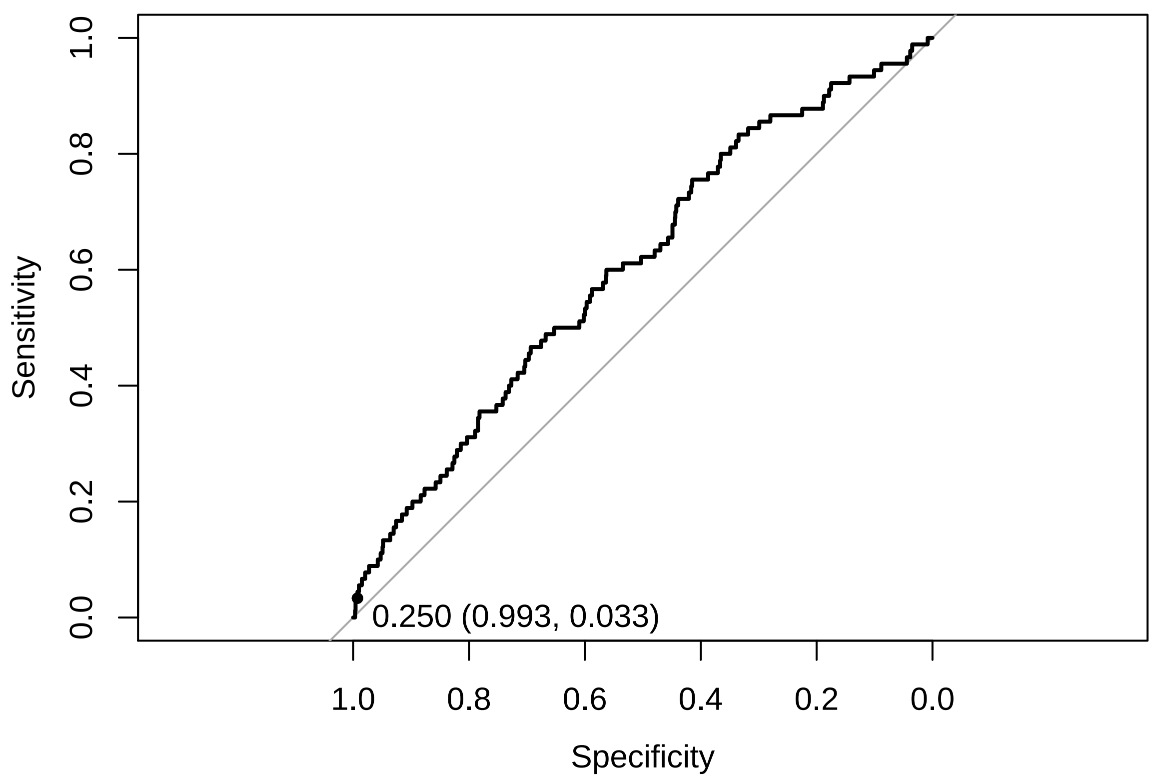
Logistic Regression Model

```
## Generalized Linear Model
##
## 3879 samples
##    8 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
##   0.9304955  0.02040338  0.004413853  0.01575431
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1539  -0.4195  -0.3278  -0.2471   3.0592
##
## Coefficients:
##                                        Estimate Std. Error z value
## (Intercept)                          12.4954171  2.6459265   4.723
## `term 60 months`                      0.5159779  0.2558082   2.017
## `verification_statusSource Verified` -0.1860183  0.1778546  -1.046
## verification_statusVerified           0.0034605  0.1589239   0.022
## purposecredit_card                   -0.6061872  0.3166867  -1.914
## purposedebt_consolidation            -0.2054820  0.2694664  -0.763
## purposeeducational                    0.6298458  0.4407702   1.429
## purposehome_improvement              -0.2021463  0.3404283  -0.594
## purposehouse                         -0.3041684  0.6617453  -0.460
## purposemajor_purchase                -0.5161738  0.3637044  -1.419
## purposemedical                        0.6188291  0.4414318   1.402
## purposemoving                         0.1320618  0.4733438   0.279
## purposeother                          0.0502626  0.2989923   0.168
## purposerenewable_energy               1.0051113  1.2354166   0.814
## purposesmall_business                 0.3741562  0.3816558   0.980
## purposevacation                       0.5583712  0.5110992   1.092
## purposewedding                       -0.9676995  0.6411342  -1.509
## fico_range_high                      -0.0203607  0.0034342  -5.929
## inq_last_6mths                        0.3378314  0.0487287   6.933
## revol_util                            0.0030583  0.0034542   0.885
## desc_empty1                          -0.1192349  0.1529052  -0.780
## dti                                   0.0009402  0.0102881   0.091
##                                      Pr(>|z|)
## (Intercept)                          2.33e-06 ***
## `term 60 months`                       0.0437 *
## `verification_statusSource Verified`   0.2956
## verification_statusVerified            0.9826
## purposecredit_card                     0.0556 .
## purposedebt_consolidation              0.4457
## purposeeducational                     0.1530
## purposehome_improvement                0.5526
## purposehouse                           0.6458
## purposemajor_purchase                  0.1558
```

```
## purposemedical                          0.1610
## purposemoving                           0.7802
## purposeother                            0.8665
## purposerenewable_energy                 0.4159
## purposesmall_business                   0.3269
## purposevacation                         0.2746
## purposewedding                          0.1312
## fico_range_high                       3.05e-09 ***
## inq_last_6mths                        4.12e-12 ***
## revol_util                              0.3760
## desc_empty1                             0.4355
## dti                                     0.9272
## ---
## 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: 1844.0  on 3857  degrees of freedom
## AIC: 1888
##
## Number of Fisher Scoring iterations: 6
##
## glm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                       Overall
## inq_last_6mths                        100.000
## fico_range_high                        85.472
## `term 60 months`                       28.870
## purposecredit_card                     27.382
## purposewedding                         21.524
## purposeeducational                     20.361
## purposemajor_purchase                  20.220
## purposemedical                         19.969
## purposevacation                        15.493
## `verification_statusSource Verified`   14.818
## purposesmall_business                  13.870
## revol_util                             12.496
## purposerenewable_energy                11.457
## desc_empty1                            10.968
## purposedebt_consolidation              10.719
## purposehome_improvement                 8.277
## purposehouse                            6.336
## purposemoving                           3.722
## purposeother                            2.117
## dti                                     1.007
## 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
##
```

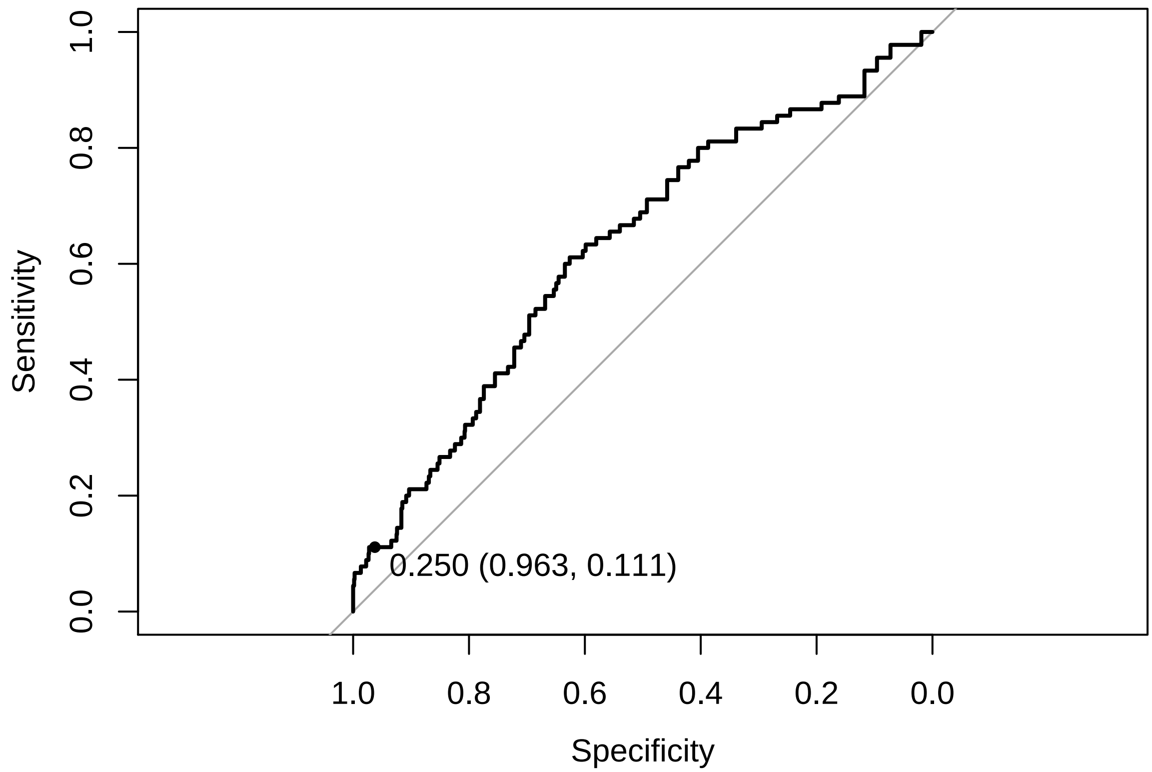



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.6048
```

Random Forest Model

```
## Random Forest
##
## 3879 samples
##    8 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    0.9286049  0.0000000  0.005863649  0.00000000
##   11    0.9303285  0.1316785  0.005751755  0.03414599
##   21    0.9271217  0.1296766  0.005591153  0.03380578
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 11.
##                 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        21   -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            21   -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 21)
##
##                                     Overall
## revol_util                         100.0000
## dti                                 98.6483
## fico_range_high                     48.9139
## inq_last_6mths                      28.7582
```

```
## verification_statusVerified          8.8902
## desc_empty1                          7.8446
## purposedebt_consolidation            7.1344
## verification_statusSource Verified   6.4590
## purposeother                         5.4426
## purposecredit_card                   4.4130
## term 60 months                       4.1780
## purposemajor_purchase                3.5212
## purposehome_improvement              3.2988
## purposesmall_business                2.5121
## purposemedical                       2.1501
## purposeeducational                   1.8843
## purposemoving                        1.8212
## purposevacation                      1.8024
## purposehouse                         0.4020
## purposewedding                       0.2704
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1200   86
##       bad     1    4
##
##                Accuracy : 0.9326
##                  95% CI : (0.9175, 0.9457)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.398
##
##                   Kappa : 0.0774
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.044444
##             Specificity : 0.999167
##          Pos Pred Value : 0.800000
##          Neg Pred Value : 0.933126
##              Prevalence : 0.069713
##          Detection Rate : 0.003098
##    Detection Prevalence : 0.003873
##       Balanced Accuracy : 0.521806
##
##        'Positive' Class : bad
##
```

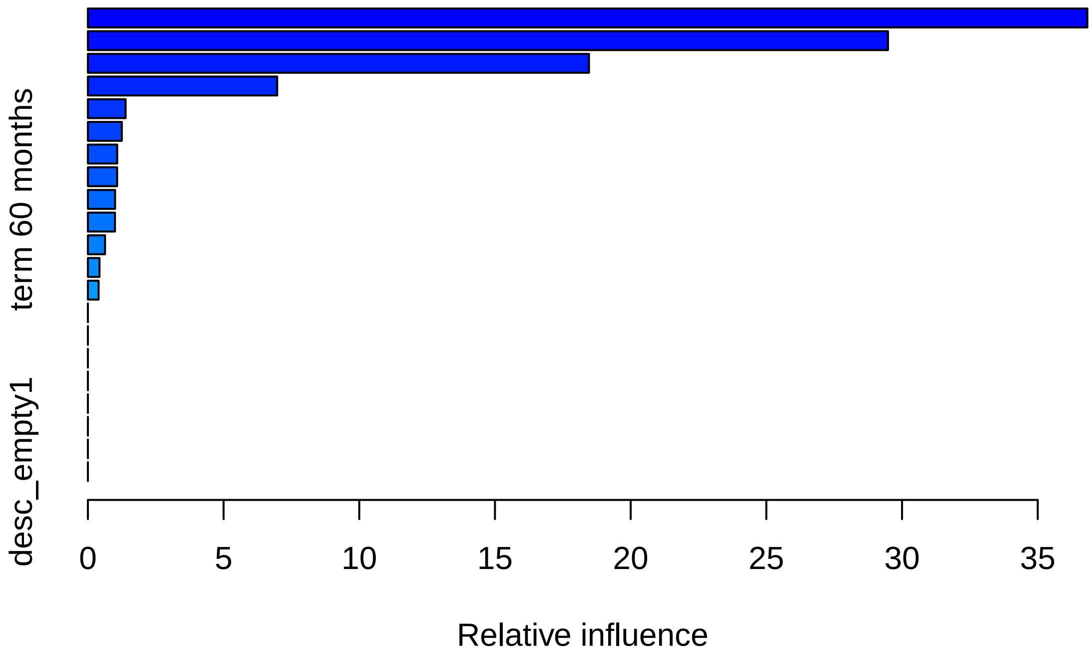



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.6326
```

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':
##
```

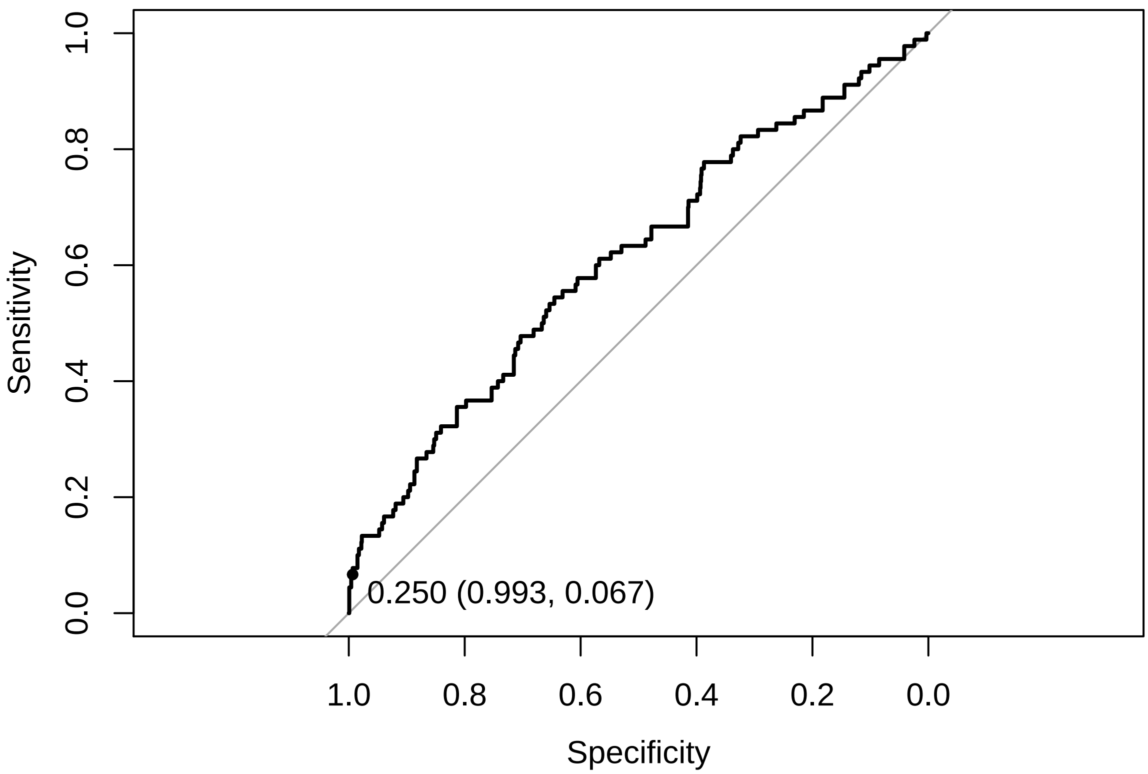
```
##     here
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## Stochastic Gradient Boosting
##
## 3879 samples
##    8 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      0.9309622  0.001521683  0.005038444
##   1                  100      0.9311271  0.013928115  0.004863730
##   1                  150      0.9311820  0.018783458  0.004810922
##   2                   50      0.9346431  0.108507612  0.004724030
##   2                  100      0.9344456  0.118716433  0.004647798
##   2                  150      0.9341684  0.117909781  0.004704876
##   3                   50      0.9341631  0.105790422  0.004720165
##   3                  100      0.9333460  0.107238602  0.004922351
##   3                  150      0.9330136  0.110015729  0.004807878
##   Kappa SD
##   0.005270038
##   0.015320117
##   0.015596949
##   0.032625211
##   0.035660790
##   0.037711962
##   0.029237127
##   0.035527897
##   0.033626069
##
## 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
## fico_range_high                                         fico_range_high
## inq_last_6mths                                           inq_last_6mths
## revol_util                                                   revol_util
## dti                                                                 dti
## purposevacation                                         purposevacation
## purposemedical                                           purposemedical
## purposeeducational                                   purposeeducational
## purposecredit_card                                   purposecredit_card
## term 60 months                                           term 60 months
## verification_statusSource Verified verification_statusSource Verified
## purposemajor_purchase                             purposemajor_purchase
## purposemoving                                             purposemoving
## purposesmall_business                             purposesmall_business
## verification_statusVerified                 verification_statusVerified
## purposedebt_consolidation                     purposedebt_consolidation
## purposehome_improvement                         purposehome_improvement
## purposehouse                                               purposehouse
## purposeother                                               purposeother
## purposerenewable_energy                         purposerenewable_energy
## purposewedding                                           purposewedding
## desc_empty1                                                 desc_empty1
##                                       rel.inf
## fico_range_high                    36.8292281
## inq_last_6mths                     29.4826593
## revol_util                         18.4645990
## dti                                 6.9749608
## purposevacation                     1.3897159
## purposemedical                      1.2500240
## purposeeducational                  1.0789692
## purposecredit_card                  1.0776871
## term 60 months                      1.0020256
## verification_statusSource Verified  0.9974563
```

