Appendix: R Code

```
library(kernlab)
library(pracma)
library(kknn)
library(caret)

d <- read.csv("./data-2.2/credit_card_data-headers.txt", sep = "\t" )

x <- data.matrix(within(d, rm(R1)))
# x <- data.matrix(d[, c("A9", "A15")])
y <- as.vector(unlist(d["R1"]))



Problem2_PartA <- function(make_plots=FALSE) {
  # wrapper function for ksvm boilerplate
  model_generator <- function(c) {
    ksvm(
      x,
      y,
      C=c,
      type="C-svc",
      kernel="vanilladot",
      scaled=TRUE,
    )
  }

  svm_plot_range <- function(c_min, c_max, filename=NULL, n = 20) {
    # Generate plot for range of C's
    c_range = logspace(log10(c_min), log10(c_max), n = n)
    error_range <- sapply(c_range, function(c) { model_generator(c)@err

    # create a chart
    if (is.null(filename)) {
      plot(c_range, error_range, log="x", xlab="C", ylab="loss")
    } else {
```

```
    png(filename=filename)
    plot(c_range, error_range, log="x", xlab="C", ylab="loss")
    dev.off()
  }
}

if (make_plots) {
  # Initial domain search
  svm_plot_range(filename = "svm_domain_search.png", c_min = 10^-4, c

  # Between 2*10^-3 and 5*10^2 there is a stable error floor
  svm_plot_range(filename = "svm_error_floor.png", c_min = 2*10^-3, c
}

# arbitrarily choosing C=10
model <- model_generator(c=10)

# copy from the homework assignment
a <- colSums(model@xmatrix[[1]] * model@coef[[1]])
a0 <- -model@b

print(a)
print(a0)

pred <- predict(model,x)
print(pred)

# create a confusion matrix
print(
  confusionMatrix(
    data=as.factor(pred),
    reference=as.factor(y)
  )
)
}

Problem2_PartB <- function(n_min, n_max, filename="knn_domain_search.png
  # wrapper function for single iteration of leave one out
```

```
knn_leave_one_out_no_test_set_iteration <- function(n, i) {
  model <- kknn(
    R1 ~ .,
    train = d[-i, ],
    test  = d[i, -ncol(d)],
    k = n,
    scale = TRUE
  )

  # return mean of nearest neighbor's R1
  # I'm not seeing a better syntax for this online
  # but it seems to
  fitted(model)
}

knn_leave_one_out_no_test_set <- function(n, confusion_matrix=FALSE) {
  probabilities <- sapply(
    seq(1,nrow(d)),
    function(i) { knn_leave_one_out_no_test_set_iteration(n, i) }
  )

  pred = round(probabilities)

  error <- mean(pred == d$R1)

  if (confusion_matrix) {
    # create a confusion matrix
    print(
      confusionMatrix(
        data=as.factor(pred),
        reference=as.factor(d$R1)
      )
    )
  }

  error
}
```

```
  knn_plot_range <- function(n_min, n_max, filename=NULL){
    n_seq = seq(n_min, n_max)
    error_range <- sapply(n_seq, knn_leave_one_out_no_test_set)

    if (is.null(filename)) {
      plot(n_seq, error_range, xlab="nearest neighbors", ylab="accuracy"
    } else {
      png(filename=filename)
      plot(n_seq, error_range, xlab="nearest neighbors", ylab="accuracy"
      dev.off()
    }
  }

  knn_plot_range(n_min, n_max, filename=filename)

  # 12 neighbors was the best classifier, so get the confusion matrix
  knn_leave_one_out_no_test_set(12, confusion_matrix = TRUE)
}
```