Pima Native Americans and Diabetes

Technical Appendix

Amalia, Cassidy, Daniel, Maggie

12/21/2018

Contents

Introduction

This is the technical appendix submitted for the Final Project portion of Statistics 320: Statistics Communications with Prof. Pamela Matheson. The assignment was to

(A) predict the probability that individual females have diabetes, and

(B) detect subsets of characteristics that are at higher risk of diabetes.

This technical appendix details the code we used in our analysis and the creation of the data visualizations that comprised our presentation.

- Presentation slides can be found here.
- The Shiny app can be found here
- The 3 page report can be found here.
- The 1 page handout can be found here.

Initial Loading

```
require(tidyverse); require(caret); require(pROC); require(ggpubr); require(gridExtra)
require(xtable); require(ROCR)
data <- read.delim("diabetes.txt", header = F, sep = ",") # assumes file is in the same directory
names(data) <- c("prg", "plasma", "bp", "thick", "insulin", "body", "pedigree", "age", "response")
```

```
data$response <- factor(data$response)
set.seed(1234)
```

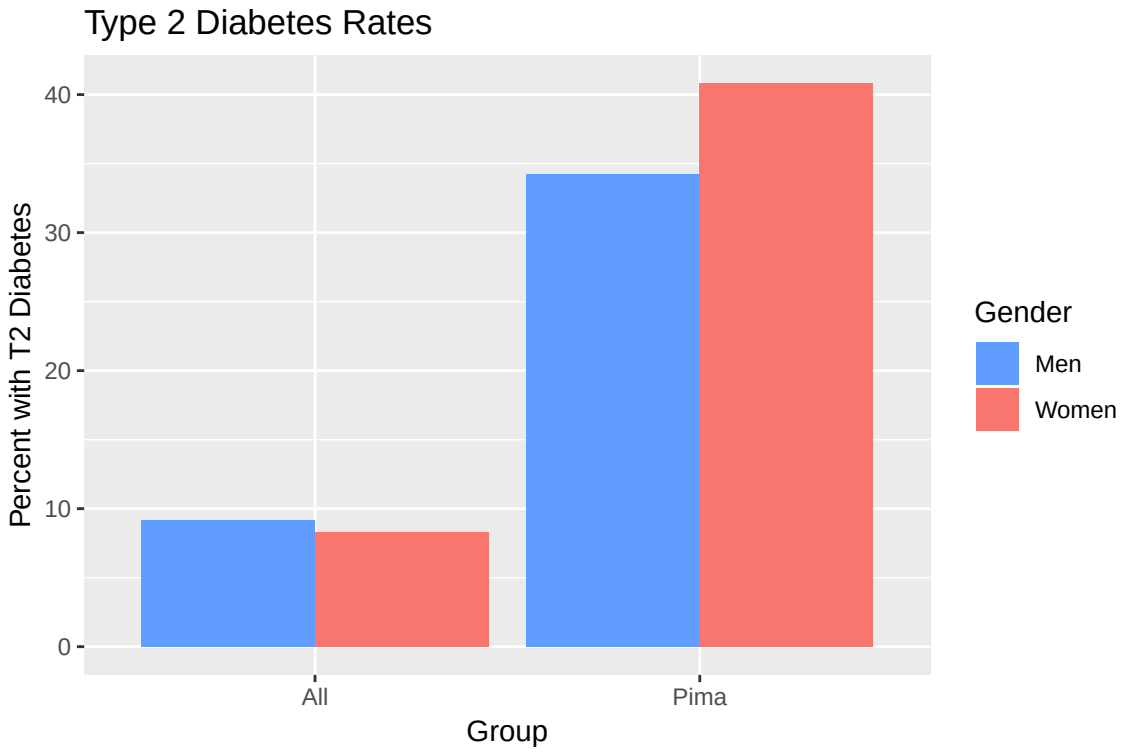
Preliminary Analysis

To begin, we looked at the response variable to confirm how many cases of type 2 diabetes there were in this sample. Our findings matched the handout: 268 out of 768 women in this sample were positive for type two diabetes. We then constructed two side by side histograms based on outside data in order to contextualize these findings.

```
round(table(data$response)/nrow(data),3)
```

```
##
##     0     1
## 0.651 0.349
```

```
data.frame("Prevalence" = c(34.2,40.8,9.2,8.3),
                  "Gender" = c("Men", "Women", "Men", "Women"),
                  "Group" = c("Pima","Pima","All","All")) %>%
ggplot(aes(x=Group,y=Prevalence,fill=factor(Gender))) +
  geom_bar(stat="identity",position="dodge") +
  scale_fill_manual(name="Gender",values=c("#619CFF", "#F8766D")) +
  xlab("Group") + ylab("Percent with T2 Diabetes") + ggtitle("Type 2 Diabetes Rates")
```



This graph revealed that our sample was relatively consistent with other estimates of diabetes prevalence among Pima Native American women. [1] Additionally, we were able to visualize how much higher the rate of

[1] "Effects of Traditional and Western Environments on Prevalence of Type 2 Diabetes in Pima Indians in Mexico and the U.S." 2006. Leslie Schulz, Peter Bennett, Eric Ravussin, Judith Kidd, Kenneth Kidd, Julian Esparza and Mauro E. Valencia. http://care.diabetesjournals.org/content/29/8/1866

type two diabetes was among Pima Native American people relative to the general population of the US as a whole.