```
## purposemajor_purchase             0.6315014
## purposemoving                     0.4266812
## purposesmall_business             0.3944922
## verification_statusVerified       0.0000000
## purposedebt_consolidation         0.0000000
## purposehome_improvement           0.0000000
## purposehouse                      0.0000000
## purposeother                      0.0000000
## purposerenewable_energy           0.0000000
## purposewedding                    0.0000000
## desc_empty1                       0.0000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                  Overall
## fico_range_high                  100.000
## inq_last_6mths                    80.052
## revol_util                        50.136
## dti                               18.939
## purposevacation                    3.773
## purposemedical                     3.394
## purposeeducational                 2.930
## purposecredit_card                 2.926
## term 60 months                     2.721
## verification_statusSource Verified   2.708
## purposemajor_purchase              1.715
## purposemoving                      1.159
## purposesmall_business              1.071
## purposeother                       0.000
## purposedebt_consolidation          0.000
## purposehome_improvement            0.000
## purposewedding                     0.000
## purposerenewable_energy            0.000
## verification_statusVerified        0.000
## purposehouse                       0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1200   86
##       bad     1    4
##
##                Accuracy : 0.9326
##                  95% CI : (0.9175, 0.9457)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.398
##
##                   Kappa : 0.0774
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.044444
##             Specificity : 0.999167
##          Pos Pred Value : 0.800000
```

```
##          Neg Pred Value : 0.933126
##              Prevalence : 0.069713
##          Detection Rate : 0.003098
##    Detection Prevalence : 0.003873
##       Balanced Accuracy : 0.521806
##
##        'Positive' Class : bad
##
```

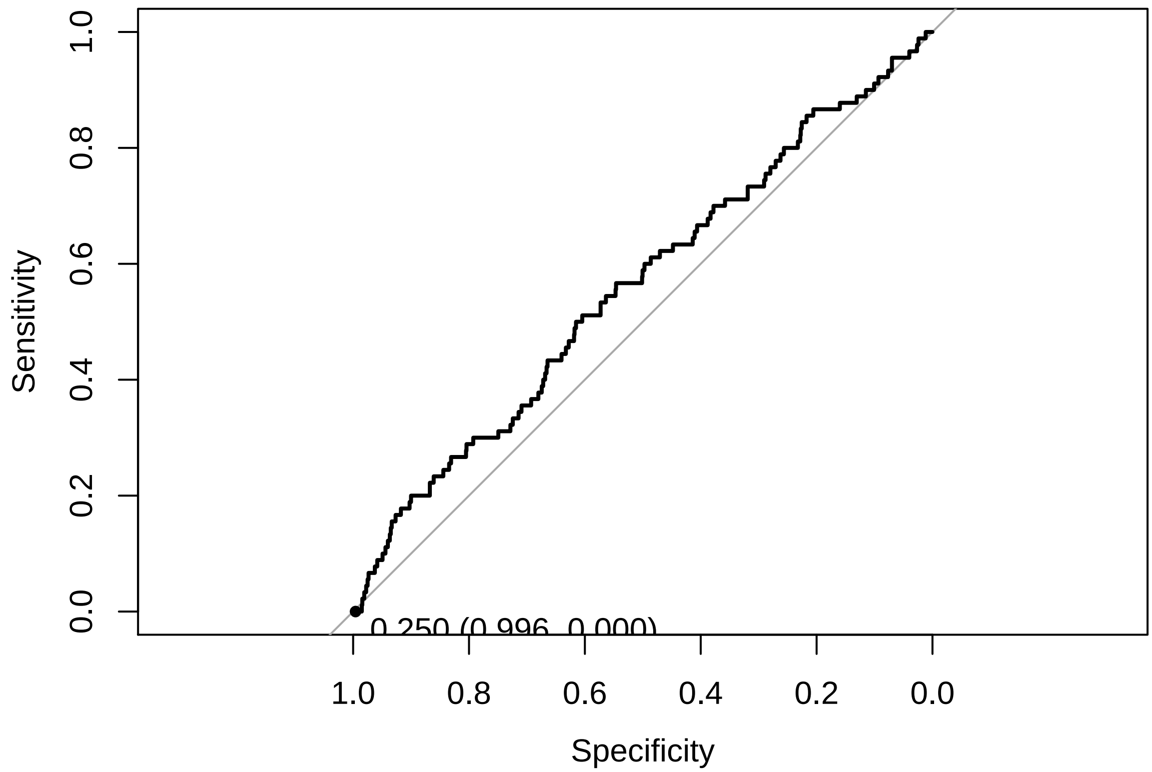



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1201 controls (dft_test$status good) < 90 cases (dft_test$status bad).
## Area under the curve: 0.6156
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 3879 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 3492, 3491, 3492, 3492, 3491, 3491, ...
```

```
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa          Accuracy SD  Kappa SD
##   0.25  0.9291055  -0.0009716599  0.00181565   0.003072658
##   0.50  0.9291055  -0.0009716599  0.00181565   0.003072658
##   1.00  0.9283343  -0.0023844932  0.00333221   0.005133377
##
## Tuning parameter 'sigma' was held constant at a value of 0.04330266
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.04330266 and C = 0.25.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                     Importance
## fico_range_high        100.000
## inq_last_6mths          74.482
## revol_util              48.884
## dti                     34.127
## purpose                 28.523
## term                    16.471
## verification_status      5.199
## desc_empty               0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1200   90
##       bad     1    0
##
##                Accuracy : 0.9295
##                  95% CI : (0.9142, 0.9429)
##     No Information Rate : 0.9303
##     P-Value [Acc > NIR] : 0.5711
##
##                   Kappa : -0.0015
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000000
##             Specificity : 0.9991674
##          Pos Pred Value : 0.0000000
##          Neg Pred Value : 0.9302326
##              Prevalence : 0.0697134
##          Detection Rate : 0.0000000
##    Detection Prevalence : 0.0007746
##       Balanced Accuracy : 0.4995837
##
##        '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: 0.5586
```

Neural Net Model

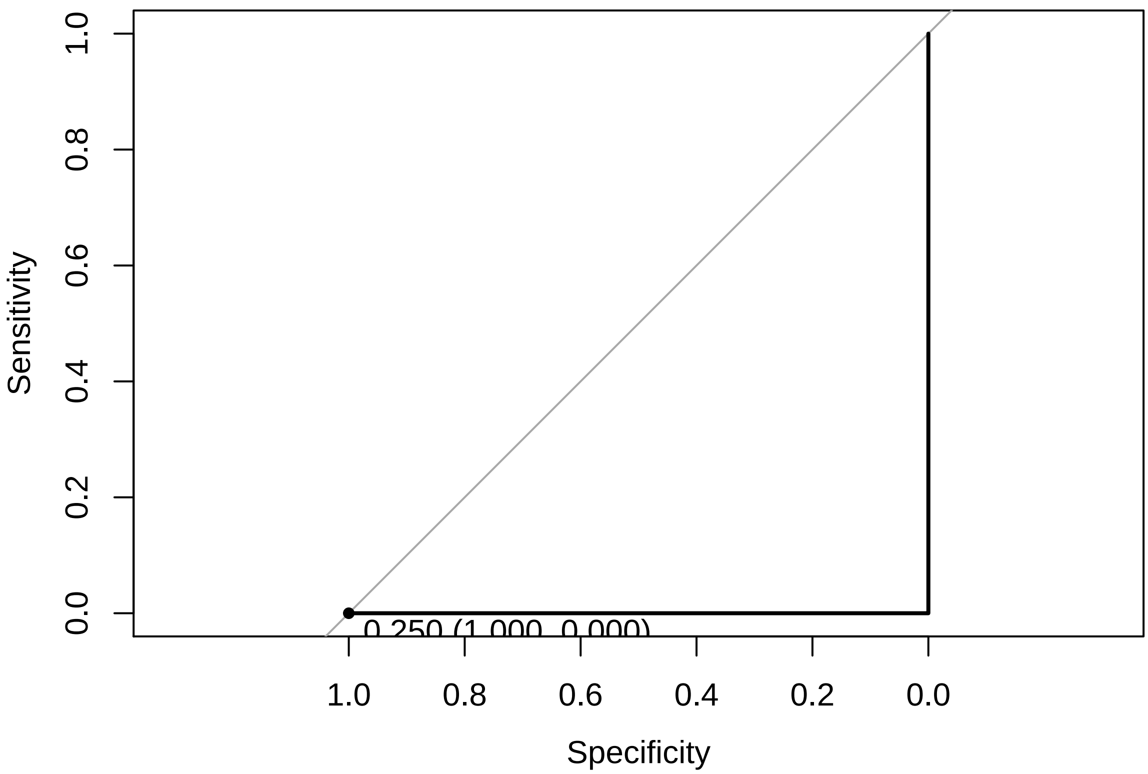
```
## Loading required package: nnet
```

```
## Neural Network
##
## 3879 samples
##    8 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  0.9314131  0.0000000000  0.006375732  0.000000000
##    1    0.0001000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    1    0.0003981072  0.9314131  0.0000000000  0.006375732  0.000000000
```

```
##    1    0.0015848932  0.9311269  0.0137287353  0.006875660  0.027560096
##    1    0.0063095734  0.9309427  0.0276526015  0.007719202  0.041856390
##    1    0.0251188643  0.9313581  0.0174853961  0.006335583  0.030470301
##    1    0.1000000000  0.9305653  0.0170773417  0.006539250  0.019153462
##    3    0.0000000000  0.9312412  0.0003995036  0.006317777  0.001997518
##    3    0.0001000000  0.9314417  0.0014173206  0.006367636  0.004912301
##    3    0.0003981072  0.9305350  0.0072093745  0.006159851  0.018918595
##    3    0.0015848932  0.9296083  0.0250393279  0.006876777  0.034008121
##    3    0.0063095734  0.9295689  0.0315582594  0.006714261  0.031565934
##    3    0.0251188643  0.9292100  0.0265299118  0.006614030  0.023508429
##    3    0.1000000000  0.9290967  0.0247928542  0.006560618  0.022928246
##    5    0.0000000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    5    0.0001000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    5    0.0003981072  0.9286870  0.0307622827  0.007488402  0.034636737
##    5    0.0015848932  0.9295163  0.0244966193  0.008014038  0.035453999
##    5    0.0063095734  0.9284471  0.0212012919  0.007994892  0.028935476
##    5    0.0251188643  0.9281816  0.0349266149  0.007525793  0.030108557
##    5    0.1000000000  0.9292096  0.0280156088  0.006695988  0.029458563
##    7    0.0000000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    7    0.0001000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    7    0.0003981072  0.9285121  0.0306403602  0.007090144  0.039206696
##    7    0.0015848932  0.9289822  0.0244694908  0.007049245  0.031633581
##    7    0.0063095734  0.9287505  0.0383024640  0.006742797  0.036849353
##    7    0.0251188643  0.9285225  0.0359703020  0.007374804  0.026272940
##    7    0.1000000000  0.9289584  0.0334823450  0.007149925  0.028646360
##    9    0.0000000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    9    0.0001000000  0.9314131  0.0000000000  0.006375732  0.000000000
##    9    0.0003981072  0.9283742  0.0242729301  0.007439244  0.031213126
##    9    0.0015848932  0.9282810  0.0278703227  0.006317317  0.031950128
##    9    0.0063095734  0.9283975  0.0319921972  0.007176717  0.028715014
##    9    0.0251188643  0.9283836  0.0377411258  0.005605744  0.030916602
##    9    0.1000000000  0.9289054  0.0316347303  0.007080991  0.027783880
##   11    0.0000000000  0.9314131  0.0000000000  0.006375732  0.000000000
##   11    0.0001000000  0.9314131  0.0000000000  0.006375732  0.000000000
##   11    0.0003981072  0.9292755  0.0283909087  0.007068210  0.038898970
##   11    0.0015848932  0.9285291  0.0196839186  0.006925561  0.024630116
##   11    0.0063095734  0.9285936  0.0289357997  0.006879712  0.034925661
##   11    0.0251188643  0.9285369  0.0343732505  0.007256518  0.035375467
##   11    0.1000000000  0.9294986  0.0281349341  0.007189684  0.028993621
##   13    0.0000000000  0.9314131  0.0000000000  0.006375732  0.000000000
##   13    0.0001000000  0.9313558  0.0006350763  0.006343543  0.003175382
##   13    0.0003981072  0.9272257  0.0235282593  0.007885154  0.038610694
##   13    0.0015848932  0.9281235  0.0318829349  0.008417973  0.039716045
##   13    0.0063095734  0.9275492  0.0333300707  0.008103395  0.031464665
##   13    0.0251188643  0.9277217  0.0201915960  0.007952914  0.027807165
##   13    0.1000000000  0.9297840  0.0273142330  0.006229462  0.022989233
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 3 and decay = 1e-04.
## a 21-3-1 network with 70 weights
## options were - entropy fitting  decay=1e-04
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1
##   -0.38    0.19   -0.42   -0.07    0.53    0.64   -0.48   -0.24    0.25
##  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
```

```
##    0.15   -0.01   -0.07    0.16    0.37   -0.52    0.12    0.37    0.11
## i18->h1 i19->h1 i20->h1 i21->h1
##    0.55   -0.64    0.27    0.18
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##    0.20    0.02   -0.01   -0.37    0.08    0.36    0.45    0.02   -0.50
##  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##   -0.45   -0.14   -0.26    0.12    0.29   -0.35    0.28   -0.07    0.50
## i18->h2 i19->h2 i20->h2 i21->h2
##    0.44    0.32    0.24    0.65
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##   -0.72    0.54    0.32    0.44   -0.25    0.33   -0.39   -0.61    0.20
##  i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##   -0.04    0.25    0.20   -0.06   -0.59   -0.11   -0.26   -0.02  -33.86
## i18->h3 i19->h3 i20->h3 i21->h3
##    0.63    0.29    0.13   -0.22
##   b->o  h1->o  h2->o  h3->o
##  -0.51  -1.48  -0.59  -3.32
## nnet variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                    Overall
## fico_range_high                   100.000
## inq_last_6mths                     15.575
## purposedebt_consolidation          14.883
## revol_util                         14.002
## purposeeducational                 13.986
## dti                                12.200
## purposesmall_business              11.995
## purposehouse                       10.443
## purposerenewable_energy             9.813
## purposecredit_card                  7.790
## purposemajor_purchase               7.319
## desc_empty1                         5.920
## verification_statusVerified         5.758
## verification_statusSource Verified   4.814
## purposevacation                     4.277
## purposewedding                      4.202
## purposemoving                       3.046
## purposehome_improvement             2.687
## purposeother                        1.720
## term 60 months                      1.715
## 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: 0.5
```

Results for Grade B Loans

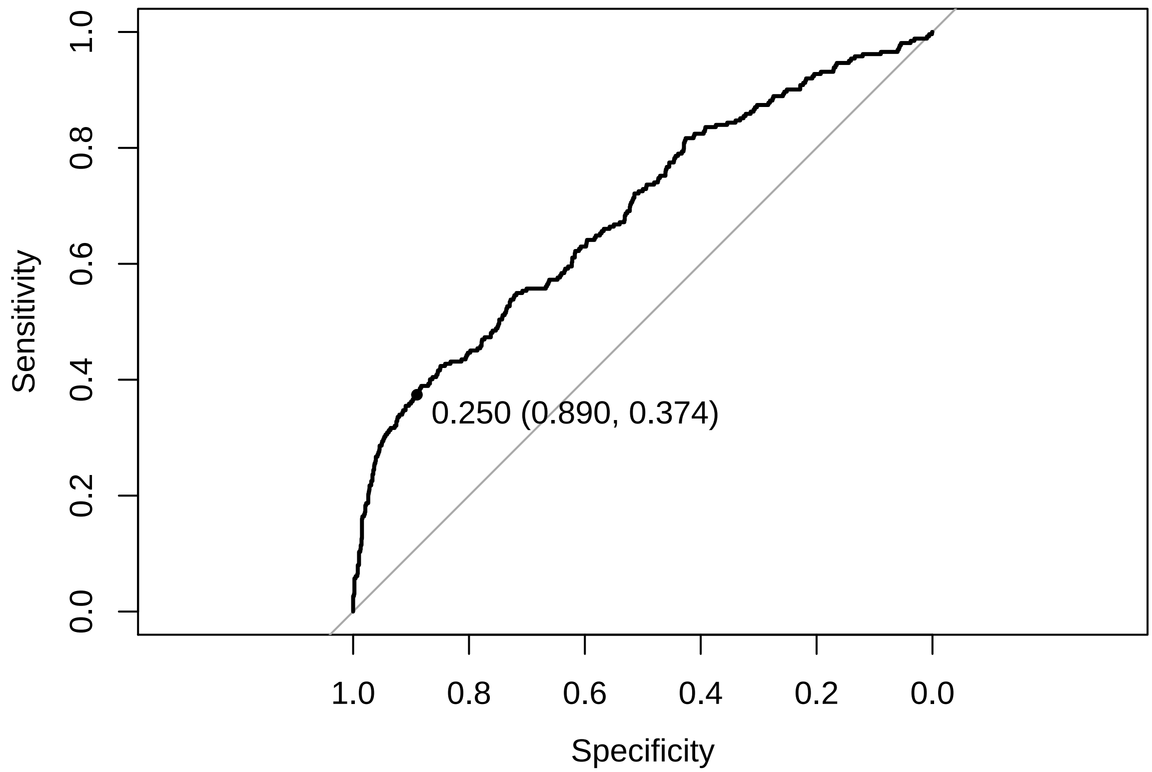
Approximately 16% of the Grade B loans in this dataset went bad. With the four models, we were able to predict between 9% and 24% of the bad loans. This predictive ability is based on a 50% probability classification cutoff. As the ROC curves show, it's possible to predict the bad loans with a higher 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
##    8 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
##   0.852189  0.1596097  0.007753126  0.02740123
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1360  -0.5811  -0.4540  -0.3566   2.6112
##
## Coefficients:
##                                       Estimate Std. Error z value
## (Intercept)                          2.6066656  1.4653265   1.779
## `term 60 months`                     0.7749892  0.1161878   6.670
## `verification_statusSource Verified` -0.0659992  0.1150297  -0.574
## verification_statusVerified          0.0666980  0.0995895   0.670
## purposecredit_card                   0.2553708  0.2791578   0.915
## purposedebt_consolidation            0.5525685  0.2542210   2.174
## purposeeducational                   0.9893367  0.3762873   2.629
## purposehome_improvement              0.3032303  0.2890816   1.049
## purposehouse                         0.5397150  0.4922989   1.096
## purposemajor_purchase                0.5753198  0.2986145   1.927
## purposemedical                       0.5736293  0.3771731   1.521
## purposemoving                        1.0354444  0.3713195   2.789
## purposeother                         0.8298115  0.2701063   3.072
## purposerenewable_energy              1.0305765  0.7071402   1.457
## purposesmall_business                1.5296503  0.2966281   5.157
## purposevacation                      0.7186741  0.4894408   1.468
## purposewedding                      -0.1303746  0.4310690  -0.302
## fico_range_high                     -0.0079534  0.0019592  -4.060
## inq_last_6mths                       0.5107813  0.0284980  17.923
## revol_util                           0.0031086  0.0018592   1.672
## desc_empty1                         -0.1450758  0.1024805  -1.416
```

