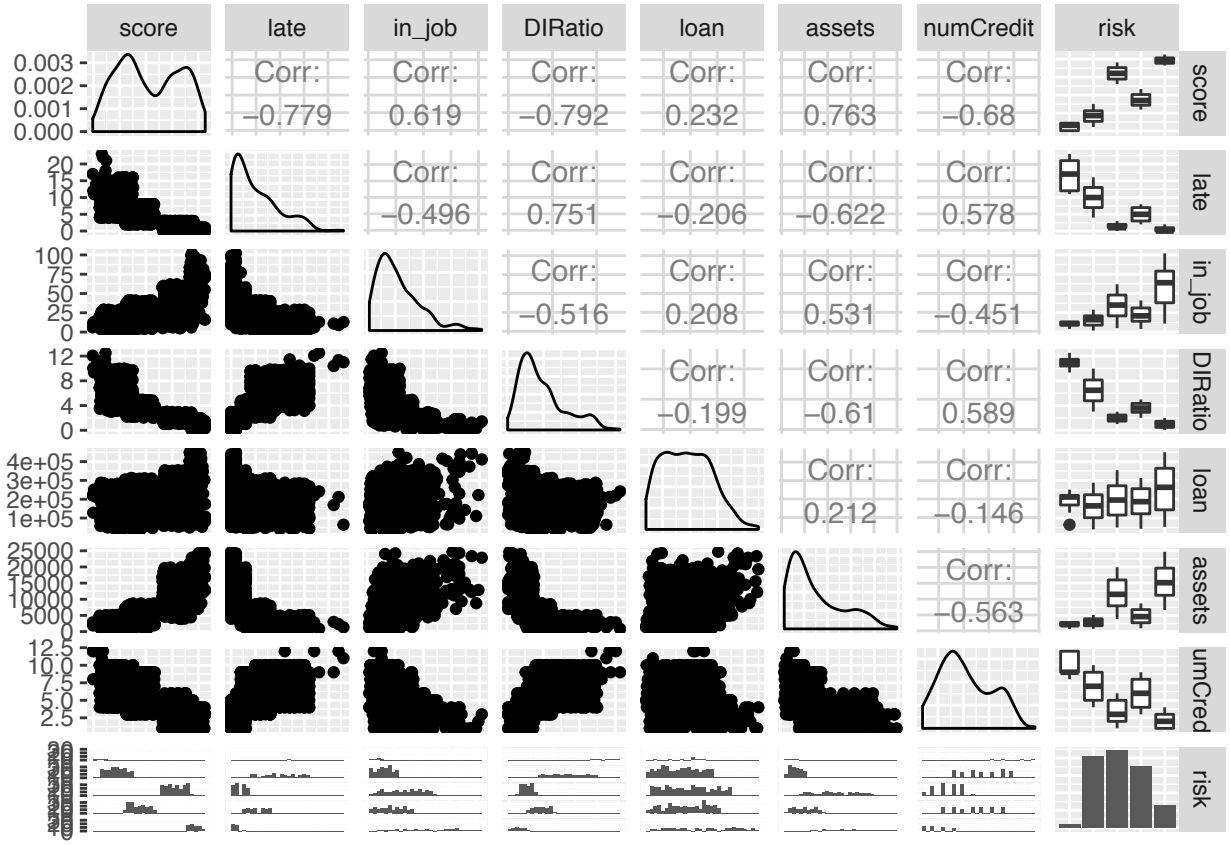
Neural Networks

Cody Frisby

2017-11-12

First, let's take a look at the data (I've removed the ID variable as it adds nothing to the analysis and renames the variables so that they aren't to long to fit on the plot).



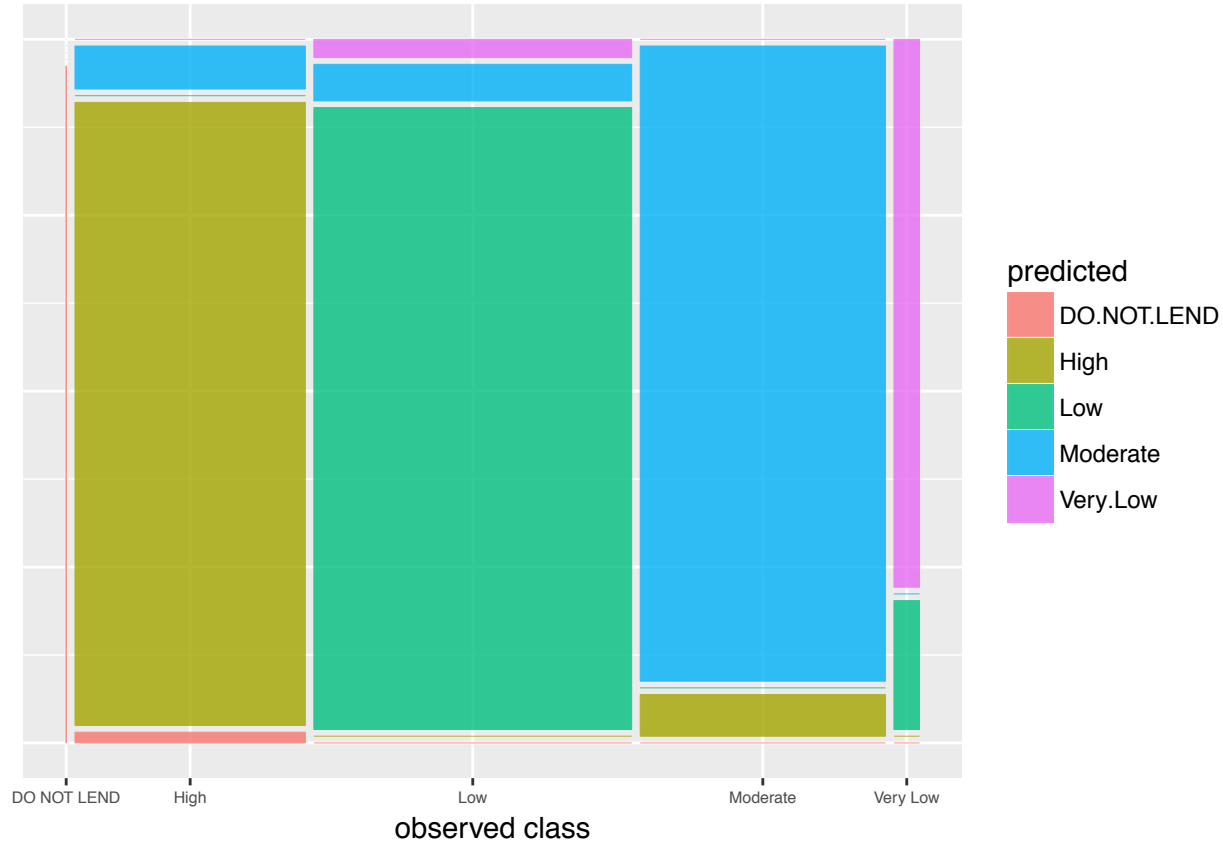
As can be seen, the scales vary vastly for the different variables. We should standardize them. I like the range method which