Distributions of individual variables

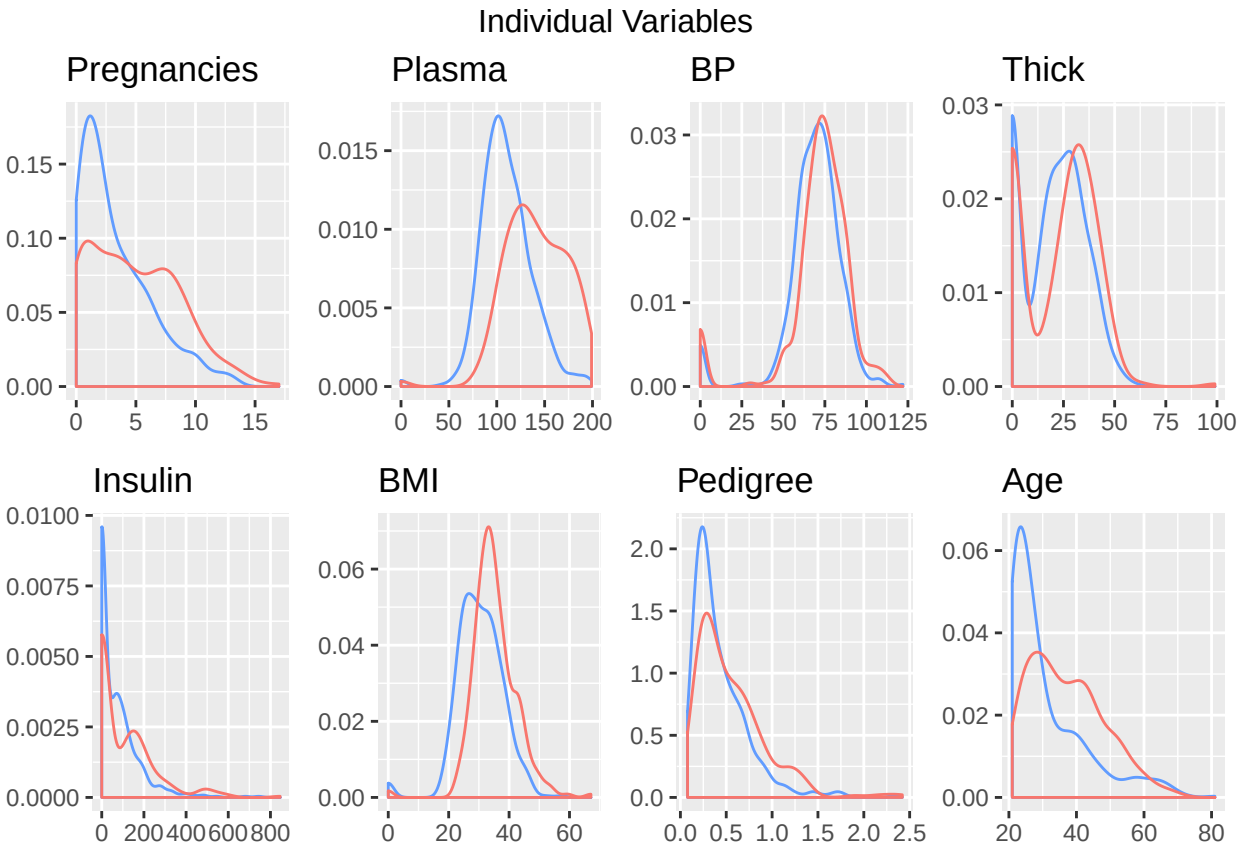
Next, we investigated each variable separately by looking at their individual distributions for the women in our sample with diabetes (`response=1`) and those without (`response=0`).

These plots were not used in the presentation, and are shown together (and without a legend) to save space. The red distributions refer to those who had diabetes.

```
gg <- theme(legend.position="none", axis.title.y = element_blank(),
            axis.title.x = element_blank())

p1 <- data %>% ggplot(aes(x=prg, group = response, color = response)) + geom_density() +
  ggtitle("Pregnancies") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p2 <- data %>% ggplot(aes(x=plasma, group = response, color = response)) + geom_density() +
  ggtitle("Plasma") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p3 <- data %>% ggplot(aes(x=bp, group = response, color = response)) + geom_density() +
  ggtitle("BP") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p4 <- data %>% ggplot(aes(x=thick, group = response, color = response)) + geom_density() +
  ggtitle("Thick") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p5 <- data %>% ggplot(aes(x=insulin, group = response, color = response)) + geom_density() +
  ggtitle("Insulin") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p6 <- data %>% ggplot(aes(x=body, group = response, color = response)) + geom_density() +
  ggtitle("BMI") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p7 <- data %>% ggplot(aes(x=pedigree, group = response, color = response)) + geom_density() +
  ggtitle("Pedigree") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))
p8 <- data %>% ggplot(aes(x=age, group = response, color = response)) + geom_density() +
  ggtitle("Age") + gg + scale_color_manual(values=c("#619CFF", "#F8766D"))

grid.arrange(grobs=list(p1,p2,p3,p4,p5,p6,p7,p8), nrow=2, common.legend=T,
             legend.position="bottom", top="Individual Variables")
```



Presentation Visualizations

In our presentation, we provided visualizations of the variables we later found to be the most influential in our model. To do this we split the sample into two groups: women who were positive for type 2 diabetes and women who were negative. Then, we constructed overlapping histograms for each of the four variables we were interested in so that we could see how the distributions for each variable differed between the two groups. Finally we added vertical lines color coded for the appropriate group representing the average value.