```
## dti                                0.0000564  0.0066925   0.008
##                                   Pr(>|z|)
## (Intercept)                        0.07526 .
## `term 60 months`                  2.56e-11 ***
## `verification_statusSource Verified`  0.56613
## verification_statusVerified        0.50303
## purposecredit_card                 0.36030
## purposedebt_consolidation          0.02974 *
## purposeeducational                 0.00856 **
## purposehome_improvement            0.29420
## purposehouse                       0.27294
## purposemajor_purchase              0.05403 .
## purposemedical                     0.12829
## purposemoving                      0.00529 **
## purposeother                       0.00213 **
## purposerenewable_energy            0.14501
## purposesmall_business             2.51e-07 ***
## purposevacation                    0.14201
## purposewedding                     0.76231
## fico_range_high                   4.92e-05 ***
## inq_last_6mths                     < 2e-16 ***
## revol_util                         0.09452 .
## desc_empty1                        0.15688
## dti                                0.99328
## ---
## 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: 3870.0  on 4907  degrees of freedom
## AIC: 3914
##
## Number of Fisher Scoring iterations: 5
##
## glm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                   Overall
## inq_last_6mths                    100.000
## `term 60 months`                   37.185
## purposesmall_business              28.738
## fico_range_high                    22.613
## purposeother                       17.102
## purposemoving                      15.518
## purposeeducational                 14.629
## purposedebt_consolidation          12.086
## purposemajor_purchase              10.707
## revol_util                          9.286
## purposemedical                      8.442
## purposevacation                     8.149
## purposerenewable_energy             8.088
## desc_empty1                         7.855
```

```
## purposehouse                          6.072
## purposehome_improvement               5.808
## purposecredit_card                    5.059
## verification_statusVerified           3.691
## `verification_statusSource Verified`   3.156
## purposewedding                        1.641
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1366  238
##       bad    14   24
##
##                Accuracy : 0.8465
##                  95% CI : (0.8282, 0.8636)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 0.2625
##
##                   Kappa : 0.1246
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.09160
##             Specificity : 0.98986
##          Pos Pred Value : 0.63158
##          Neg Pred Value : 0.85162
##              Prevalence : 0.15956
##          Detection Rate : 0.01462
##    Detection Prevalence : 0.02314
##       Balanced Accuracy : 0.54073
##
##        'Positive' Class : bad
##
```



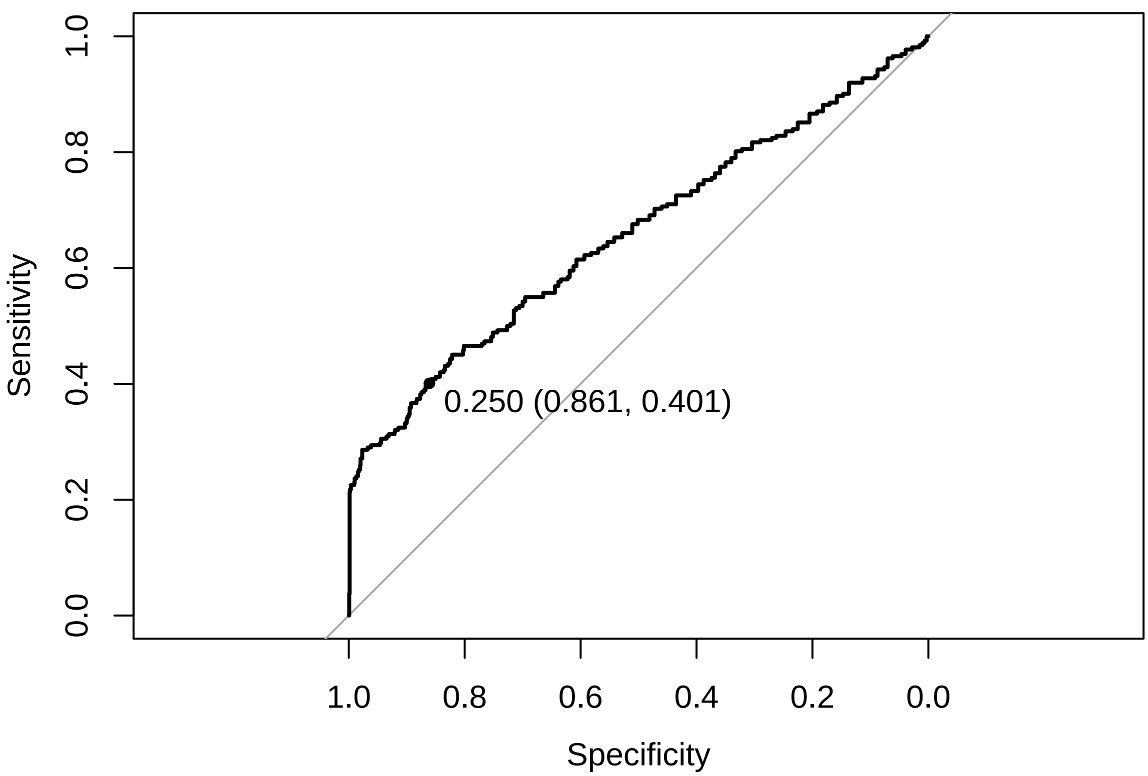

```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.6863
```

Random Forest Model

```
## Random Forest
##
## 4929 samples
##    8 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    0.8635367  0.2297458  0.007410032  0.02667733
##   11    0.8644320  0.3041105  0.005917967  0.01578816
##   21    0.8590118  0.2950494  0.006640022  0.01442449
##
## Accuracy was used to select the optimal model using  the largest value.
```

```
## The final value used for the model was mtry = 11.
##                 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        21   -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            21   -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 21)
##
##                                    Overall
## dti                               100.0000
## revol_util                         93.8266
## inq_last_6mths                     73.9867
## fico_range_high                    64.4162
## verification_statusVerified         8.8135
## desc_empty1                         8.4223
## verification_statusSource Verified   6.9597
## purposedebt_consolidation           6.3432
## purposeother                        4.0093
## term 60 months                      3.7008
## purposesmall_business               3.6719
## purposemajor_purchase               3.1389
## purposecredit_card                  3.0284
## purposehome_improvement             2.9899
## purposeeducational                  1.8292
## purposemoving                       1.5512
## purposemedical                      1.3957
## purposewedding                      0.7976
## purposehouse                        0.3181
## purposevacation                     0.2359
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1366  200
```

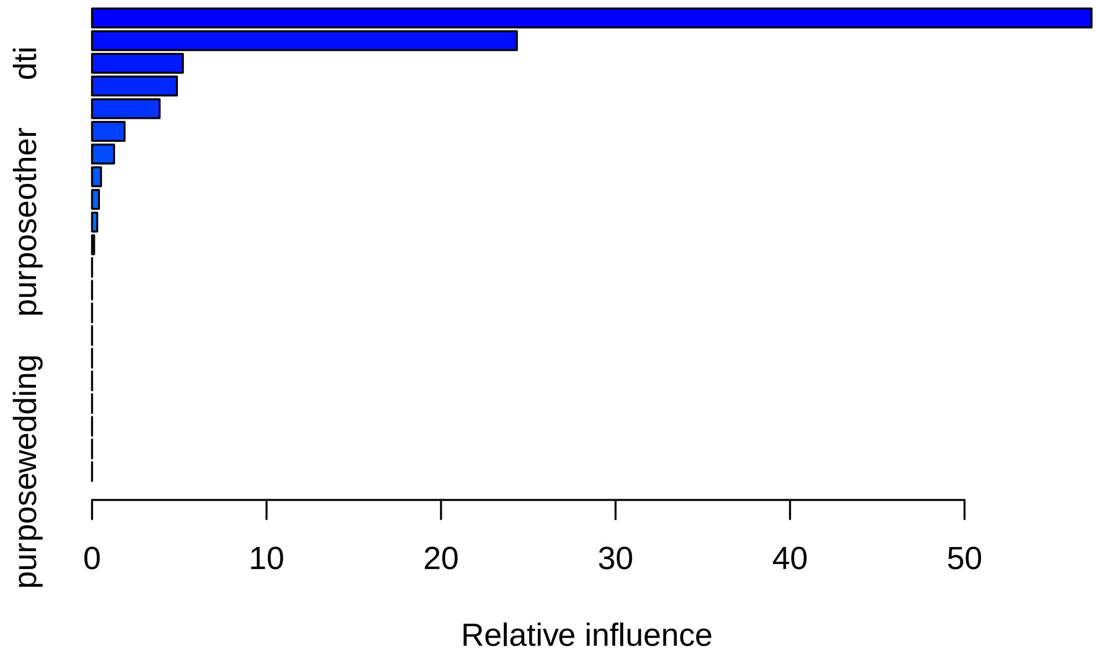
```
##       bad    14   62
##
##                Accuracy : 0.8697
##                  95% CI : (0.8524, 0.8856)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 0.0005217
##
##                   Kappa : 0.3179
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.23664
##             Specificity : 0.98986
##          Pos Pred Value : 0.81579
##          Neg Pred Value : 0.87229
##              Prevalence : 0.15956
##          Detection Rate : 0.03776
##    Detection Prevalence : 0.04629
##       Balanced Accuracy : 0.61325
##
##        '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: 0.6603
```

Gradient Boost Model

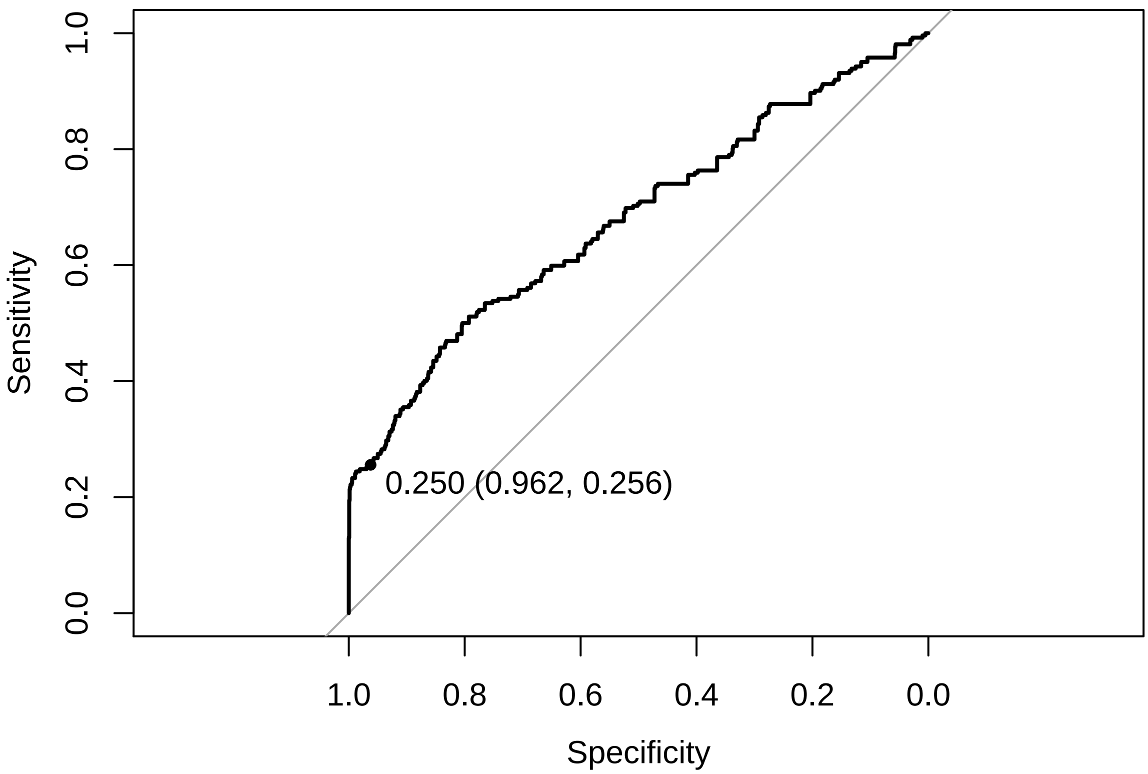
```
## Stochastic Gradient Boosting
##
## 4929 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 4929, 4929, 4929, 4929, 4929, 4929, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.8644154  0.3092467  0.006107559
##   1                  100      0.8653681  0.3090312  0.006084367
##   1                  150      0.8656568  0.3089530  0.005874036
##   2                   50      0.8746644  0.3144393  0.005847093
##   2                  100      0.8740015  0.3161049  0.005644078
##   2                  150      0.8731007  0.3171293  0.005771584
##   3                   50      0.8745322  0.3144797  0.006221729
##   3                  100      0.8733393  0.3155837  0.006097918
##   3                  150      0.8721475  0.3172719  0.006542461
##   Kappa SD
##   0.02061755
##   0.02084243
##   0.02070615
##   0.01939746
##   0.02062341
##   0.02115355
##   0.02157120
##   0.02160084
##   0.02312819
##
## 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
## inq_last_6mths                                          inq_last_6mths
## fico_range_high                                        fico_range_high
## dti                                                                dti
## purposesmall_business                            purposesmall_business
## term 60 months                                          term 60 months
## revol_util                                                  revol_util
## purposemoving                                            purposemoving
## purposecredit_card                                  purposecredit_card
## desc_empty1                                                desc_empty1
## purposeother                                              purposeother
## purposeeducational                                  purposeeducational
## verification_statusSource Verified verification_statusSource Verified
## verification_statusVerified               verification_statusVerified
## purposedebt_consolidation                    purposedebt_consolidation
## purposehome_improvement                        purposehome_improvement
## purposehouse                                              purposehouse
## purposemajor_purchase                            purposemajor_purchase
## purposemedical                                          purposemedical
## purposerenewable_energy                        purposerenewable_energy
## purposevacation                                        purposevacation
## purposewedding                                          purposewedding
##                                       rel.inf
## inq_last_6mths                     57.2804494
## fico_range_high                    24.3446578
## dti                                 5.2022772
## purposesmall_business               4.8616249
## term 60 months                      3.8714283
## revol_util                          1.8618391
## purposemoving                       1.2603728
## purposecredit_card                  0.5133002
## desc_empty1                         0.3937440
## purposeother                        0.2931790
```

```
## purposeeducational                  0.1171271
## verification_statusSource Verified  0.0000000
## verification_statusVerified         0.0000000
## purposedebt_consolidation           0.0000000
## purposehome_improvement             0.0000000
## purposehouse                        0.0000000
## purposemajor_purchase               0.0000000
## purposemedical                      0.0000000
## purposerenewable_energy             0.0000000
## purposevacation                     0.0000000
## purposewedding                      0.0000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                     Overall
## inq_last_6mths                     100.0000
## fico_range_high                     42.5008
## dti                                  9.0821
## purposesmall_business                8.4874
## term 60 months                       6.7587
## revol_util                           3.2504
## purposemoving                        2.2004
## purposecredit_card                   0.8961
## desc_empty1                          0.6874
## purposeother                         0.5118
## purposeeducational                   0.2045
## purposemedical                       0.0000
## purposerenewable_energy              0.0000
## verification_statusVerified          0.0000
## purposevacation                      0.0000
## purposehome_improvement              0.0000
## purposedebt_consolidation            0.0000
## verification_statusSource Verified   0.0000
## purposemajor_purchase                0.0000
## purposewedding                       0.0000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1377  205
##       bad     3   57
##
##                Accuracy : 0.8733
##                  95% CI : (0.8563, 0.889)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 0.0001048
##
##                   Kappa : 0.3132
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.21756
##             Specificity : 0.99783
##          Pos Pred Value : 0.95000
```

