Appendix B - R Code

```
library(dplyr)
library(reshape2)
library(ggplot2)
library(faraway)
library(tidyverse)
library(bestglm)
library(caret)
library(rFSA)
library(regclass)
library(cowplot)
library(pROC)
set.seed(1)
rm(list = ls())
importedData <- read.csv(file = "diabetes.csv")
names(importedData)[names(importedData) == "DiabetesPedigreeFunction"] <- "Pedigree"
summary(importedData)

cleanData <- filter(importedData, Glucose > 0, BloodPressure > 0, SkinThickness > 0, Insulin > 0, BMI >

summary(cleanData)

# Pairs
pairs(cleanData)


# Correlation heat map
cc = cor(cleanData, method = "pearson")
cc_df <- as.data.frame(cc)
cc_df$Vars = row.names(cc_df)
ccm = melt(cc_df, id = "Vars")
ccm$Vars <- factor(ccm$Vars, levels = row.names(cc_df))
ggplot(ccm, aes(x = variable, y = Vars)) +
  geom_tile(aes(fill = value), colour = "grey45") +
  coord_equal() +
  geom_text(size = 7, aes(label = round(value,2))) +
  scale_fill_gradient(low = "navy", high = "darkorange") +
  theme(axis.text.y = element_text(size = 15, face = "bold", colour = "grey25"),
        legend.title = element_text(size = 15, face = "bold"),legend.position = "bottom",
        axis.text.x = element_text(size = 15, angle = 90, face = "bold",colour = "grey25", vjust = 0.5,
        panel.background = element_blank(), panel.border = element_rect(fill = NA, colour = NA),
        axis.ticks = element_blank()) +
  labs(x= "", y = "", fill = "Pearson's Correlation") +
  scale_x_discrete(position = "top") +
  scale_y_discrete(limits = rev(levels(ccm$Vars)))
```

```
##############################
# CV Data
##############################
names(cleanData)[names(cleanData) == "Pedigree"] <- "DiabetesPedigreeFunction"
n <- nrow(cleanData)
index <- sample(1:n, n*0.8, replace=FALSE)
trainDat <- cleanData[index, ]
testDat <- cleanData[-index, ]

# Cross fold validation
nFolds <- 5
folds <- createFolds(trainDat$Outcome, k = nFolds)


########################
# Best subset selection
########################
best.logit <- bestglm(trainDat, family = binomial("logit"), IC = "AIC", method = "exhaustive")
summary(best.logit$BestModel)
best.logit$Subsets


x <- glmFSA(Outcome ~ ., data = trainDat, interactions = TRUE, return.models = TRUE)
x$solutions
x

# Logit with 5 predictors (Pregnancies,   Glucose + BMI + PedigreeFunction + Age) has the lowest AIC of
########################
# 5 Interactions
########################
LR.prob <- predict(best.logit$BestModel, newdata = cleanData, type = "response")
plotData <- cbind(cleanData, LR.prob)
plotData$AgeGroup <- cut(plotData$Age, breaks=c(20,25,40,55,70), right = FALSE)
plotData$logAge <- log(plotData$Age)
plotData$GlucoseGroup <- cut(plotData$Glucose, breaks=c(50,80,105,185,200), right = FALSE)
plotData <- plotData[complete.cases(plotData), ]

par(mfrow = c(1,2))
fixedage <- ggplot(data = plotData) +
  aes(x = Glucose, colour = AgeGroup, group = AgeGroup, y = LR.prob) +
  geom_point() +
  stat_smooth(method = "glm", se = FALSE, method.args = list(family=binomial)) +
  ylab("Probability of GDM") +
  xlab("Glucose (mg/dL)") +
  scale_color_manual(name="Age",
                     labels=c("<25","25-39","40-54", "55-70"),
                     values=c("red","green","blue","purple")) +
  ggtitle("Glucose vs Fixed Age") +
  theme(plot.title = element_text(lineheight=2, hjust=0.5, size = 15))

fixedglucose <- ggplot(data = plotData) +
  aes(x = Age, colour = GlucoseGroup, group = GlucoseGroup, y = LR.prob) +
  geom_point() +
```

```
  stat_smooth(method = "glm", se = FALSE, method.args = list(family=binomial)) +
  ylab("Probability of GDM") +
  xlab("Age (years)") +
  ggtitle("Age vs Fixed Glucose") +
  theme(plot.title = element_text(lineheight=2, hjust=0.5, size = 15)) +
  scale_color_manual(name="Glucose",
                     labels=c("<50","50-79","80-104", "104-184"),
                     values=c("red","green","blue","purple"))

plot_grid(fixedage, fixedglucose, labels="")


########################
# ROC on test data
########################
par(mfrow = c(1,1))
LR.model <- glm(Outcome ~ Glucose + BMI + DiabetesPedigreeFunction + Age, family = binomial("logit"), t
LR.prob <- predict(LR.model, testDat, type = "response")
g <- roc(Outcome ~ LR.prob, data = testDat)
plot(g)


########################
# 5 Fold Cross Validation
########################
train.acc <- c()
test.acc <- c()
train.acc.temp <- c()
test.acc.temp <- c()
for(i in 1:nFolds) {
  LR.model <- glm(Outcome ~ Glucose + BMI + DiabetesPedigreeFunction + Age, family = binomial("logit"),
  trainPred <- predict(LR.model, trainDat[-folds[[i]], ], type = "response")
  LR.pred <- rep(0, dim(trainDat[-folds[[i]], ])[1])
  LR.pred[trainPred > .5] <- 1
  trainTable <- table(LR.pred, trainDat[-folds[[i]], ]$Outcome)


  testPred <- predict(LR.model, trainDat[folds[[i]], ], type = "response")
  LR.pred <- rep(0, dim(trainDat[folds[[i]], ])[1])
  LR.pred[testPred > .5] <- 1
  testTable <- table(LR.pred, trainDat[folds[[i]], ]$Outcome)

  train.acc.temp <- c(train.acc.temp, (trainTable[1,1]+trainTable[2,2])/sum(trainTable))
  test.acc.temp <- c(test.acc.temp, (testTable[1,1]+testTable[2,2])/sum(testTable))
}
train.acc <- rbind(train.acc, train.acc.temp)
test.acc <- rbind(test.acc, test.acc.temp)
rowMeans(train.acc)
rowMeans(test.acc)

########################
# Test accuracy
########################
```

```r
LR.model <- glm(Outcome ~ Glucose + BMI + DiabetesPedigreeFunction + Age, family = binomial("logit"), t
LR.prob <- predict(LR.model, testDat, type = "response")
LR.pred <- rep(0, dim(testDat)[1])
LR.pred[LR.prob > .5] <- 1
LR.test.table <- table(LR.pred, testDat$Outcome)
LR.test.acc <- (LR.test.table[1,1]+LR.test.table[2,2])/sum(LR.test.table)
LR.test.spec <- LR.test.table[1,1]/(LR.test.table[1,1] + LR.test.table[2,2])
LR.test.sens <- LR.test.table[1,1]/(LR.test.table[1,1] + LR.test.table[2,1])
LR.test.acc
LR.test.spec
LR.test.sens
```