This is similar to what we did in the preliminary analysis section; the only difference (in addition to narrowing down the number of variables we presented) is we made the graphs more visually appealing and easy to understand for someone unfamiliar with the dataset, i.e. we can see that in all four cases, the red line (diabetes positive) is further to the right than the blue line (diabetes negative).

The code for this is below. Note x-axes were manually set, and cut off some outliers. This was done in order to make the difference in means appear as clearly as possible in the presentation.

```
d2<-filter(data, response==1)%>%mutate(Diabetes=factor(ifelse(response==1,
                                                    "Positive", NA)))
d3<-filter(data, response==0)%>%mutate(Diabetes=factor(ifelse(response==0,
                                                    "Negative", NA)))

p_1 <- ggplot(d2,aes(x=plasma)) +
  geom_histogram(data=d2,aes(x=plasma, fill=Diabetes), alpha = 0.7)+
  geom_histogram(data=d3,aes(x=plasma, fill=Diabetes), alpha = 0.7) +
  theme(plot.title=element_text(hjust=0.5)) +
  labs(x="Plasma Glucose Concentration", y="Number of People", title="Plasma") +
  theme(title = element_text(size=16), legend.position = "bottom") +
```

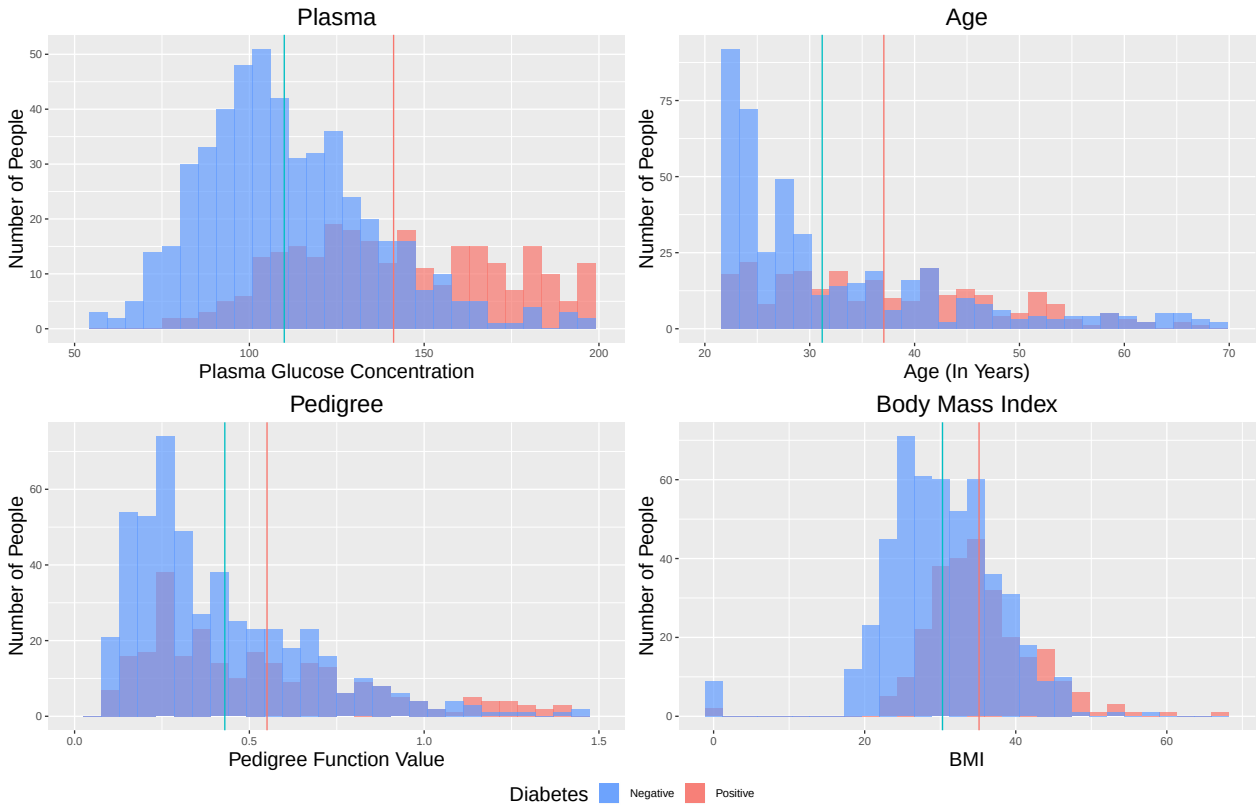
```
  scale_fill_manual(values=c("#619CFF", "#F8766D")) +
  geom_vline(aes(xintercept =mean(d2$plasma), col="#619CFF"), show.legend=F)+
  geom_vline(aes(xintercept =mean(d3$plasma), col="#F8766D"), show.legend=F) +
  scale_x_continuous(limits=c(50,200))

p_2 <- ggplot(d2,aes(x=age)) +
  geom_histogram(data=d2,aes(x=age, fill=Diabetes), alpha = 0.7)+
  geom_histogram(data=d3,aes(x=age, fill=Diabetes), alpha = 0.7) +
  theme(plot.title=element_text(hjust=0.5)) +
  labs(x="Age (In Years)", y="Number of People", title="Age")+
  theme(title = element_text(size=16), legend.position="bottom")+
  scale_fill_manual(values=c("#619CFF", "#F8766D")) +
  geom_vline(aes(xintercept =mean(d2$age), col="#619CFF"), show.legend=F)  +
  geom_vline(aes(xintercept =mean(d3$age), col="#F8766D"), show.legend=F)  +
  scale_x_continuous(limits=c(20,70))

p_3 <- ggplot(d2,aes(x=pedigree)) +
  geom_histogram(data=d2,aes(x=pedigree, fill=Diabetes), alpha = 0.7)+
  geom_histogram(data=d3,aes(x=pedigree, fill=Diabetes), alpha = 0.7)+
  theme(plot.title=element_text(hjust=0.5)) +
  labs(x="Pedigree Function Value", y="Number of People", title="Pedigree")+
  theme(title = element_text(size=16), legend.position = "bottom")+
  scale_fill_manual(values=c("#619CFF", "#F8766D")) +
  scale_x_continuous(limits=c(0,1.5)) +
  geom_vline(aes(xintercept =mean(d2$pedigree), col="#619CFF"), show.legend=F)  +
  geom_vline(aes(xintercept=mean(d3$pedigree), col="#F8766D"), show.legend=F)

p_4 <- ggplot(d2,aes(x=body)) +
  geom_histogram(data=d2,aes(x=body, fill=Diabetes), alpha = 0.7) +
  geom_histogram(data=d3,aes(x=body, fill=Diabetes), alpha = 0.7) +
  theme(plot.title=element_text(hjust=0.5)) +
  labs(x="BMI", y="Number of People", title="Body Mass Index") +
  scale_fill_manual(values=c("#619CFF", "#F8766D")) +
  geom_vline(aes(xintercept =mean(d2$body), col="#619CFF"), show.legend=F) +
  geom_vline(aes(xintercept=mean(d3$body), col="#F8766D"), show.legend=F) +
  theme(title = element_text(size=16), legend.position = "bottom")

ggarrange(p_1, p_2, p_3, p_4, ncol=2, nrow=2, common.legend = T, legend= "bottom")
```