```
##          Neg Pred Value : 0.87042
##              Prevalence : 0.15956
##          Detection Rate : 0.03471
##    Detection Prevalence : 0.03654
##       Balanced Accuracy : 0.60769
##
##        'Positive' Class : bad
##
```

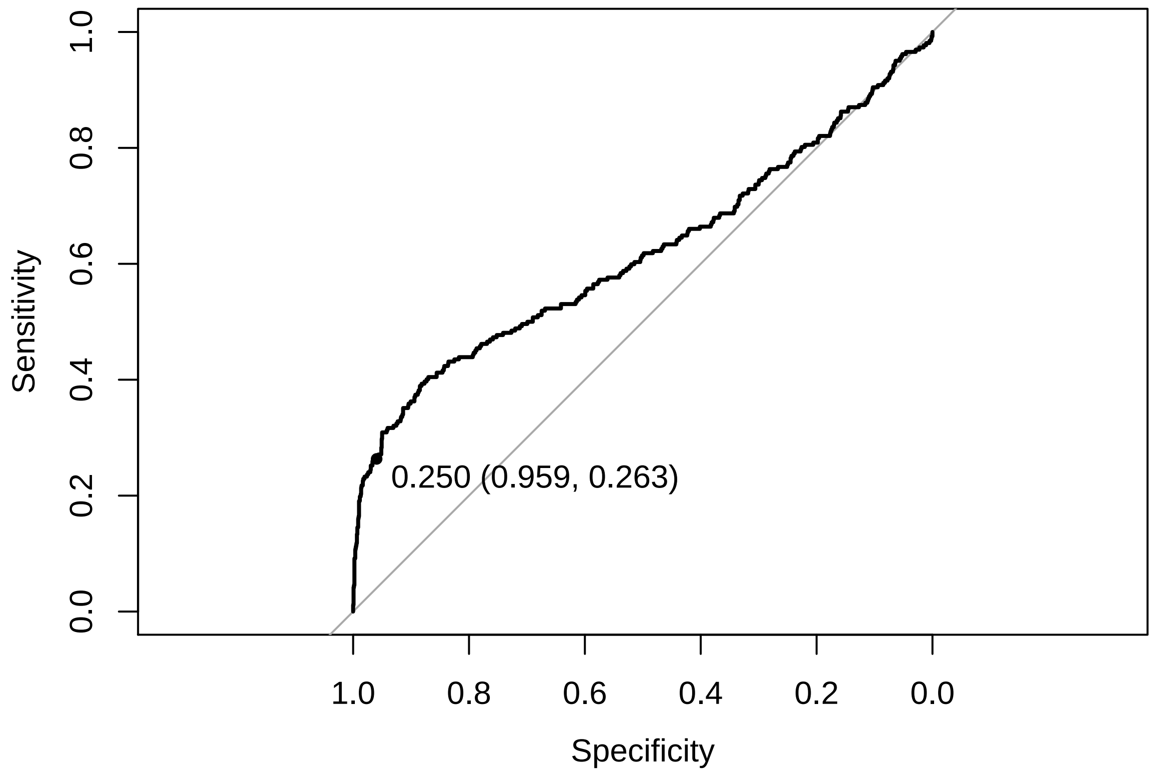



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 1380 controls (dft_test$status good) < 262 cases (dft_test$status bad).
## Area under the curve: 0.6826
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 4929 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 4437, 4437, 4435, 4436, 4436, 4436, ...
```

```
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.8573750  0.2108097  0.007277658  0.04830960
##   0.50  0.8577798  0.2060922  0.007509251  0.06221716
##   1.00  0.8547372  0.1822184  0.008875300  0.06742620
##
## Tuning parameter 'sigma' was held constant at a value of 0.04893444
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.04893444 and C = 0.5.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                     Importance
## inq_last_6mths        100.0000
## purpose                45.3402
## term                   26.3585
## revol_util             14.0407
## dti                    10.9562
## verification_status     6.4893
## fico_range_high         0.6599
## desc_empty              0.0000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1366  218
##       bad    14   44
##
##                Accuracy : 0.8587
##                  95% CI : (0.8409, 0.8752)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 0.02211
##
##                   Kappa : 0.2305
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.16794
##             Specificity : 0.98986
##          Pos Pred Value : 0.75862
##          Neg Pred Value : 0.86237
##              Prevalence : 0.15956
##          Detection Rate : 0.02680
##    Detection Prevalence : 0.03532
##       Balanced Accuracy : 0.57890
##
##        '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: 0.6177
```

Neural Net Model

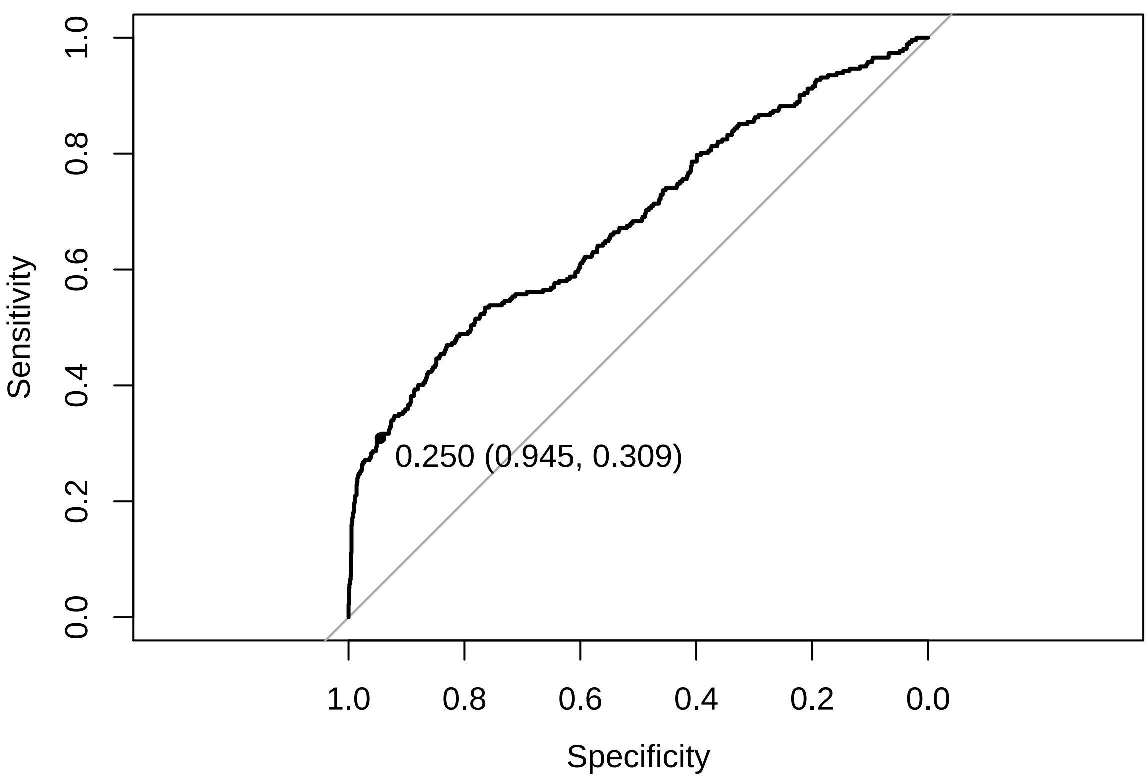
```
## Neural Network
##
## 4929 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 4929, 4929, 4929, 4929, 4929, 4929, ...
##
## Resampling results across tuning parameters:
##
##   size  decay         Accuracy   Kappa          Accuracy SD  Kappa SD
##    1    0.0000000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##    1    0.0001000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##    1    0.0003981072  0.8407410   0.0000000000  0.004976671  0.0000000000
##    1    0.0015848932  0.8487407   0.1166407755  0.011358002  0.1365143555
##    1    0.0063095734  0.8531737   0.2004297574  0.009228100  0.1189325188
```

```
##    1    0.0251188643  0.8553651   0.2160149502  0.010171553  0.1059756145
##    1    0.1000000000  0.8517362   0.1782672851  0.009737439  0.1115714820
##    3    0.0000000000  0.8412487   0.0091964584  0.006256018  0.0459822919
##    3    0.0001000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##    3    0.0003981072  0.8407410   0.0000000000  0.004976671  0.0000000000
##    3    0.0015848932  0.8551443   0.2128687367  0.011967744  0.1145620389
##    3    0.0063095734  0.8557416   0.2486093983  0.009780391  0.0657582899
##    3    0.0251188643  0.8571805   0.2591216913  0.007994162  0.0478734894
##    3    0.1000000000  0.8590925   0.2657295803  0.007724655  0.0518105067
##    5    0.0000000000  0.8416681   0.0119041551  0.006768089  0.0595207754
##    5    0.0001000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##    5    0.0003981072  0.8415231   0.0123868524  0.006202067  0.0619342621
##    5    0.0015848932  0.8549162   0.2295495491  0.010272947  0.1001840499
##    5    0.0063095734  0.8589054   0.2641858045  0.008775057  0.0607178222
##    5    0.0251188643  0.8599502   0.2731247357  0.007920201  0.0433039572
##    5    0.1000000000  0.8592197   0.2661697848  0.009908361  0.0554534089
##    7    0.0000000000  0.8406767  -0.0001277291  0.004971349  0.0006386456
##    7    0.0001000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##    7    0.0003981072  0.8457168   0.0694274181  0.010527439  0.1274343736
##    7    0.0015848932  0.8563950   0.2394995791  0.010595310  0.0909267908
##    7    0.0063095734  0.8538728   0.2481857002  0.008832236  0.0463600544
##    7    0.0251188643  0.8595199   0.2688256920  0.008179993  0.0382096255
##    7    0.1000000000  0.8607912   0.2780053813  0.009214849  0.0502216190
##    9    0.0000000000  0.8413440   0.0114596623  0.006239504  0.0572983114
##    9    0.0001000000  0.8429545   0.0246018624  0.008271495  0.0851642907
##    9    0.0003981072  0.8543364   0.2142199862  0.009522589  0.1065915709
##    9    0.0015848932  0.8565097   0.2556394978  0.009160070  0.0632149725
##    9    0.0063095734  0.8561892   0.2496591403  0.008537921  0.0669825609
##    9    0.0251188643  0.8579707   0.2692553082  0.009095702  0.0440635750
##    9    0.1000000000  0.8598078   0.2693059927  0.007041489  0.0438599165
##   11    0.0000000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##   11    0.0001000000  0.8424812   0.0224418467  0.008217192  0.0759065637
##   11    0.0003981072  0.8532770   0.2266956646  0.010493452  0.0922037995
##   11    0.0015848932  0.8583485   0.2589424437  0.006793353  0.0609402202
##   11    0.0063095734  0.8596881   0.2719421286  0.009379392  0.0410828002
##   11    0.0251188643  0.8592253   0.2687510216  0.008084814  0.0394419196
##   11    0.1000000000  0.8603411   0.2709956446  0.008373771  0.0412558508
##   13    0.0000000000  0.8404333   0.0125978144  0.004925539  0.0609115540
##   13    0.0001000000  0.8407410   0.0000000000  0.004976671  0.0000000000
##   13    0.0003981072  0.8540589   0.2139475456  0.010608414  0.1175720286
##   13    0.0015848932  0.8572756   0.2529803857  0.010846403  0.0852681075
##   13    0.0063095734  0.8598439   0.2770275008  0.008318171  0.0441191172
##   13    0.0251188643  0.8606197   0.2798537325  0.007142113  0.0338834065
##   13    0.1000000000  0.8616783   0.2822570034  0.006644506  0.0306703299
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 13 and decay = 0.1.
## a 21-13-1 network with 300 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    0.02
## i18->h1 i19->h1 i20->h1 i21->h1
```

```
##    0.00    0.00    0.00    0.00
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.03
## i18->h2 i19->h2 i20->h2 i21->h2
##    0.00    0.00    0.00    0.00
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##   -1.66    1.29   -0.10    1.13    0.22    0.24   -2.08    1.66   -0.52
##  i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##   -0.45    0.81   -1.09    0.32    0.05    0.14   -2.02    0.05    0.03
## i18->h3 i19->h3 i20->h3 i21->h3
##   -5.62    0.01    0.15   -0.12
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h4 i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   -0.32
## i18->h4 i19->h4 i20->h4 i21->h4
##    0.00   -0.02    0.00   -0.01
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h5 i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.01
## i18->h5 i19->h5 i20->h5 i21->h5
##    0.00    0.00    0.00    0.00
##   b->h6  i1->h6  i2->h6  i3->h6  i4->h6  i5->h6  i6->h6  i7->h6  i8->h6
##    0.02    0.26    0.05   -0.14    0.06   -0.45   -0.04    0.02    0.01
##  i9->h6 i10->h6 i11->h6 i12->h6 i13->h6 i14->h6 i15->h6 i16->h6 i17->h6
##    0.19    0.00    0.03    0.19    0.02    0.07   -0.05   -0.03    0.00
## i18->h6 i19->h6 i20->h6 i21->h6
##   -0.71    0.07    0.19   -0.27
##   b->h7  i1->h7  i2->h7  i3->h7  i4->h7  i5->h7  i6->h7  i7->h7  i8->h7
##   -1.62   -2.92    1.29    1.23    0.25   -0.73   -1.49    0.88    0.73
##  i9->h7 i10->h7 i11->h7 i12->h7 i13->h7 i14->h7 i15->h7 i16->h7 i17->h7
##   -0.16   -0.50    0.25   -2.24   -0.97   -4.40    0.79    1.65    0.00
## i18->h7 i19->h7 i20->h7 i21->h7
##   -0.23   -0.03    0.89    0.20
##   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    0.00    0.00    0.00
##  i9->h8 i10->h8 i11->h8 i12->h8 i13->h8 i14->h8 i15->h8 i16->h8 i17->h8
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.02
## i18->h8 i19->h8 i20->h8 i21->h8
##    0.00    0.00    0.00    0.00
##   b->h9  i1->h9  i2->h9  i3->h9  i4->h9  i5->h9  i6->h9  i7->h9  i8->h9
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h9 i10->h9 i11->h9 i12->h9 i13->h9 i14->h9 i15->h9 i16->h9 i17->h9
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
## i18->h9 i19->h9 i20->h9 i21->h9
##    0.00    0.00    0.00    0.00
##   b->h10  i1->h10  i2->h10  i3->h10  i4->h10  i5->h10  i6->h10  i7->h10
##     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
##  i8->h10  i9->h10 i10->h10 i11->h10 i12->h10 i13->h10 i14->h10 i15->h10
##     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## i16->h10 i17->h10 i18->h10 i19->h10 i20->h10 i21->h10
```

```
##     0.00     0.07     0.00     0.00     0.00     0.00
##   b->h11  i1->h11  i2->h11  i3->h11  i4->h11  i5->h11  i6->h11  i7->h11
##     0.29     2.45     0.51     1.39    -0.90    -1.15    -0.05    -0.29
##  i8->h11  i9->h11 i10->h11 i11->h11 i12->h11 i13->h11 i14->h11 i15->h11
##     1.01     0.97    -0.44     1.73    -0.71     0.67     1.61    -0.14
## i16->h11 i17->h11 i18->h11 i19->h11 i20->h11 i21->h11
##    -0.85    -0.01     0.83     0.03     0.65     0.26
##   b->h12  i1->h12  i2->h12  i3->h12  i4->h12  i5->h12  i6->h12  i7->h12
##     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
##  i8->h12  i9->h12 i10->h12 i11->h12 i12->h12 i13->h12 i14->h12 i15->h12
##     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## i16->h12 i17->h12 i18->h12 i19->h12 i20->h12 i21->h12
##     0.00     0.02     0.00     0.00     0.00     0.00
##   b->h13  i1->h13  i2->h13  i3->h13  i4->h13  i5->h13  i6->h13  i7->h13
##     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
##  i8->h13  i9->h13 i10->h13 i11->h13 i12->h13 i13->h13 i14->h13 i15->h13
##     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
## i16->h13 i17->h13 i18->h13 i19->h13 i20->h13 i21->h13
##     0.00     0.02     0.00     0.01     0.00     0.01
##   b->o  h1->o  h2->o  h3->o  h4->o  h5->o  h6->o  h7->o  h8->o  h9->o
##   0.20   0.20   0.20  -2.75   0.01   0.20  -0.22  -1.37   0.20   0.01
## h10->o h11->o h12->o h13->o
##   0.20   0.72   0.20   0.20
## nnet variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                    Overall
## fico_range_high                   100.0000
## revol_util                         14.6737
## inq_last_6mths                      9.5596
## dti                                 8.2863
## term 60 months                      5.8268
## purposesmall_business               3.6930
## purposedebt_consolidation           2.8446
## verification_statusVerified         2.6410
## purposeother                        2.1343
## purposeeducational                  1.6456
## purposemoving                       1.3324
## purposevacation                     1.2172
## desc_empty1                         1.1106
## purposehome_improvement             1.0625
## purposemajor_purchase               0.9861
## purposewedding                      0.7776
## purposehouse                        0.5814
## purposecredit_card                  0.3769
## verification_statusSource Verified   0.2735
## purposerenewable_energy             0.1176
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good  bad
##       good 1357  198
##       bad    23   64
```

```
##
##                Accuracy : 0.8654
##                  95% CI : (0.8479, 0.8816)
##     No Information Rate : 0.8404
##     P-Value [Acc > NIR] : 0.002696
##
##                   Kappa : 0.312
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.24427
##             Specificity : 0.98333
##          Pos Pred Value : 0.73563
##          Neg Pred Value : 0.87267
##              Prevalence : 0.15956
##          Detection Rate : 0.03898
##    Detection Prevalence : 0.05298
##       Balanced Accuracy : 0.61380
##
##        '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: 0.6835
```