We can check out how well our neural net does by using the model to predict the probabilities of each class. We can then grab the class name of each observation by plucking off the name of the class of the largest probability. Comparing these in a confusion matrix, where the rows are the observed class and the columns are the predicted class, we get the results in the table below.

	DO.NOT.LEND	High	Low	Moderate	Very.Low
DO NOT LEND	3	0	0	0	0
High	1	58	0	4	0
Low	0	0	68	4	2
Moderate	0	4	0	61	0
Very Low	0	0	4	0	17

As we can see, we've miss-classified a few of the test set. We can compute the miss-class error by counting all those that aren't on the diagonal divided by the total number.

Doing this quickly in R we get a miss-class error of 0.0840708. Not too bad, on the first try. This can also be visualized by creating a mosaic plot. The colors of the plot represent the predicted class. The large solid colors show (for the most part) where we get it right. You can see that we get some of the predicted classes wrong, but overall the predicted classes agree with the observed ones. You can also see where the "DO NOT LEND" class and somewhat the "Very Low" class are very small in comparison to the others.



Earlier, I mentioned that we should standardize the variables. But is it even worth it. How does the model do if we don't standardize? Using the same techniques as before, where we pick the predicted class based on the class with the largest probability for each row of our test dataset, we get the results in the table below. As you can see, we do not have the same number of predicted classes!

	High	Low
DO NOT LEND	3	0
High	63	0
Low	45	29
Moderate	61	4
Very Low	14	7

Computing the error is a little different since we don't have a square confusion matrix. But, it's not too hard. The error is 0.5929204. Much, much worse AND we end up classifying all the predicitons into ONLY two classes. Looks like standardizing, at least in this case, paid off.

Appendix I

R Code

```
library(nnet)
## data reading
df <- read.csv("~/Documents/school/info3130/data/Chapter11Exercise_TrainingData.csv")
df <- df[-1]
newnames <- c("score", "late", "in_job", "DIRatio", "loan", "assets", "numCredit", "risk")
names(df) <- newnames
## scatterplot matrix:
GGally::ggpairs(df)
## first scale your data, excluding the class variable
df_range <- as.data.frame(apply(df[-8], 2,
                                function(x) x / diff(range(x))))
df_range$class <- df$risk
## get ready to split
set.seed(1234)
n <- dim(df_range)[1]
x <- seq(1, n)
s <- sample(x, n * (2/3), replace = FALSE)
####
train <- df_range[s, ]
test <- df_range[-s, ]
## model fitting and testing
fit <- nnet(class ~ ., data = train, size = 10, maxit = 1e5)
p <- predict(fit, test)
write.csv(p, "~/Documents/school/info3130/temp/precitions.csv",
          row.names = FALSE)
train2 <- df[s, ]
test2 <- df[-s, ]
fit2 <- nnet(risk ~ ., data = train2, size = 10, maxit = 1e5)
p2 <- predict(fit2, test2)
write.csv(p2, "~/Documents/school/info3130/temp/precitions2.csv",
          row.names = FALSE)
### Evaluating the first model
# read in the predicted probs
p <- read.csv("~/Documents/school/info3130/temp/precitions.csv")
## try to visualize our predictions compared to actual class
f <- function(x) {
  m <- which.max(x)
  n <- names(m)
}
classes <- apply(p, 1, f)
temp <- table(observed = test$class, predicted = classes)
knitr::kable(temp)
n <- dim(test)[1]
missed <- n - sum(diag(temp))
error <- missed / n
# visualizing the predictions compared to the observed classes
library(ggplot2)
library(ggmosaic)
temp2 <- data.frame(observed = test$class, predicted = classes)
```

```r
# add column that says where or not we predict correct
library(data.table)
dt <- data.table(temp2)
dt <- dt[, Freq := .N, by = observed]
gm <- ggplot(data = dt)
gm <- gm + geom_mosaic(aes(weight = Freq, x = product(observed), fill = predicted)) + xlab("observed cla
gm
## Evaluating the model where we didn't use standardized data
p2 <- read.csv("~/Documents/school/info3130/temp/precitions2.csv")
classes2 <- apply(p2, 1, f)
temp3 <- table(observed = test$class, predicted = classes2)
knitr::kable(temp3)
missed2 <- n - (63 + 29)
error2 <- missed2 / n
```