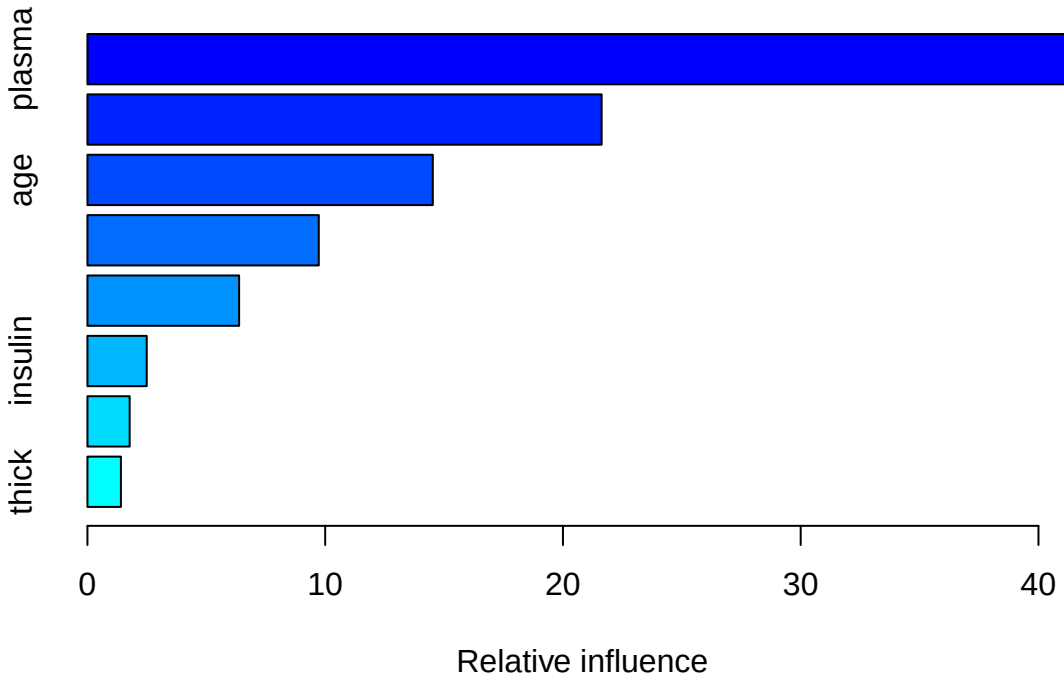
Model

We used `caret`'s `gbm()` (gradient boost model) function to initially create a classifier on all the variables available. Training options consisted of 5-fold cross-validation with a 0.75 train-test split. Given the function's random nature, a seed was set in order to ensure reproducibility. In the presentation we referred to this as the Full Model.

Full Model

A summary of the model object gives us the relative influences of the variables.

```
summary(objModel)
```




```
##               var   rel.inf
## plasma     plasma 42.058613
## body         body 21.627825
## age           age 14.526647
## pedigree pedigree  9.729920
## prg           prg  6.379842
## insulin   insulin  2.493085
## bp             bp  1.775946
## thick       thick  1.408122
```