Results for Grade C Loans

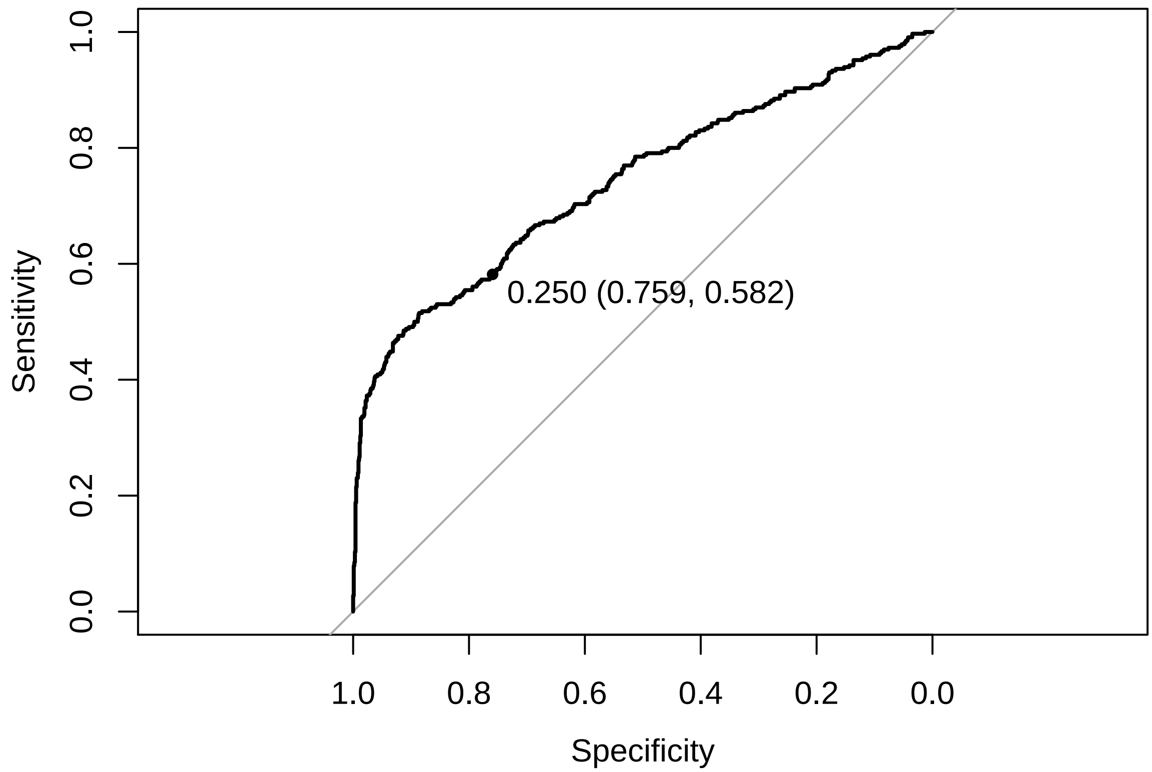
Approximately 25% of the Grade C loans in this dataset went bad. With the four models, we were able to correctly predict between 38% and 40% of the bad loans. This predictive ability is based on a 50% probability classification cutoff. As the ROC curves show, it's possible to predict the bad loans with 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
##    8 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
##   0.8138437  0.4066162  0.008216202  0.02338866
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4061  -0.6929  -0.5200   0.1874   2.3881
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          -1.251999   1.511652  -0.828 0.407539
## `term 60 months`                      0.411822   0.123927   3.323 0.000890
## `verification_statusSource Verified` -0.573011   0.121457  -4.718 2.38e-06
## verification_statusVerified          -0.360102   0.100025  -3.600 0.000318
## purposecredit_card                   -0.138060   0.279595  -0.494 0.621459
## purposedebt_consolidation             0.032247   0.258738   0.125 0.900814
## purposeeducational                    0.013438   0.380726   0.035 0.971844
## purposehome_improvement               0.069998   0.295355   0.237 0.812659
## purposehouse                          0.152596   0.460962   0.331 0.740617
## purposemajor_purchase                -0.061935   0.313679  -0.197 0.843477
## purposemedical                        0.646623   0.382410   1.691 0.090852
## purposemoving                        -0.160948   0.461403  -0.349 0.727222
## purposeother                          0.213156   0.274585   0.776 0.437582
## purposerenewable_energy               0.231298   1.159867   0.199 0.841936
## purposesmall_business                 0.738967   0.311577   2.372 0.017707
## purposevacation                      -0.090012   0.514706  -0.175 0.861173
```

```
## purposewedding                        0.026225   0.367832   0.071 0.943161
## fico_range_high                      -0.001371   0.002140  -0.641 0.521701
## inq_last_6mths                        0.594746   0.027186  21.877  < 2e-16
## revol_util                            0.001253   0.001671   0.750 0.453423
## desc_empty1                          -0.150426   0.104784  -1.436 0.151119
## dti                                   0.013190   0.006709   1.966 0.049282
##
## (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
## 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: 4430.0  on 3918  degrees of freedom
## Residual deviance: 3704.1  on 3897  degrees of freedom
## AIC: 3748.1
##
## Number of Fisher Scoring iterations: 4
##
## glm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                       Overall
## inq_last_6mths                       100.0000
## `verification_statusSource Verified`  21.4387
## verification_statusVerified           16.3213
## `term 60 months`                      15.0530
## purposesmall_business                 10.6971
## dti                                    8.8403
## purposemedical                         7.5802
## desc_empty1                            6.4112
## purposeother                           3.3926
```

```
## revol_util                               3.2710
## fico_range_high                          2.7719
## purposecredit_card                       2.0992
## purposemoving                            1.4355
## purposehouse                             1.3540
## purposehome_improvement                  0.9235
## purposerenewable_energy                  0.7514
## purposemajor_purchase                    0.7424
## purposevacation                          0.6391
## purposedebt_consolidation                0.4090
## purposewedding                           0.1648
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  951 207
##       bad    25 123
##
##                Accuracy : 0.8224
##                  95% CI : (0.8005, 0.8427)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : 5.861e-11
##
##                   Kappa : 0.4246
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.37273
##             Specificity : 0.97439
##          Pos Pred Value : 0.83108
##          Neg Pred Value : 0.82124
##              Prevalence : 0.25268
##          Detection Rate : 0.09418
##    Detection Prevalence : 0.11332
##       Balanced Accuracy : 0.67356
##
##        'Positive' Class : bad
##
```



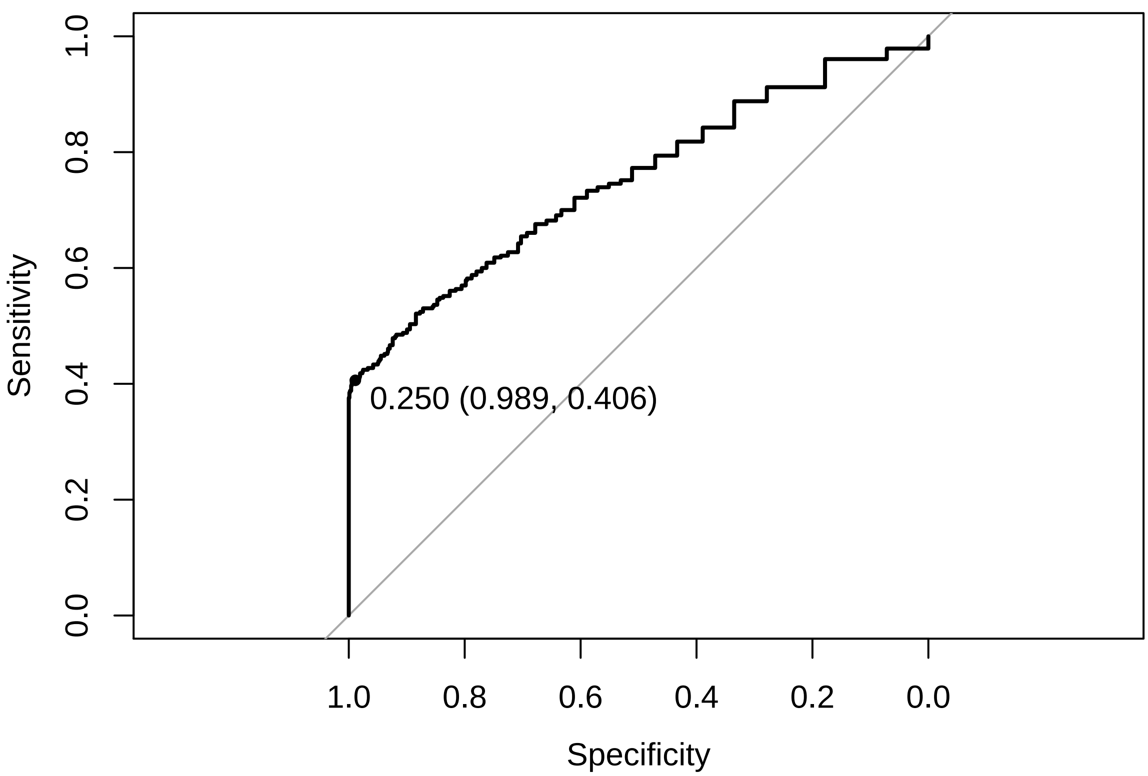

```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.7416
```

Random Forest Model

```
## Random Forest
##
## 3919 samples
##    8 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    0.8365669  0.4503175  0.007509750  0.02266792
##   11    0.8286954  0.4478922  0.007974369  0.02465436
##   21    0.8223741  0.4341683  0.009437999  0.02736135
##
## Accuracy was used to select the optimal model using  the largest value.
```

```
## The final value used for the model was mtry = 2.
##                 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        21   -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            21   -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 21)
##
##                                    Overall
## inq_last_6mths                    100.0000
## revol_util                         14.6770
## dti                                13.1020
## fico_range_high                    11.5631
## verification_statusSource Verified   2.4565
## purposesmall_business               2.3090
## term 60 months                      2.2409
## desc_empty1                         1.8831
## verification_statusVerified         1.6869
## purposecredit_card                  1.6640
## purposedebt_consolidation           1.2683
## purposemedical                      1.2456
## purposeother                        1.2055
## purposehome_improvement             1.1912
## purposehouse                        0.9306
## purposemajor_purchase               0.7762
## purposeeducational                  0.6415
## purposewedding                      0.5613
## purposemoving                       0.2707
## purposevacation                     0.2428
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  975 205
```

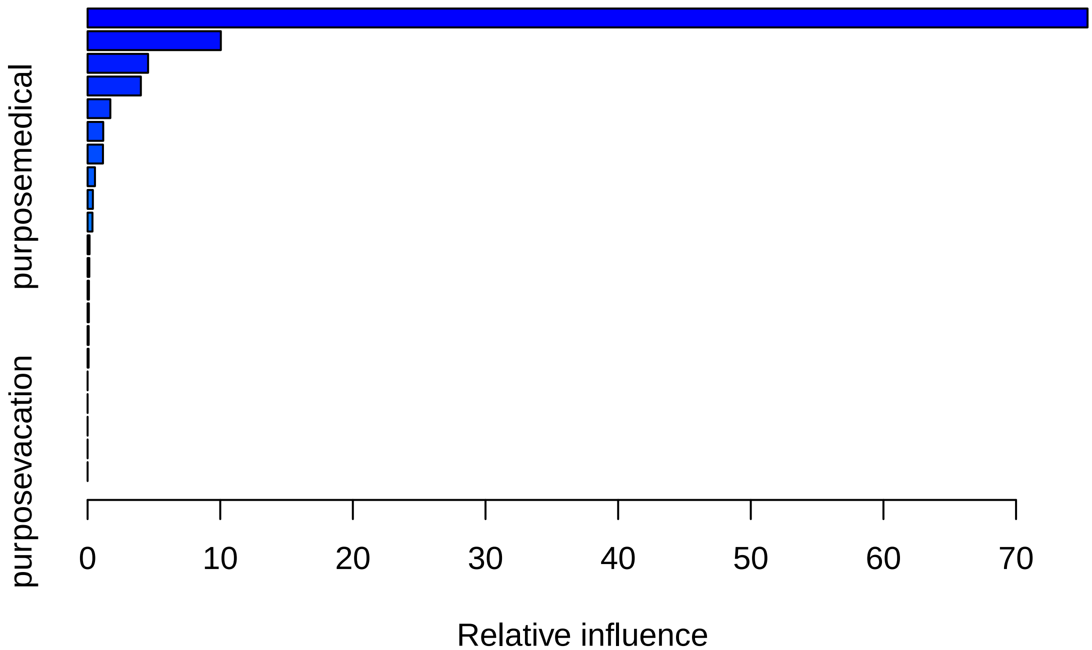
```
##       bad     1 125
##
##                Accuracy : 0.8423
##                  95% CI : (0.8213, 0.8616)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4749
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.37879
##             Specificity : 0.99898
##          Pos Pred Value : 0.99206
##          Neg Pred Value : 0.82627
##              Prevalence : 0.25268
##          Detection Rate : 0.09571
##    Detection Prevalence : 0.09648
##       Balanced Accuracy : 0.68888
##
##        '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: 0.7594
```

Gradient Boost Model

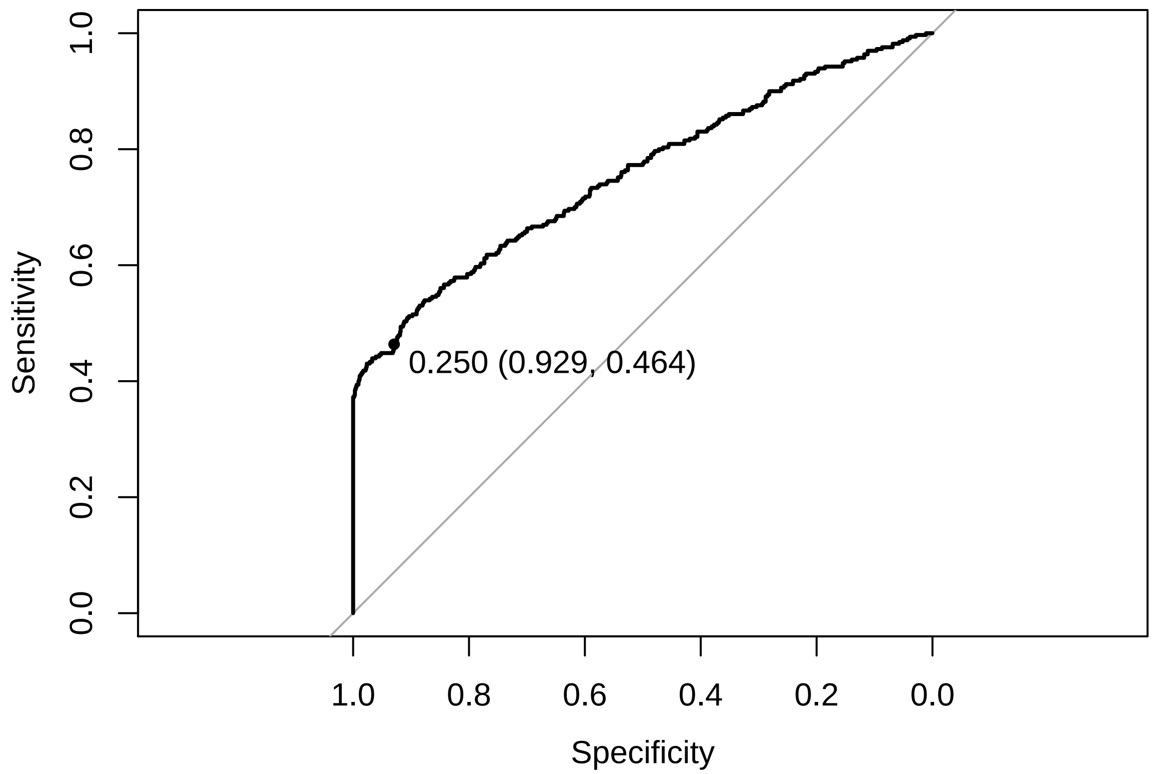
```
## Stochastic Gradient Boosting
##
## 3919 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 3919, 3919, 3919, 3919, 3919, 3919, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.8362721  0.4547099  0.008531325
##   1                  100      0.8360478  0.4555170  0.008456059
##   1                  150      0.8359370  0.4565208  0.008671268
##   2                   50      0.8370730  0.4543484  0.008788119
##   2                  100      0.8364379  0.4544259  0.009366213
##   2                  150      0.8360161  0.4550669  0.008445269
##   3                   50      0.8372903  0.4550478  0.008426846
##   3                  100      0.8361295  0.4545047  0.007967066
##   3                  150      0.8350416  0.4534665  0.007881488
##   Kappa SD
##   0.02301799
##   0.02293629
##   0.02438623
##   0.02347742
##   0.02562136
##   0.02361713
##   0.02378847
##   0.02295802
##   0.02297655
##
## 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
## fico_range_high                                            fico_range_high
## revol_util                                                      revol_util
## dti                                                                    dti
## purposesmall_business                                purposesmall_business
## term 60 months                                              term 60 months
## verification_statusSource Verified verification_statusSource Verified
## purposemedical                                              purposemedical
## verification_statusVerified                    verification_statusVerified
## purposecredit_card                                      purposecredit_card
## purposehouse                                                  purposehouse
## purposeother                                                  purposeother
## desc_empty1                                                    desc_empty1
## purposehome_improvement                            purposehome_improvement
## purposemajor_purchase                                purposemajor_purchase
## purposewedding                                              purposewedding
## purposedebt_consolidation                        purposedebt_consolidation
## purposeeducational                                      purposeeducational
## purposemoving                                                purposemoving
## purposerenewable_energy                            purposerenewable_energy
## purposevacation                                            purposevacation
##                                         rel.inf
## inq_last_6mths                      75.38963304
## fico_range_high                     10.04426241
## revol_util                           4.56059225
## dti                                  4.01426623
## purposesmall_business                1.71344338
## term 60 months                       1.17910405
## verification_statusSource Verified  1.15814242
## purposemedical                       0.55574332
## verification_statusVerified          0.39802162
## purposecredit_card                   0.36672265
```