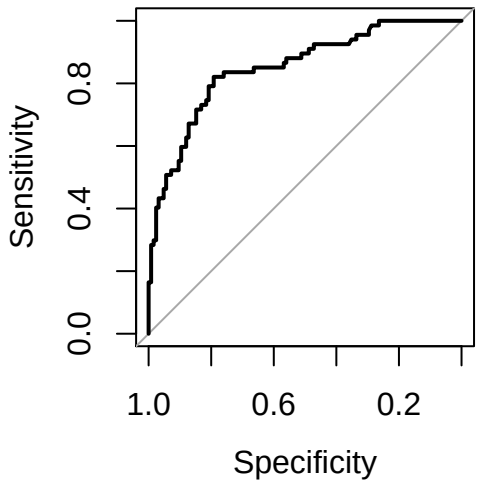
The values from this summary were included in the presentation in `ggplot` form manually for aesthetic purposes. The code to reproduce it is below. Because it would be redundant to display it again, the output is supressed.

```
df <- data.frame("var"=c("Plasma","BMI","Age","Family","Preg","Insulin","BP","Triceps"),
                 "val" =c(42,21.7,14.5,9.7,6.4,2.5,1.8,1.4))
df$var <- factor(df$var, levels(df$var) <-
                   rev(c("Plasma","BMI","Age","Family","Preg","Insulin","BP","Triceps")) )
df$col <- c(rep(1,4),rep(0,4))

df %>%
  ggplot(aes(factor(var), val, fill=factor(col))) + geom_bar(stat='identity') +
  labs(title="Relative Influence",y="",x="") +
  scale_fill_manual(values=c("grey", "#F8766D")) +
  theme(title=element_text(size=30), axis.text.y = element_text(size=16), legend.position="none") +
  coord_flip()
```

We make predictions on the text set, plot an AUC curve, print the AUC, and the some overall metrics:

```
predictions <- predict(object=objModel, testDF[,predictorsNames], type='prob')
auc <- roc(testDF$response2, predictions[,2])
plot(auc); auc$auc
```




```
## Area under the curve: 0.8506
```

```
pred<-factor(ifelse(predictions$Y>0.50,"Y","N"),levels=c("N","Y"))
conf<-confusionMatrix(pred,testDF$response2)
conf$overall; conf$byClass
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   0.7916666667   0.5382950583   0.7273004777   0.8467961093   0.6510416667
## AccuracyPValue  McnemarPValue
##   0.0000151339   0.8743670612
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.8480000            0.6865672            0.8346457
##       Neg Pred Value            Precision               Recall
##            0.7076923            0.8346457            0.8480000
##                   F1           Prevalence       Detection Rate
##            0.8412698            0.6510417            0.5520833
## Detection Prevalence    Balanced Accuracy
##            0.6614583            0.7672836
```

Evaluating the Full Model

We find that the full variable model has an AUC of 85%, overall accuracy of 79%, sensitivity of 85%, and specificity of 69%. Though the specificity is not as high as we might like, it is worth noting that the positive predictive value of 83% is not too far from 85%.

This full variable model is implemented in the Shiny app linked in the Introduction section. Though this satisfied Task (A), we recognized that Task (B) would be hard to complete with 8-variable model. For example, while an x-y plot is possible with two variables, and with three variables, a 3-D plot is possible, visualizing the interaction of more than three variables is impossible.

Therefore we also built a two-variable model, which we referred to in the presentation as the Field Model.

Further Evaluation

Although we did not mention it in the presentation, we also did analysis on sensitivity-specificity tradeoffs for different decision thresholds. We opted instead to use 0.5 as our decision threshold in order to avoid overcomplicating the presentation.

We did, however, note that choosing different cutoffs could more specifically suit the goals of our diabetes predictive model, especially for a field model, i.e. maximizing sensitivity at the cost of specificity by choosing a lower cutoff. This is a valid approach, especially when the disease/what is being classified is especially prevalent.

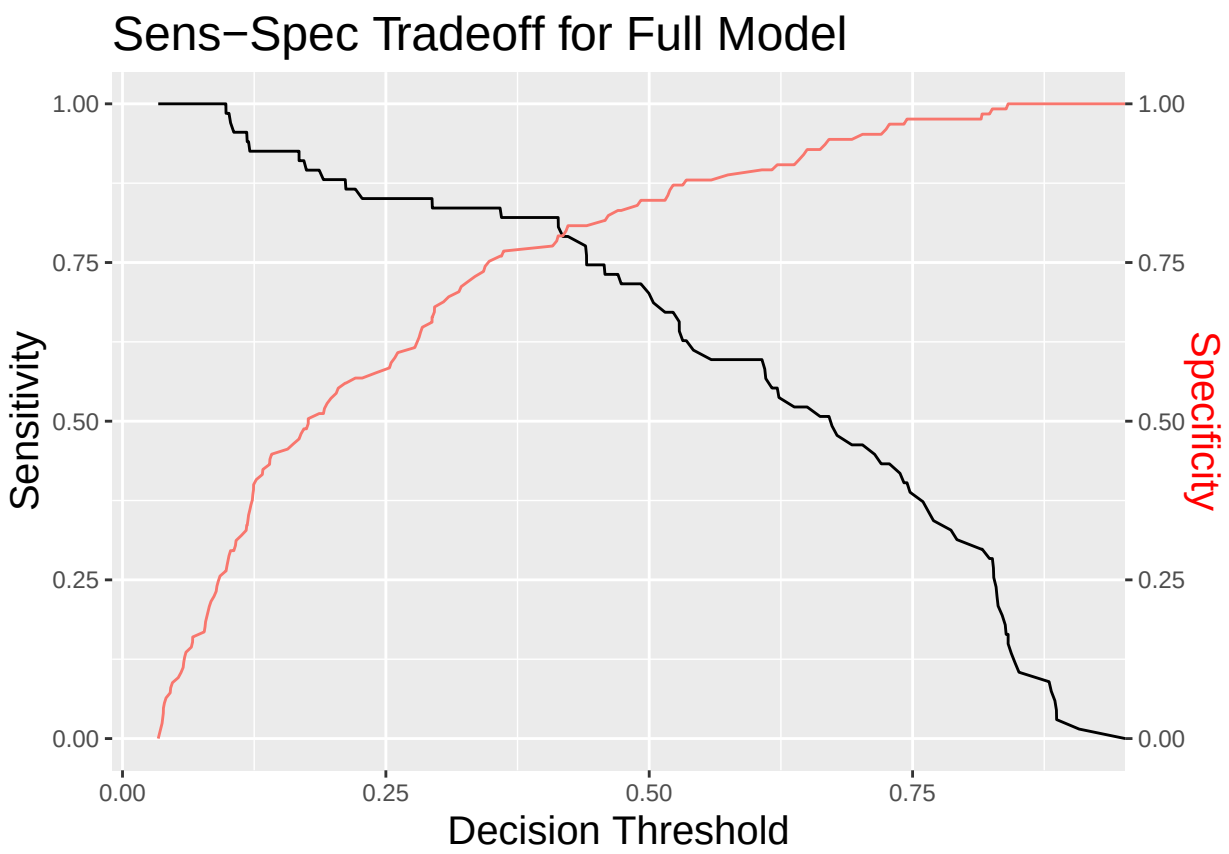
"According to the proposed method, the cutoff value is higher in places where the disease is less prevalent... For example, the cut-off value for a group of athletes exercising (higher risk/prevalence of dehydration) should be lower than that in general population."[2]

```
predictions <- prediction(predictions$Y,testDF$response2)

sens <- data.frame(x=unlist(performance(predictions, "sens")@x.values),
                   y=unlist(performance(predictions, "sens")@y.values))
spec <- data.frame(x=unlist(performance(predictions, "spec")@x.values),
                   y=unlist(performance(predictions, "spec")@y.values))

gg2 <- theme(axis.title.y.right = element_text(colour = "red"), legend.position="none",
        title = element_text(size=15))

sens %>% ggplot(aes(x,y)) +
  geom_line() +
  geom_line(data=spec, aes(x,y,col="red")) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Specificity")) +
  labs(x='Decision Threshold', y="Sensitivity") +
  ggtitle("Sens-Spec Tradeoff for Full Model") +
  gg2
```



[2]"On determining the most appropriate test cut-off value: the case of tests with continuous results." 2016. Farrokh Habibzadeh, Parham Habibzadeh and Mahboobeh Yadollahie. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5082211/

Field Model

Although `Blood Plasma` was the most significant variable in the relative influence graph earlier, because it obtaining it requires lab work done on a blood sample taken after an 8 hour fast, we decided to choose the two easiest to obtain but influential variables. These were `BMI` and `Age`. This was done so that, in addition to making detecting subsets of at-risk Pima women possible, that the model could be used by epidemiologists on the ground.

Evaluating the Field Model

```
conf2$overall; conf2$byClass
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##     0.69791667     0.31622252     0.62765681     0.76194734     0.65104167
## AccuracyPValue  McnemarPValue
##     0.09807022     0.35801972
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.8000000            0.5074627            0.7518797
##       Neg Pred Value            Precision               Recall
##            0.5762712            0.7518797            0.8000000
##                   F1           Prevalence       Detection Rate
##            0.7751938            0.6510417            0.5208333
## Detection Prevalence    Balanced Accuracy
##            0.6927083            0.6537313
```

This model still has a decent sensitivity (80%), but suffers from a lower specificity (51%). Despite this low specificity, we stand by the usefulness of this model as its positive predictive power is relatively high - 75%. This means unlike the commonly asked brain teaser P(disease | test positive) of a highly reliable test, that 75% of individuals (in the test set) diagnosed with diabetes with this model actually had diabetes. [3]