```
## purposehouse                        0.14742591
## purposeother                        0.12793321
## desc_empty1                         0.09779792
## purposehome_improvement             0.09208653
## purposemajor_purchase               0.08000205
## purposewedding                      0.07482301
## purposedebt_consolidation           0.00000000
## purposeeducational                  0.00000000
## purposemoving                       0.00000000
## purposerenewable_energy             0.00000000
## purposevacation                     0.00000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                      Overall
## inq_last_6mths                     100.00000
## fico_range_high                     13.32313
## revol_util                           6.04936
## dti                                  5.32469
## purposesmall_business                2.27278
## term 60 months                       1.56401
## verification_statusSource Verified   1.53621
## purposemedical                       0.73716
## verification_statusVerified          0.52795
## purposecredit_card                   0.48644
## purposehouse                         0.19555
## purposeother                         0.16970
## desc_empty1                          0.12972
## purposehome_improvement              0.12215
## purposemajor_purchase                0.10612
## purposewedding                       0.09925
## purposevacation                      0.00000
## purposerenewable_energy              0.00000
## purposeeducational                   0.00000
## purposedebt_consolidation            0.00000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  971 201
##       bad     5 129
##
##                Accuracy : 0.8423
##                  95% CI : (0.8213, 0.8616)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4802
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.39091
##             Specificity : 0.99488
##          Pos Pred Value : 0.96269
```

```
##          Neg Pred Value : 0.82850
##              Prevalence : 0.25268
##          Detection Rate : 0.09877
##    Detection Prevalence : 0.10260
##       Balanced Accuracy : 0.69289
##
##        'Positive' Class : bad
##
```

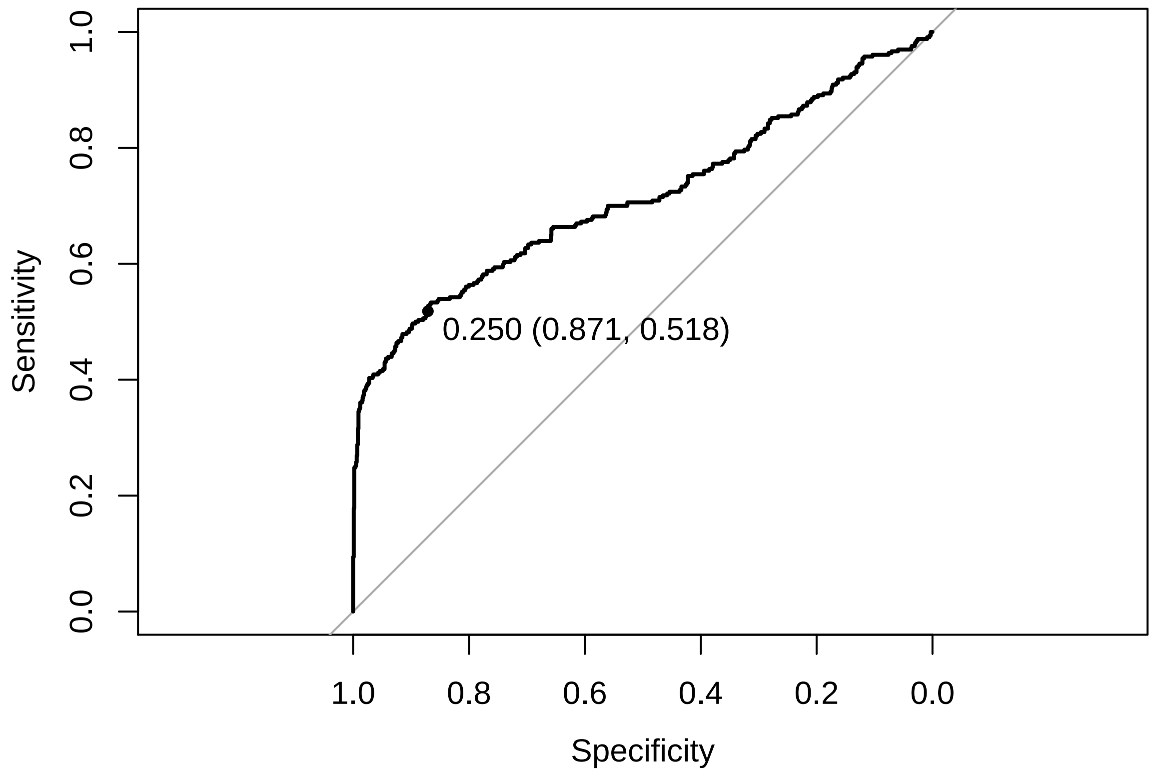



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 976 controls (dft_test$status good) < 330 cases (dft_test$status bad).
## Area under the curve: 0.7566
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 3919 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 3527, 3527, 3527, 3528, 3527, 3527, ...
```

```
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.8160258  0.4021399  0.01507231   0.05982590
##   0.50  0.8257222  0.4305123  0.01673570   0.06672544
##   1.00  0.8241910  0.4249018  0.01649474   0.06404348
##
## Tuning parameter 'sigma' was held constant at a value of 0.0562208
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.0562208 and C = 0.5.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                     Importance
## inq_last_6mths         100.000
## purpose                 38.107
## revol_util              37.528
## fico_range_high         29.879
## dti                     24.441
## term                    17.971
## desc_empty               8.837
## verification_status      0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  956 204
##       bad    20 126
##
##                Accuracy : 0.8285
##                  95% CI : (0.8069, 0.8486)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : 1.301e-12
##
##                   Kappa : 0.4431
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.38182
##             Specificity : 0.97951
##          Pos Pred Value : 0.86301
##          Neg Pred Value : 0.82414
##              Prevalence : 0.25268
##          Detection Rate : 0.09648
##    Detection Prevalence : 0.11179
##       Balanced Accuracy : 0.68066
##
##        '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: 0.7144
```

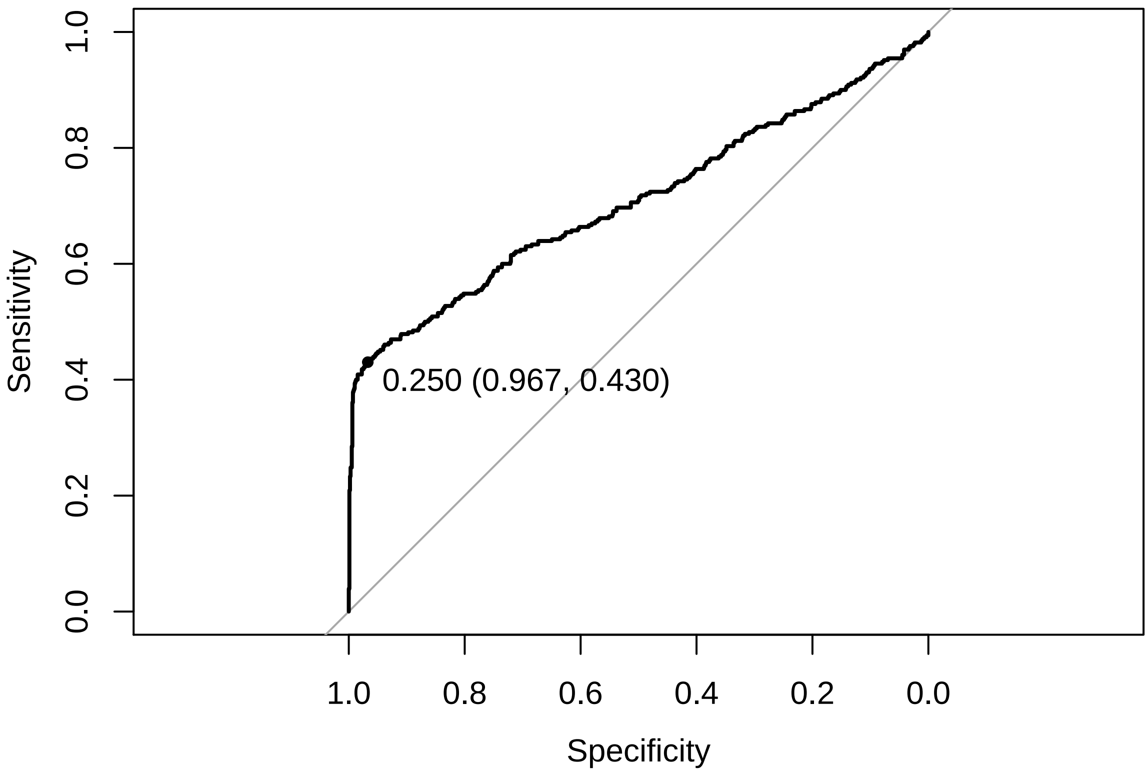
Neural Net Model

```
## Neural Network
##
## 3919 samples
##    8 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  0.7480047  0.00000000  0.010015870  0.00000000
##    1    0.0001000000  0.7516928  0.01952268  0.020822304  0.09761340
##    1    0.0003981072  0.7480047  0.00000000  0.010015870  0.00000000
##    1    0.0015848932  0.7909048  0.23374915  0.043463119  0.23023325
##    1    0.0063095734  0.7950709  0.25415281  0.044751563  0.23097544
```

```
##    1    0.0251188643  0.8054315  0.30706738  0.041489070  0.21588117
##    1    0.1000000000  0.8065198  0.30932314  0.040605822  0.21724274
##    3    0.0000000000  0.7516664  0.01924308  0.021199565  0.09621539
##    3    0.0001000000  0.7510316  0.01681626  0.017904807  0.08231580
##    3    0.0003981072  0.7513582  0.01760178  0.016736656  0.08460303
##    3    0.0015848932  0.8186408  0.40064981  0.029256086  0.12490857
##    3    0.0063095734  0.8201159  0.39171254  0.029200224  0.14910677
##    3    0.0251188643  0.8316046  0.45133657  0.008365264  0.02215051
##    3    0.1000000000  0.8286975  0.43219994  0.018508366  0.09288444
##    5    0.0000000000  0.7540829  0.03407987  0.023031256  0.11710293
##    5    0.0001000000  0.7514437  0.01719863  0.017551195  0.08599315
##    5    0.0003981072  0.7574881  0.05349042  0.029760707  0.14787280
##    5    0.0015848932  0.8262249  0.42877502  0.022881260  0.09325212
##    5    0.0063095734  0.8272593  0.42819183  0.018708405  0.09243742
##    5    0.0251188643  0.8278466  0.42984584  0.018575743  0.09243975
##    5    0.1000000000  0.8309561  0.44828891  0.008488499  0.02066506
##    7    0.0000000000  0.7514745  0.01869612  0.021497303  0.09348061
##    7    0.0001000000  0.7578189  0.05283301  0.027679272  0.14607220
##    7    0.0003981072  0.7641631  0.08817639  0.036458848  0.18030920
##    7    0.0015848932  0.8201535  0.39279091  0.031012615  0.15024157
##    7    0.0063095734  0.8298681  0.44505053  0.009207250  0.02719876
##    7    0.0251188643  0.8320125  0.45054702  0.010016655  0.02485425
##    7    0.1000000000  0.8331041  0.45343530  0.008605648  0.02324025
##    9    0.0000000000  0.7513591  0.01729011  0.017200131  0.08645055
##    9    0.0001000000  0.7511452  0.01819044  0.021320557  0.09095221
##    9    0.0003981072  0.8057343  0.31648497  0.037545095  0.20209627
##    9    0.0015848932  0.8276359  0.43631845  0.018181462  0.06014145
##    9    0.0063095734  0.8310256  0.44750878  0.010288537  0.02780237
##    9    0.0251188643  0.8312916  0.44866756  0.008525254  0.02218252
##    9    0.1000000000  0.8334961  0.45434413  0.008152155  0.02088718
##   11    0.0000000000  0.7543690  0.03506311  0.022770580  0.12155039
##   11    0.0001000000  0.7571886  0.04969359  0.024700787  0.13775329
##   11    0.0003981072  0.8240135  0.43561778  0.022647632  0.04528829
##   11    0.0015848932  0.8245680  0.42802112  0.027149795  0.08091973
##   11    0.0063095734  0.8313240  0.44912979  0.008719301  0.02435835
##   11    0.0251188643  0.8313275  0.45029157  0.008775793  0.02406260
##   11    0.1000000000  0.8331847  0.45494653  0.009068101  0.02445899
##   13    0.0000000000  0.7475608  0.01207566  0.010340406  0.06037828
##   13    0.0001000000  0.7554780  0.03814348  0.027181737  0.13134034
##   13    0.0003981072  0.8250546  0.42044100  0.022844478  0.10520535
##   13    0.0015848932  0.8272524  0.43004714  0.021432433  0.09200950
##   13    0.0063095734  0.8307558  0.44808661  0.009110397  0.02395384
##   13    0.0251188643  0.8321576  0.45155917  0.008869074  0.02248919
##   13    0.1000000000  0.8324660  0.45280839  0.008614003  0.02266017
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 9 and decay = 0.1.
## a 21-9-1 network with 208 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
##    0.04    2.90   -0.33   -0.94    0.60   -0.50    0.94   -0.45   -0.21
##  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1
##    2.75   -0.10   -0.20   -1.48    0.01    1.08   -0.07   -2.07   -0.01
## i18->h1 i19->h1 i20->h1 i21->h1
```

```
##    1.06   -1.45   -1.66    0.52
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##    0.00    0.17   -0.18   -0.13   -0.10   -0.07   -0.01    0.01   -0.01
##   i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##   -0.02    0.05   -0.01    0.08    0.00    0.05    0.00    0.01    0.00
## i18->h2 i19->h2 i20->h2 i21->h2
##    0.24    0.03   -0.08    0.17
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##    0.02    0.00    0.32    0.17    0.43    0.19   -0.01   -0.01    0.01
##   i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##    0.02    0.05   -0.01   -1.02    0.00    0.02    0.35    0.00    0.14
## i18->h3 i19->h3 i20->h3 i21->h3
##   -0.15   -0.93   -1.04   -0.13
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4
##   -0.59    1.82    1.53    1.50    1.22    0.74   -0.88    0.44   -1.91
##   i9->h4 i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4
##    1.31   -2.76   -1.84    1.27    0.00   -2.51   -1.61    1.26    0.03
## i18->h4 i19->h4 i20->h4 i21->h4
##   -6.59    0.01   -0.56    0.04
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##   i9->h5 i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.01
## i18->h5 i19->h5 i20->h5 i21->h5
##    0.00   -0.01    0.00    0.00
##   b->h6  i1->h6  i2->h6  i3->h6  i4->h6  i5->h6  i6->h6  i7->h6  i8->h6
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##   i9->h6 i10->h6 i11->h6 i12->h6 i13->h6 i14->h6 i15->h6 i16->h6 i17->h6
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.18
## i18->h6 i19->h6 i20->h6 i21->h6
##    0.00    0.01    0.00    0.00
##   b->h7  i1->h7  i2->h7  i3->h7  i4->h7  i5->h7  i6->h7  i7->h7  i8->h7
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##   i9->h7 i10->h7 i11->h7 i12->h7 i13->h7 i14->h7 i15->h7 i16->h7 i17->h7
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.02
## i18->h7 i19->h7 i20->h7 i21->h7
##    0.00    0.00    0.00    0.00
##   b->h8  i1->h8  i2->h8  i3->h8  i4->h8  i5->h8  i6->h8  i7->h8  i8->h8
##    0.00    0.05   -0.05   -0.03   -0.03   -0.02    0.00    0.00    0.00
##   i9->h8 i10->h8 i11->h8 i12->h8 i13->h8 i14->h8 i15->h8 i16->h8 i17->h8
##   -0.01    0.01    0.00    0.02    0.00    0.02    0.00    0.00    0.01
## i18->h8 i19->h8 i20->h8 i21->h8
##    0.07   -0.02   -0.02    0.05
##   b->h9  i1->h9  i2->h9  i3->h9  i4->h9  i5->h9  i6->h9  i7->h9  i8->h9
##    0.08    0.37   -0.16   -0.73   -0.06    0.37    0.00   -0.02   -0.13
##   i9->h9 i10->h9 i11->h9 i12->h9 i13->h9 i14->h9 i15->h9 i16->h9 i17->h9
##   -0.11    0.08   -0.10    0.02    0.03   -0.21   -0.01    0.13    0.00
## i18->h9 i19->h9 i20->h9 i21->h9
##   -0.54    0.08   -0.09   -0.06
##  b->o h1->o h2->o h3->o h4->o h5->o h6->o h7->o h8->o h9->o
##  0.72  2.37  0.71 -2.41 -4.33  0.79  0.01  0.72  0.72  1.48
## nnet variable importance
##
##   only 20 most important variables shown (out of 21)
```

```
##
##                                   Overall
## fico_range_high                   100.000
## inq_last_6mths                     47.244
## revol_util                         32.872
## term 60 months                     29.230
## verification_statusVerified        25.865
## desc_empty1                        22.521
## purposeother                       21.922
## dti                                21.527
## verification_statusSource Verified  20.904
## purposecredit_card                 16.943
## purposedebt_consolidation          15.026
## purposesmall_business              13.851
## purposemajor_purchase              12.938
## purposewedding                     10.196
## purposemedical                      9.553
## purposehouse                        6.334
## purposevacation                     5.996
## purposemoving                       4.605
## purposeeducational                  4.228
## purposehome_improvement             3.217
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  957 195
##       bad    19 135
##
##                Accuracy : 0.8361
##                  95% CI : (0.8149, 0.8558)
##     No Information Rate : 0.7473
##     P-Value [Acc > NIR] : 6.907e-15
##
##                   Kappa : 0.4731
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4091
##             Specificity : 0.9805
##          Pos Pred Value : 0.8766
##          Neg Pred Value : 0.8307
##              Prevalence : 0.2527
##          Detection Rate : 0.1034
##    Detection Prevalence : 0.1179
##       Balanced Accuracy : 0.6948
##
##        '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: 0.7093
```

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 50% and 55% 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.