Further Evaluation

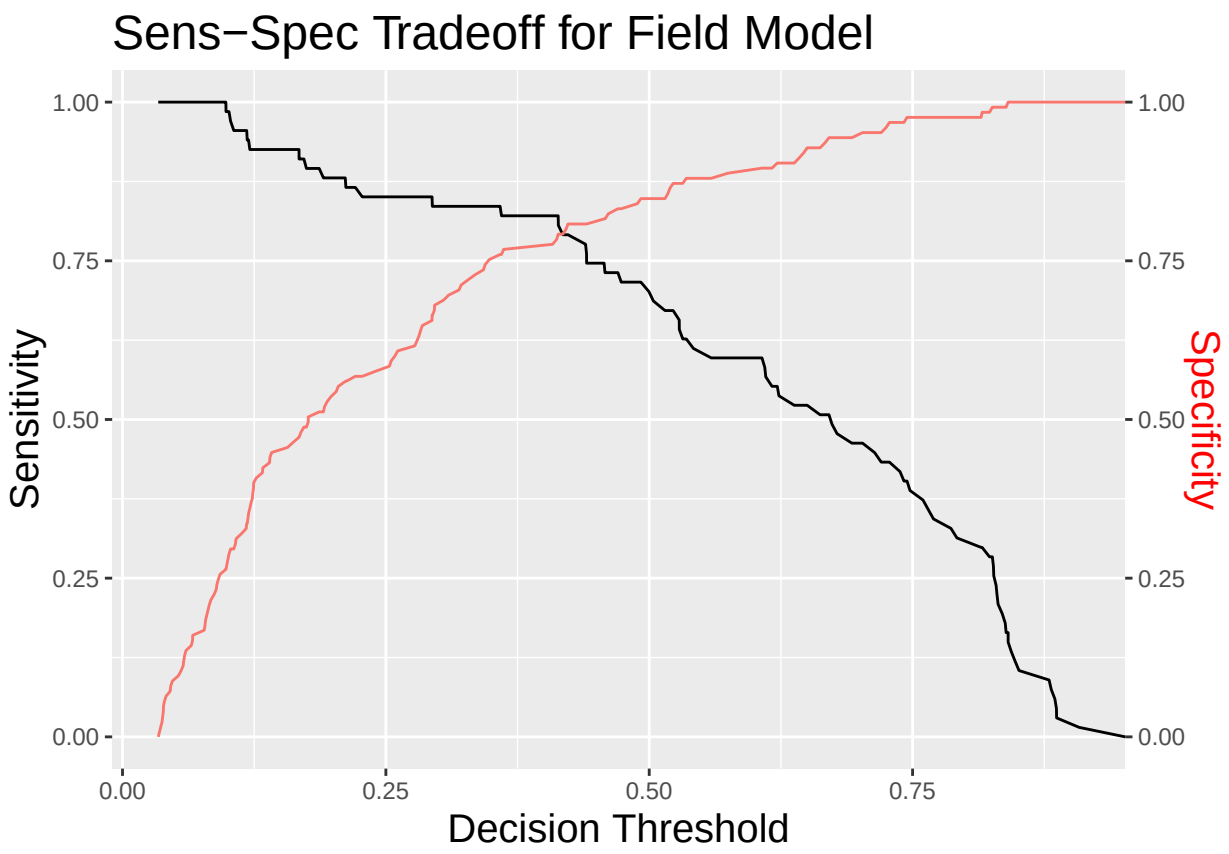
The same sensitivity-specificity tradeoff analysis was done as in the case of the Full Model.

```
predictions2 <- prediction(predictions2$Y,testDF$response2)

sens2 <- data.frame(x=unlist(performance(predictions, "sens")@x.values),
                    y=unlist(performance(predictions, "sens")@y.values))
spec2 <- data.frame(x=unlist(performance(predictions, "spec")@x.values),
                    y=unlist(performance(predictions, "spec")@y.values))

sens2 %>% ggplot(aes(x,y)) +
  geom_line() +
  geom_line(data=spec2, aes(x,y,col="red")) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Specificity")) +
  labs(x='Decision Threshold', y="Sensitivity") +
  ggtitle("Sens-Spec Tradeoff for Field Model") +
  gg2
```

[3] "A patient goes to see a doctor. The doctor performs a test with 99 percent reliability–that is, 99 percent of people who are sick test positive and 99 percent of the healthy people test negative. The doctor knows that only 1 percent of the people in the country are sick. Now the question is: if the patient tests positive, what are the chances the patient is sick?... The intuitive answer is 99 percent, but the correct answer is 50 percent."



Field Model Predictions On Fabricated Data

This model was used later to make predictions on fabricated data consisting of all Age/BMI variations with both variables ranging from 20-60 in increments of 5. The results are shown below:

```
temp <- data.frame(expand.grid(bmi=seq(20,60,5),age=seq(20,60,5)))
temp2 <- data.frame(cbind(rep(0,81),rep(0,81),rep(0,81),rep(0,81),rep(0,81),temp[,1],rep(0,81),
                          temp[,2],rep(0,81),rep(0,81)) )
names(temp2) <- names(testDF)
predictions3 <- predict(objModel2, temp2[,c(6,8)], type='prob')
fin <- cbind(temp,predictions3$Y)
fin$`predictions3$Y` <- ifelse(fin$`predictions3$Y`<0.25, "Low",
                               ifelse(fin$`predictions3$Y`<0.4, "Medium",
                               ifelse(fin$`predictions3$Y`<0.5, "High", "Very High")))
names(fin)[3] <- 'Pred'
print(xtable(spread(fin, bmi, Pred), include.rownames = T))
```

% latex table generated in R 3.4.2 by xtable 1.8-3 package % Thu Dec 13 01:02:10 2018

This table was prettified and color-coded in Microsoft Word, then included in the presentation. Both the raw xtable and the color-coded one can be found on the next page.

Shiny App

Our Full Model is hosted at the link included in the Introduction, and involved saving the Full Model into an .rds, which was then loaded in Shiny.

	age	20	25	30	35	40	45	50	55	60
1	20.00	Low	Low	Low	Medium	Medium	Medium	High	High	High
2	25.00	Low	Low	Low	Medium	Medium	High	High	High	High
3	30.00	Low	Low	Medium	High	High	Very High	Very High	Very High	Very High
4	35.00	Low	Low	High	Very High	Very High	Very High	Very High	Very High	Very High
5	40.00	Low	Low	Very High	Very High	Very High	Very High	Very High	Very High	Very High
6	45.00	Low	Low	Very High	Very High	Very High	Very High	Very High	Very High	Very High
7	50.00	Low	Low	Very High	Very High	Very High	Very High	Very High	Very High	Very High
8	55.00	Low	Low	Very High	Very High	Very High	Very High	Very High	Very High	Very High
9	60.00	Low	Low	Very High	Very High	Very High	Very High	Very High	Very High	Very High

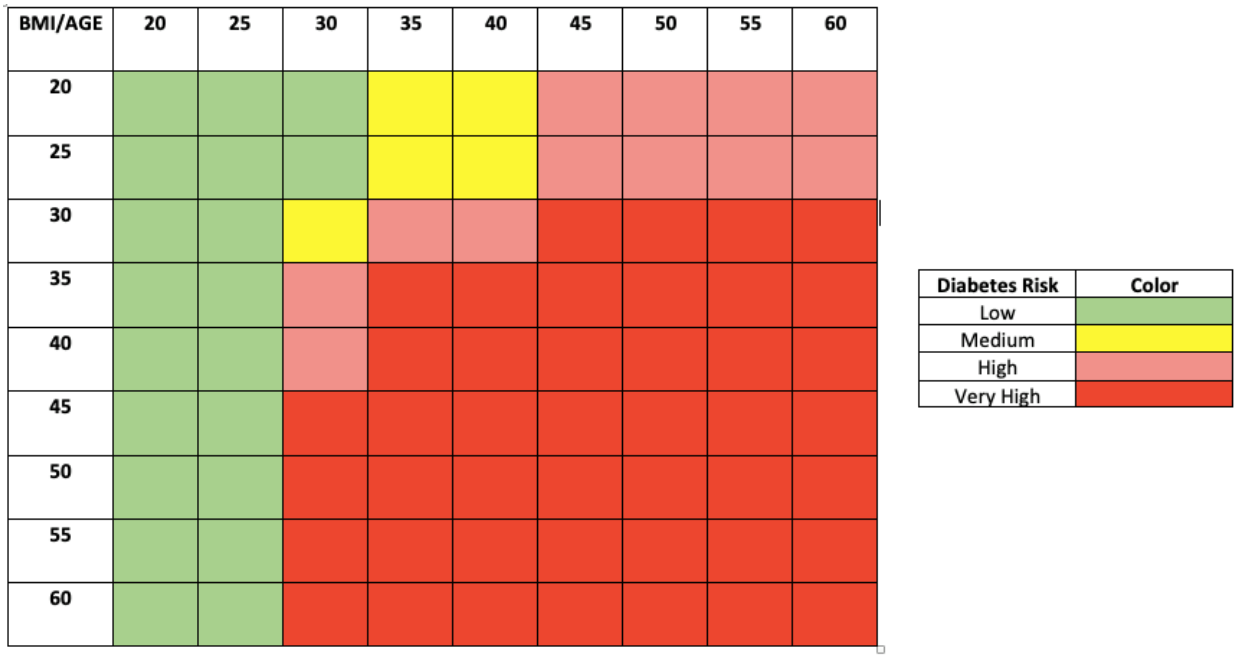


Figure 1: Table Used In the Presentation

```
saveRDS(objModel, "final_model.rds")
```

The code for the Shiny app is also included below:

```
library(caret)
library(shiny)
library(gbm)

ui <- navbarPage("Diabetes Diagnosis Tool",
  tabPanel("Prediction"),
    sidebarLayout(
      sidebarPanel(
        numericInput('PRG', 'Number of pregnancies', 1, min = 0, max = 20, step = 1,
                     width = NULL),
        numericInput('PLASMA', 'Blood plasma glucose levels', 120, min = 0, max = 200, step = 1,
                     width = NULL),
        numericInput('BP', 'Blood pressure', 80, min = 0, max = 200, step = 1,
                     width = NULL),
        numericInput('THICK', 'Triceps skinfold thickness', 20, min = 0, max = 1000, step = 1,
```

```
                    width = NULL),
        numericInput('INSULIN', 'Salivary insulin, ', 10, min = 0, max = 200, step = 1,
                    width = NULL),
        numericInput('BODY', 'BMI', 30, min = 0, max = 100, step = 0.1,
                    width = NULL),
        numericInput('PEDIGREE', 'Family history', 0.5, min = 0, max = 3, step = 0.01,
                    width = NULL),
        numericInput('AGE', 'AGE', 30, min = 0, max = 100, step = 1,
                    width = NULL)
      ),
      mainPanel(
        p("Probability of diabetes: ", style="font-size:40px"),
        htmlOutput("summary"),
        htmlOutput("summary2"),
        br(),
        p("Values greater than 0.5 reflect model's decision to predict as having diabetes.")
      )
    )
)

server <- function(input, output) {

  model <- readRDS("final_model.rds")
  pred_names <- c("PRG", "PLASMA", "BP", "THICK", "INSULIN", "BODY", "PEDIGREE", "AGE")

  get_pred <- reactive({
    df <- data.frame("PRG"=input$PRG, "PLASMA"=input$PLASMA, "BP"=input$BP, "THICK"=input$THICK,
                     "INSULIN"=input$INSULIN, "BODY"=input$BODY,"PEDIGREE"=input$PEDIGREE,
                     "AGE"=input$AGE, "RESPONSE"=0, "RESPONSE2"=0)
    res <- predict(model, df[,pred_names], type='prob')
    res$Y
    })

  output$summary <- renderText({
    val <- ifelse(get_pred()<0.25, "Low",
                          ifelse(get_pred()<0.4, "Medium",
                          ifelse(get_pred()<0.5, "High", "Very High")))
    if (get_pred()>0.5){
      paste("<span style=\"font-size:40px; color: red\">", val, "</span>")
    } else if (get_pred()>0.4) {
      paste("<span style=\"font-size:30px; color: pink\">", val, "</span>")
    } else {
      paste("<span style=\"font-size:30px\">", val, "</span>")
    }
  })

  output$summary2 <- renderText({
    paste0("<span style=\"font-size:20px\">(", round(get_pred(),2), ")</span>")
    })
}

shinyApp(ui = ui, server = server)
```