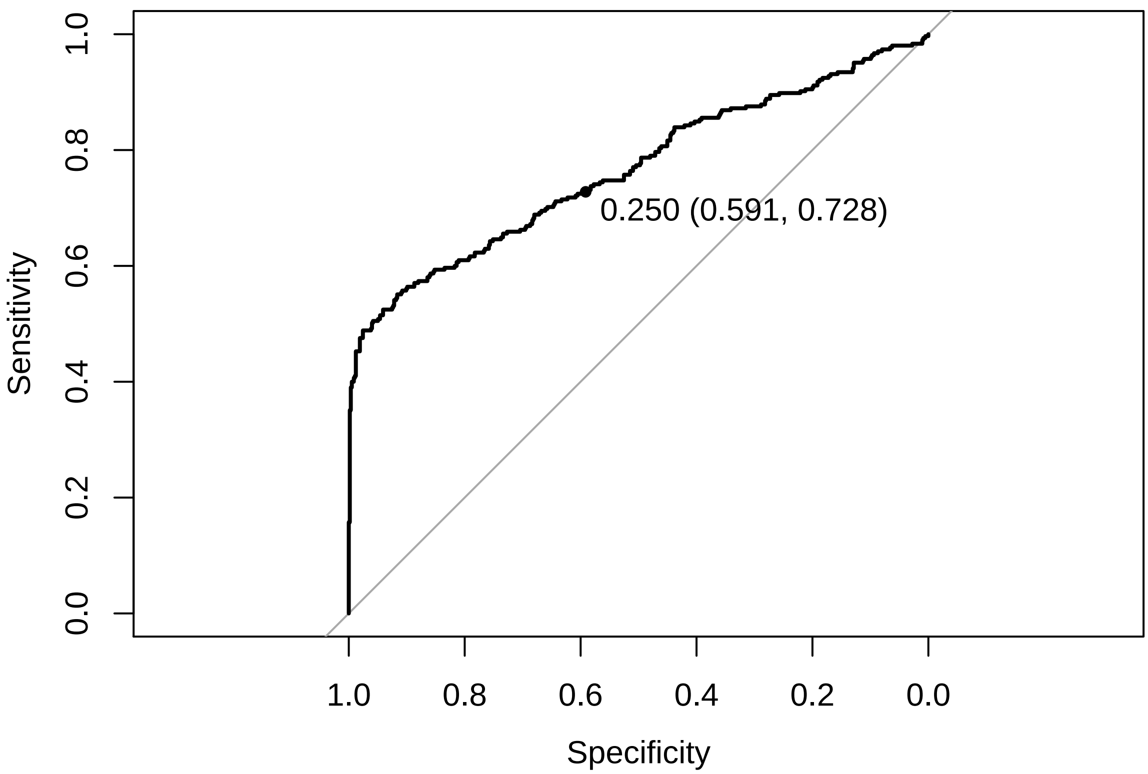
Logistic Regression Model

```
## Generalized Linear Model
##
## 2643 samples
##    8 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
##   0.7669394  0.4429243  0.0144435    0.03018417
##
##
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4377  -0.7808  -0.5590   0.7581   2.3274
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           1.491465   1.800121   0.829 0.407367
## `term 60 months`                      0.152443   0.125584   1.214 0.224797
## `verification_statusSource Verified` -0.485628   0.137047  -3.544 0.000395
## verification_statusVerified          -0.384243   0.113535  -3.384 0.000713
## purposecredit_card                    1.337500   0.414940   3.223 0.001267
## purposedebt_consolidation             0.982025   0.396301   2.478 0.013213
## purposeeducational                    2.020800   0.543584   3.718 0.000201
## purposehome_improvement               0.821894   0.439731   1.869 0.061612
## purposehouse                          1.379345   0.613116   2.250 0.024466
## purposemajor_purchase                 1.238372   0.443302   2.794 0.005214
## purposemedical                        1.733564   0.550579   3.149 0.001640
## purposemoving                         1.217768   0.581316   2.095 0.036185
## purposeother                          1.416774   0.415691   3.408 0.000654
## purposerenewable_energy               0.241555   1.111748   0.217 0.827994
## purposesmall_business                 1.455390   0.437421   3.327 0.000877
## purposevacation                       1.253586   0.775425   1.617 0.105955
## purposewedding                        0.481543   0.511241   0.942 0.346239
## fico_range_high                      -0.005888   0.002573  -2.288 0.022121
## inq_last_6mths                        0.625151   0.031230  20.018  < 2e-16
## revol_util                           -0.001596   0.001919  -0.832 0.405461
## desc_empty1                          -0.302932   0.122667  -2.470 0.013528
## dti                                   0.005366   0.007560   0.710 0.477793
##
## (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
## 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: 2709.9  on 2621  degrees of freedom
## AIC: 2753.9
##
## Number of Fisher Scoring iterations: 5
##
## glm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                     Overall
## inq_last_6mths                      100.000
## purposeeducational                   17.678
## `verification_statusSource Verified`  16.799
## purposeother                         16.116
## verification_statusVerified          15.995
## purposesmall_business                15.706
## purposecredit_card                   15.182
## purposemedical                       14.804
## purposemajor_purchase                13.011
## purposedebt_consolidation            11.417
## desc_empty1                          11.375
## fico_range_high                      10.459
## purposehouse                         10.265
## purposemoving                         9.482
## purposehome_improvement               8.342
## purposevacation                       7.067
## `term 60 months`                      5.033
## purposewedding                        3.660
## revol_util                            3.104
## dti                                   2.488
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  523 137
##       bad    52 168
##
##                Accuracy : 0.7852
##                  95% CI : (0.7566, 0.8119)
##     No Information Rate : 0.6534
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4926
##  Mcnemar's Test P-Value : 9.957e-10
##
##             Sensitivity : 0.5508
##             Specificity : 0.9096
##          Pos Pred Value : 0.7636
##          Neg Pred Value : 0.7924
##              Prevalence : 0.3466
##          Detection Rate : 0.1909
##    Detection Prevalence : 0.2500
##       Balanced Accuracy : 0.7302
##
##        'Positive' Class : bad
##
```



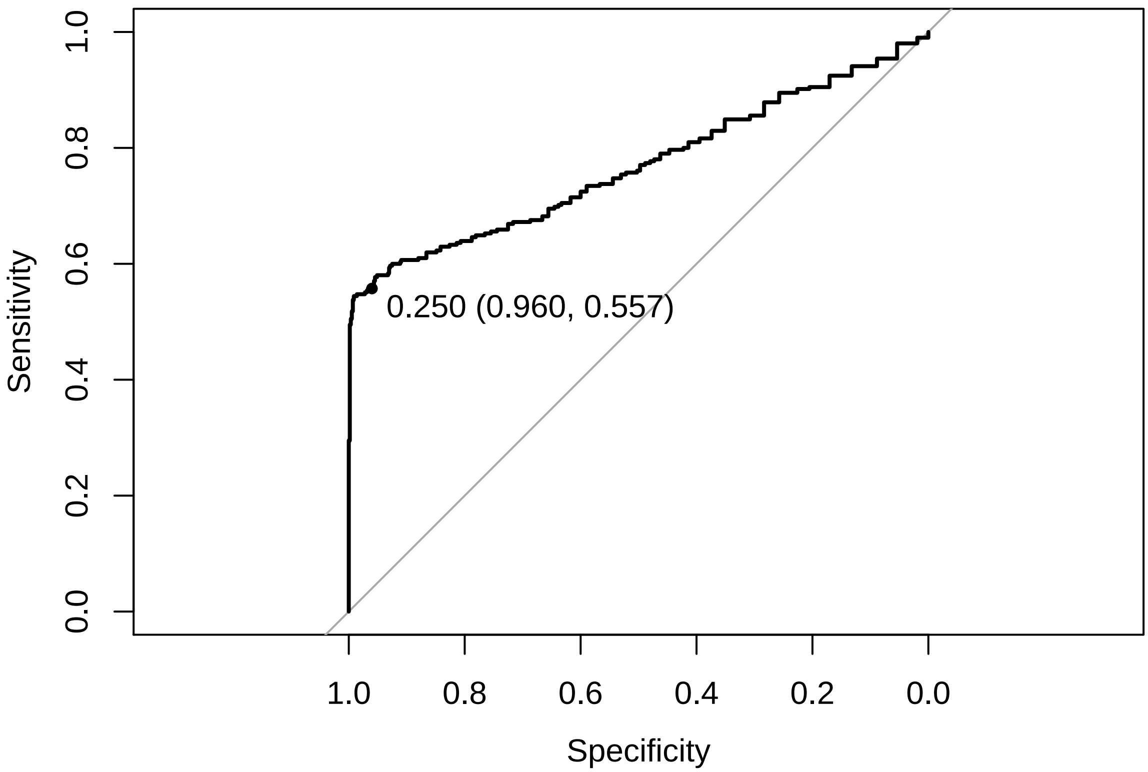

```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.7656
```

Random Forest Model

```
## Random Forest
##
```

```
## 2643 samples
##    8 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    0.8072887  0.5188201  0.01360084   0.03084039
##   11    0.7975190  0.5106143  0.01149480   0.02631682
##   21    0.7897465  0.4960485  0.01281513   0.03072646
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
##                 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        21   -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            21   -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 21)
##
##                                    Overall
## inq_last_6mths                    100.0000
## fico_range_high                    27.5088
## dti                                14.4767
## revol_util                         13.7005
## desc_empty1                         2.8735
## verification_statusVerified         2.7675
## verification_statusSource Verified  2.3282
## purposedebt_consolidation           2.2714
```

```
## term 60 months                          2.1481
## purposeeducational                      1.6575
## purposesmall_business                   1.3230
## purposecredit_card                      1.2363
## purposeother                            1.2314
## purposewedding                          0.9940
## purposehouse                            0.9853
## purposemedical                          0.9422
## purposemajor_purchase                   0.9149
## purposehome_improvement                 0.8528
## purposemoving                           0.4964
## purposevacation                         0.2773
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  571 144
##       bad     4 161
##
##                Accuracy : 0.8318
##                  95% CI : (0.8054, 0.856)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5838
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5279
##             Specificity : 0.9930
##          Pos Pred Value : 0.9758
##          Neg Pred Value : 0.7986
##              Prevalence : 0.3466
##          Detection Rate : 0.1830
##    Detection Prevalence : 0.1875
##       Balanced Accuracy : 0.7605
##
##        'Positive' Class : bad
##
```

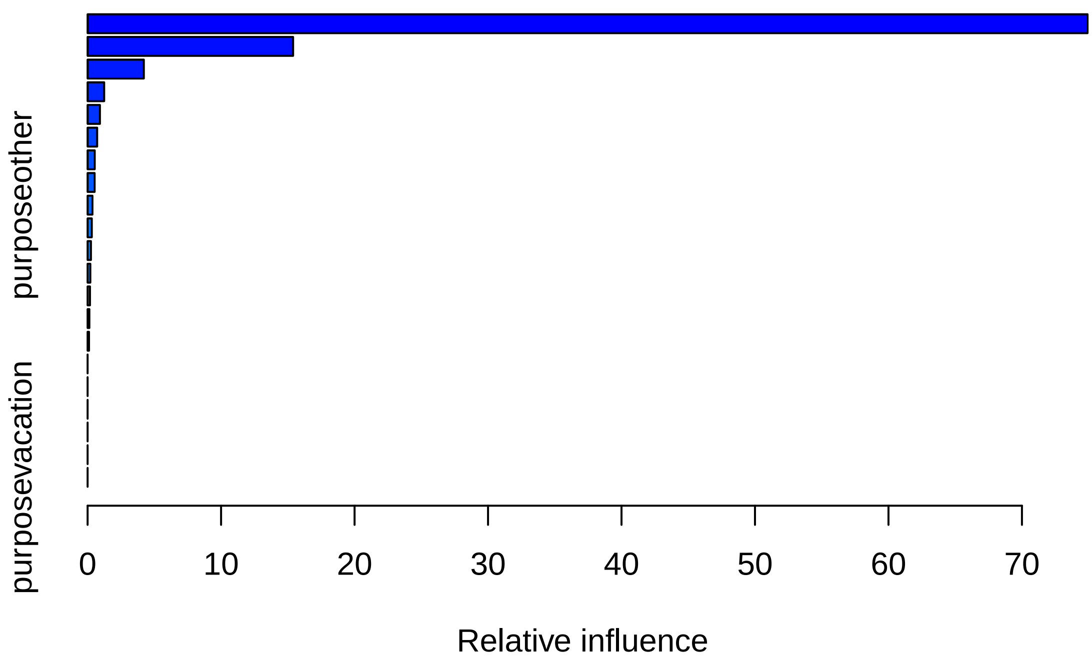



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.7722
```

Gradient Boost Model

```
## Stochastic Gradient Boosting
##
## 2643 samples
##    8 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      0.8099034  0.5283529  0.009035082
##   1                  100      0.8097057  0.5299870  0.008979183
##   1                  150      0.8082771  0.5278111  0.008679405
##   2                   50      0.8103259  0.5313183  0.008795803
##   2                  100      0.8078193  0.5275557  0.008633598
```

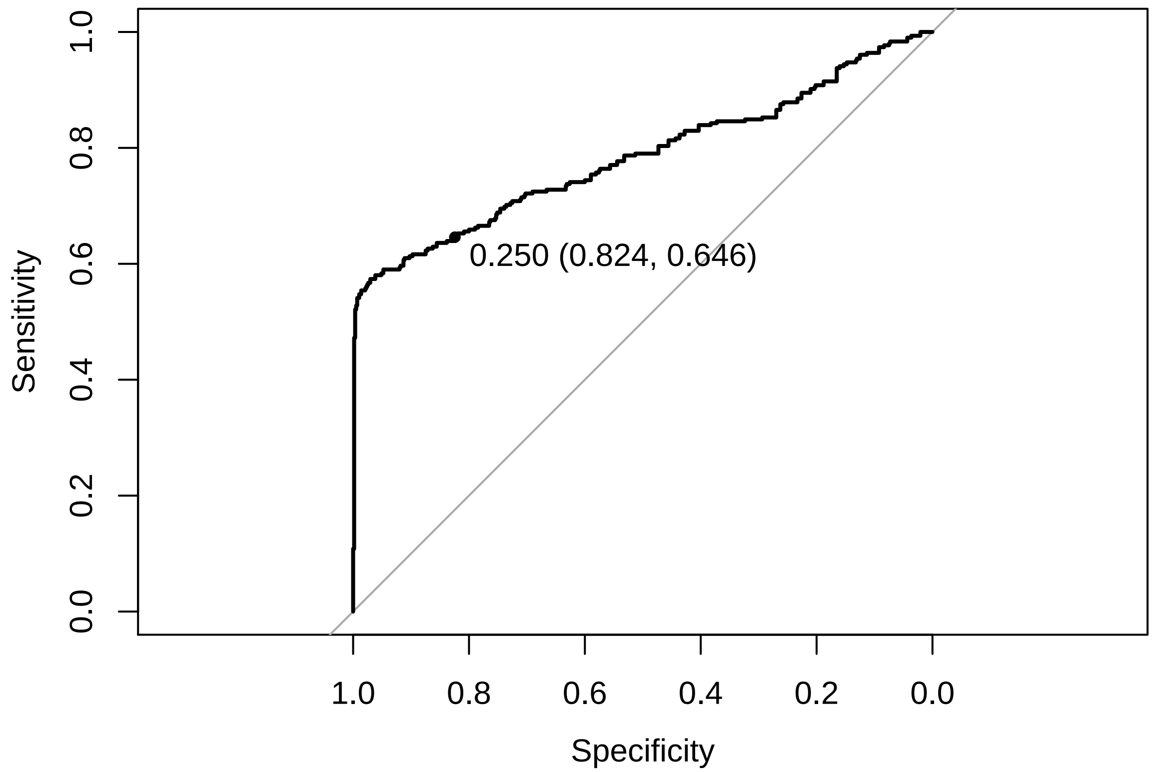
```
##   2                  150      0.8060178  0.5251464  0.008771310
##   3                   50      0.8099152  0.5310855  0.008975219
##   3                  100      0.8071769  0.5274579  0.009814729
##   3                  150      0.8052954  0.5251520  0.008925213
##   Kappa SD
##   0.02184971
##   0.02226121
##   0.02250329
##   0.02174921
##   0.02150534
##   0.02181398
##   0.02255440
##   0.02410728
##   0.02422920
##
## 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
## inq_last_6mths                                    inq_last_6mths
## fico_range_high                                  fico_range_high
## dti                                                          dti
## revol_util                                            revol_util
## desc_empty1                                          desc_empty1
## purposeeducational                            purposeeducational
## verification_statusVerified          verification_statusVerified
## purposedebt_consolidation              purposedebt_consolidation
## purposeother                                        purposeother
```

```
## purposewedding                                            purposewedding
## purposesmall_business                              purposesmall_business
## verification_statusSource Verified verification_statusSource Verified
## purposecredit_card                                    purposecredit_card
## purposehouse                                                purposehouse
## purposemedical                                            purposemedical
## term 60 months                                            term 60 months
## purposehome_improvement                          purposehome_improvement
## purposemajor_purchase                              purposemajor_purchase
## purposemoving                                              purposemoving
## purposerenewable_energy                          purposerenewable_energy
## purposevacation                                          purposevacation
##                                        rel.inf
## inq_last_6mths                      74.9147041
## fico_range_high                     15.3949208
## dti                                  4.2110905
## revol_util                           1.2406003
## desc_empty1                          0.9220962
## purposeeducational                   0.7149647
## verification_statusVerified          0.5324504
## purposedebt_consolidation            0.5256145
## purposeother                         0.3610020
## purposewedding                       0.3132003
## purposesmall_business                0.2497529
## verification_statusSource Verified  0.2036492
## purposecredit_card                   0.1786348
## purposehouse                         0.1313693
## purposemedical                       0.1059500
## term 60 months                       0.0000000
## purposehome_improvement              0.0000000
## purposemajor_purchase                0.0000000
## purposemoving                        0.0000000
## purposerenewable_energy              0.0000000
## purposevacation                      0.0000000
## gbm variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                      Overall
## inq_last_6mths                      100.0000
## fico_range_high                      20.5499
## dti                                   5.6212
## revol_util                            1.6560
## desc_empty1                           1.2309
## purposeeducational                    0.9544
## verification_statusVerified           0.7107
## purposedebt_consolidation             0.7016
## purposeother                          0.4819
## purposewedding                        0.4181
## purposesmall_business                 0.3334
## verification_statusSource Verified   0.2718
## purposecredit_card                    0.2385
## purposehouse                          0.1754
## purposemedical                        0.1414
```

```
## purposevacation                          0.0000
## purposerenewable_energy                  0.0000
## purposehome_improvement                  0.0000
## purposemajor_purchase                    0.0000
## term 60 months                           0.0000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  570 140
##       bad     5 165
##
##                Accuracy : 0.8352
##                  95% CI : (0.809, 0.8592)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.594
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5410
##             Specificity : 0.9913
##          Pos Pred Value : 0.9706
##          Neg Pred Value : 0.8028
##              Prevalence : 0.3466
##          Detection Rate : 0.1875
##    Detection Prevalence : 0.1932
##       Balanced Accuracy : 0.7661
##
##        'Positive' Class : bad
##
```

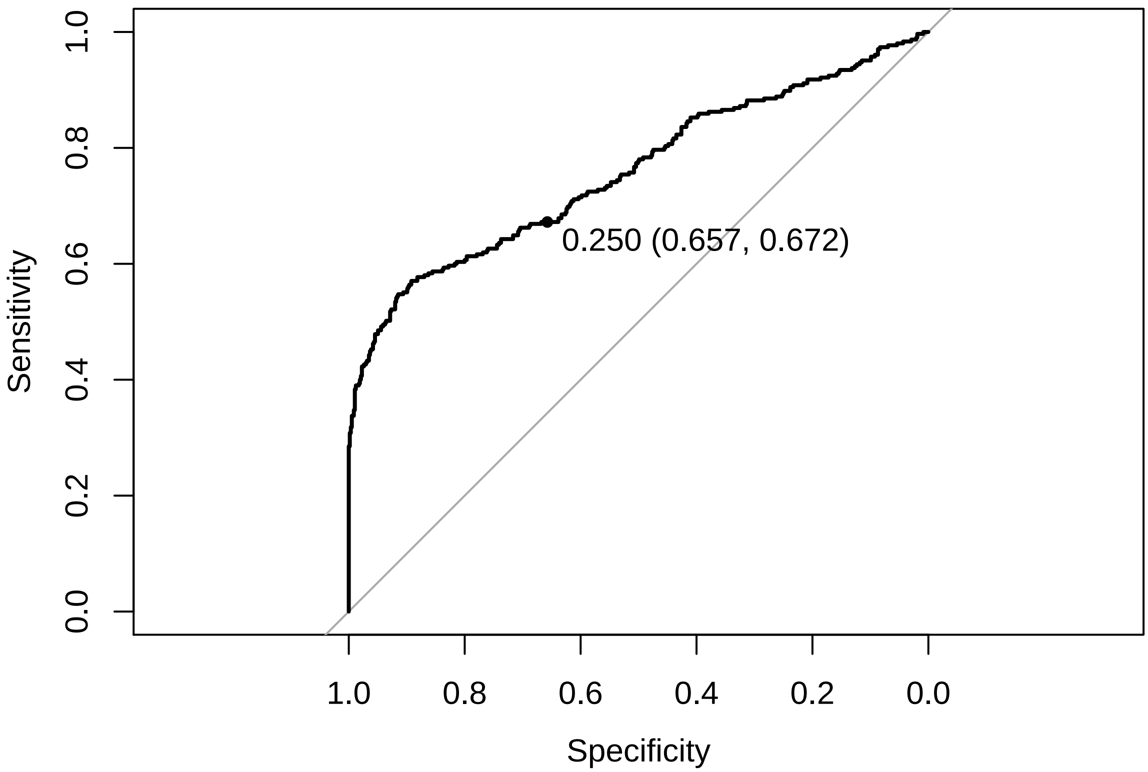



```
##
## Call:
## roc.default(response = dft_test$status, predictor = testProbs[,     "bad"])
##
## Data: testProbs[, "bad"] in 575 controls (dft_test$status good) < 305 cases (dft_test$status bad).
## Area under the curve: 0.786
```

SVM Model

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 2643 samples
##    8 predictor
##    2 classes: 'good', 'bad'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (10 fold)
##
## Summary of sample sizes: 2379, 2378, 2379, 2378, 2379, 2378, ...
##
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.25  0.7680597  0.4390726  0.02277914   0.05644214
##   0.50  0.7699551  0.4381526  0.01962623   0.04926353
##   1.00  0.7714874  0.4403406  0.02172485   0.05751002
##
## Tuning parameter 'sigma' was held constant at a value of 0.06062483
```

```
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were sigma = 0.06062483 and C = 1.
## Length  Class   Mode
##      1   ksvm     S4
## ROC curve variable importance
##
##                     Importance
## inq_last_6mths         100.000
## verification_status     40.336
## revol_util              34.723
## fico_range_high         28.638
## purpose                 27.408
## dti                     18.325
## term                     4.675
## desc_empty               0.000
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  541 154
##       bad    34 151
##
##                Accuracy : 0.7864
##                  95% CI : (0.7578, 0.813)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4803
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4951
##             Specificity : 0.9409
##          Pos Pred Value : 0.8162
##          Neg Pred Value : 0.7784
##              Prevalence : 0.3466
##          Detection Rate : 0.1716
##    Detection Prevalence : 0.2102
##       Balanced Accuracy : 0.7180
##
##        '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: 0.7585
```

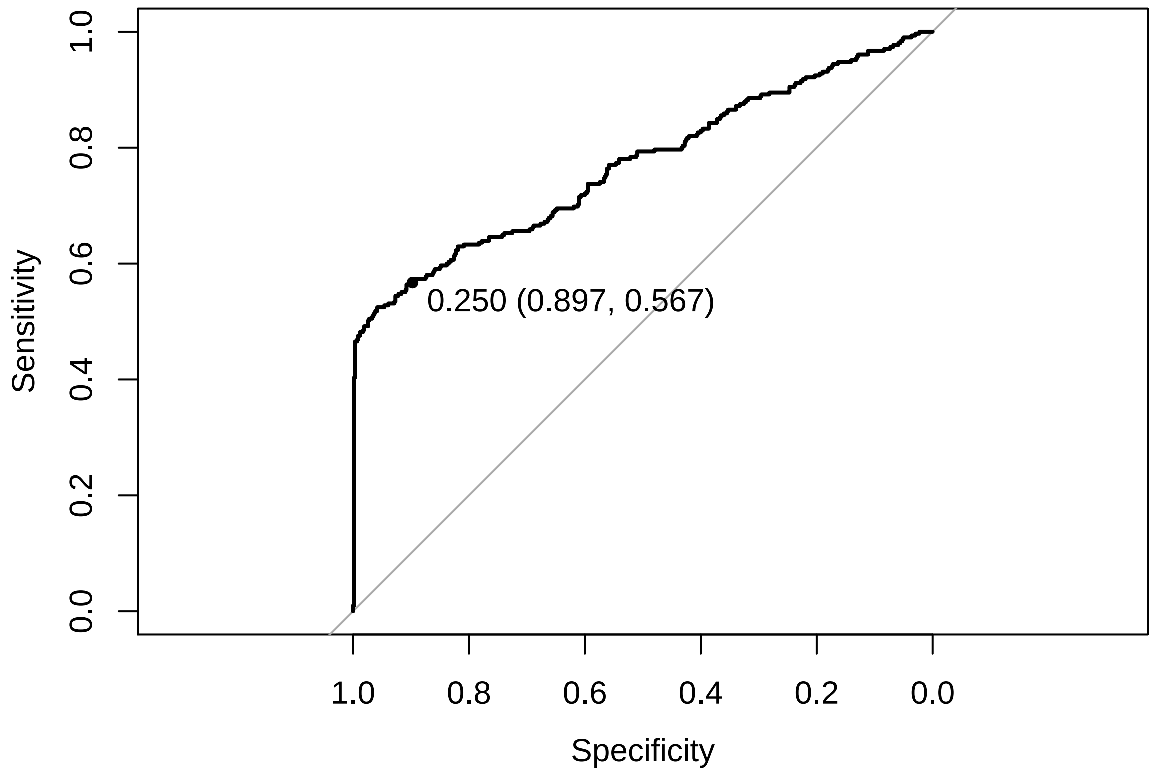
Neural Net Model

```
## Neural Network
##
## 2643 samples
##    8 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  0.6553773  -0.0001450387  0.01238213   0.0007251936
##    1    0.0001000000  0.6555418   0.0000000000  0.01206876   0.0000000000
##    1    0.0003981072  0.6555418   0.0000000000  0.01206876   0.0000000000
##    1    0.0015848932  0.7333902   0.2722616278  0.06400915   0.2287529041
##    1    0.0063095734  0.7523111   0.3347744631  0.05847237   0.2145769770
```

```
##    1    0.0251188643  0.7519539   0.3390896762  0.06591546   0.2172389166
##    1    0.1000000000  0.7837515   0.4489962047  0.02949431   0.0972592452
##    3    0.0000000000  0.6608487   0.0188889453  0.03059824   0.0944447263
##    3    0.0001000000  0.6613682   0.0193627255  0.02935757   0.0968136277
##    3    0.0003981072  0.6607323   0.0185113446  0.02985357   0.0925567229
##    3    0.0015848932  0.7726141   0.4173058468  0.03674919   0.1279318235
##    3    0.0063095734  0.7843429   0.4568207944  0.01634241   0.0335082214
##    3    0.0251188643  0.7868776   0.4649654121  0.01201741   0.0295795915
##    3    0.1000000000  0.7865019   0.4654653221  0.01129562   0.0289682310
##    5    0.0000000000  0.6669711   0.0379917175  0.03810899   0.1316480772
##    5    0.0001000000  0.6814472   0.0883949171  0.04887215   0.1804791893
##    5    0.0003981072  0.6918558   0.1272728579  0.05937417   0.2089224766
##    5    0.0015848932  0.7852134   0.4603308199  0.01265076   0.0292360378
##    5    0.0063095734  0.7772929   0.4354070352  0.02844671   0.0991062351
##    5    0.0251188643  0.7871518   0.4651495737  0.01124060   0.0278770432
##    5    0.1000000000  0.7875522   0.4682035851  0.01143192   0.0259327587
##    7    0.0000000000  0.6680740   0.0508587980  0.04042666   0.1421632685
##    7    0.0001000000  0.6763680   0.0741131612  0.05002900   0.1685590919
##    7    0.0003981072  0.7280194   0.2700421128  0.06974802   0.2255121847
##    7    0.0015848932  0.7809109   0.4445257854  0.03131738   0.0972768156
##    7    0.0063095734  0.7840379   0.4582542206  0.01564602   0.0323683364
##    7    0.0251188643  0.7862125   0.4631520456  0.01348706   0.0316789585
##    7    0.1000000000  0.7878547   0.4687511778  0.01330897   0.0321417230
##    9    0.0000000000  0.6709584   0.0560557447  0.04689750   0.1550263937
##    9    0.0001000000  0.6608072   0.0184989608  0.02761946   0.0924948038
##    9    0.0003981072  0.7751966   0.4277439471  0.03232257   0.1105113443
##    9    0.0015848932  0.7860965   0.4624767833  0.01250001   0.0281391454
##    9    0.0063095734  0.7874936   0.4646038954  0.01119326   0.0267149658
##    9    0.0251188643  0.7860750   0.4644388826  0.01280742   0.0286719106
##    9    0.1000000000  0.7862460   0.4651112831  0.01164034   0.0277631331
##   11    0.0000000000  0.6738889   0.0668656211  0.04128075   0.1576152100
##   11    0.0001000000  0.6658737   0.0367550559  0.03930048   0.1272127510
##   11    0.0003981072  0.7806713   0.4508187475  0.02097186   0.0357949447
##   11    0.0015848932  0.7846966   0.4557123718  0.01397236   0.0388314131
##   11    0.0063095734  0.7819397   0.4464257682  0.02703781   0.0966179937
##   11    0.0251188643  0.7854361   0.4636847771  0.01329389   0.0300522626
##   11    0.1000000000  0.7873730   0.4676942262  0.01211112   0.0264805433
##   13    0.0000000000  0.6651686   0.0344454146  0.03132068   0.1192241420
##   13    0.0001000000  0.6856509   0.1055543552  0.05322335   0.1917759872
##   13    0.0003981072  0.7723436   0.4309998406  0.03843899   0.0795214345
##   13    0.0015848932  0.7857123   0.4596262098  0.01119605   0.0287774704
##   13    0.0063095734  0.7861501   0.4617868856  0.01258371   0.0272395072
##   13    0.0251188643  0.7846294   0.4642836052  0.01526401   0.0333396509
##   13    0.1000000000  0.7852868   0.4637166188  0.01137387   0.0291163828
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were size = 7 and decay = 0.1.
## a 21-7-1 network with 162 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    0.04
## i18->h1 i19->h1 i20->h1 i21->h1
```

```
##    0.00    0.00    0.00    0.00
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.02
## i18->h2 i19->h2 i20->h2 i21->h2
##    0.00    0.01    0.00    0.00
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3
##   -0.54   -0.44   -0.70    3.07   -1.56    0.55   -1.95    1.43   -0.12
##  i9->h3 i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3
##    0.35    1.84   -1.81   -1.02    0.04    0.20    0.00   -0.19    0.01
## i18->h3 i19->h3 i20->h3 i21->h3
##   -0.05    1.20    0.40   -0.23
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4
##    0.00    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00
##  i9->h4 i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.53
## i18->h4 i19->h4 i20->h4 i21->h4
##    0.00    0.05    0.00    0.00
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5
##    0.16   -0.31    0.30    0.09    0.68   -0.05   -0.51   -0.64   -2.66
##  i9->h5 i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5
##    0.95    0.75   -1.72   -1.00    0.45   -0.40    0.20    0.30    0.02
## i18->h5 i19->h5 i20->h5 i21->h5
##   -4.48    0.00    1.68    0.01
##   b->h6  i1->h6  i2->h6  i3->h6  i4->h6  i5->h6  i6->h6  i7->h6  i8->h6
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h6 i10->h6 i11->h6 i12->h6 i13->h6 i14->h6 i15->h6 i16->h6 i17->h6
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.19
## i18->h6 i19->h6 i20->h6 i21->h6
##    0.00    0.05    0.00    0.01
##   b->h7  i1->h7  i2->h7  i3->h7  i4->h7  i5->h7  i6->h7  i7->h7  i8->h7
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
##  i9->h7 i10->h7 i11->h7 i12->h7 i13->h7 i14->h7 i15->h7 i16->h7 i17->h7
##    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.02
## i18->h7 i19->h7 i20->h7 i21->h7
##    0.00    0.01    0.00    0.00
##  b->o h1->o h2->o h3->o h4->o h5->o h6->o h7->o
##  1.39  1.03  1.52 -3.64  0.23 -4.60  1.39  1.39
## nnet variable importance
##
##   only 20 most important variables shown (out of 21)
##
##                                     Overall
## fico_range_high                    100.0000
## revol_util                          24.3975
## inq_last_6mths                      11.4453
## verification_statusVerified          9.6633
## dti                                  6.7289
## purposemoving                        6.1746
## desc_empty1                          6.1556
## purposecredit_card                   6.0739
## purposeeducational                   5.5001
## purposemedical                       4.9880
```

```
## purposehouse                        4.7840
## purposehome_improvement             4.5813
## purposeother                        4.4260
## purposedebt_consolidation           4.3538
## verification_statusSource Verified   4.2378
## purposemajor_purchase               2.4551
## purposesmall_business               1.6808
## term 60 months                      1.4569
## purposewedding                      1.3474
## purposerenewable_energy             0.6118
## Confusion Matrix and Statistics
##
##           Reference
## Prediction good bad
##       good  560 153
##       bad    15 152
##
##                Accuracy : 0.8091
##                  95% CI : (0.7815, 0.8346)
##     No Information Rate : 0.6534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5284
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4984
##             Specificity : 0.9739
##          Pos Pred Value : 0.9102
##          Neg Pred Value : 0.7854
##              Prevalence : 0.3466
##          Detection Rate : 0.1727
##    Detection Prevalence : 0.1898
##       Balanced Accuracy : 0.7361
##
##        '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: 0.7702
```