Project 03

Indrajith Wasala Mudiyanselage

Question 01

a) Most of the predictors are useful in predicting the response (Outcome) except Blood Pressure. Because the distributions of Blood Pressure are quite similar for the two levels of the outcome variable.

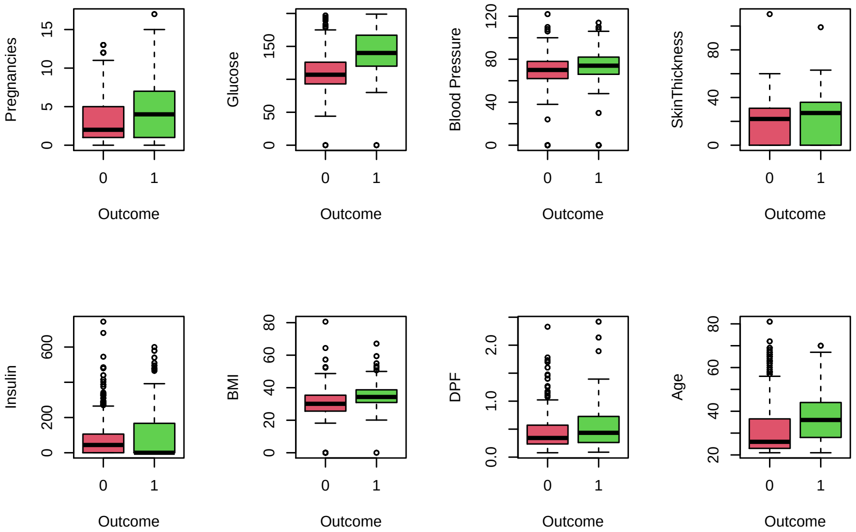

Figure 1: Exploratory analysis plots

b) In the full model, the Predictor SkinThickness is not significant (p-value=0.90> 0.05). Therefore remove only SkinThickness. The fitted reduced model is as good as the full model since the chi-square test gives the p-value=0.9024 (>0.05). Also, note that when comparing the reduced model with the null model, it gives the p-value as $2.2 \times 10^{-16} (< 0.05)$. Therefore all the selected predictors are significant.

c)

Let $p(x) = P(outcome = 1 \mid \mathbf{X} = \mathbf{x})$. Then the reduced model is

$$\begin{aligned} logit\big(p(x)\big) = &-8.027 + 0.126(Pregnancies) + 0.034(Glucose) - 0.010(BloodPressure) - 0.001(Insulin) + 0.078(BMI) \\ &+ 0.889(DPF) + 0.013(Age) \end{aligned}$$

	Estimates	Standard_Err	Lower_Limit	Upper_Limit
(Intercept)	-8.0273	0.4306	-8.8713	-7.1833
Pregnancies..	0.1264	0.0200	0.0872	0.1656
Glucose..	0.0337	0.0022	0.0294	0.0380
BloodPressure..	-0.0096	0.0032	-0.0159	-0.0033
Insulin..	-0.0012	0.0005	-0.0022	-0.0002
BMI..	0.0779	0.0085	0.0612	0.0945
DiabetesPedigreeFunction..	0.8895	0.1855	0.5259	1.2531
Age..	0.0129	0.0057	0.0017	0.0240

Table 1: Summary of estimates with 0.95 CI

Pregnancies: Coefficient 0.1264 implies that a one-unit change in Pregnancies results in a 0.1264 unit increase in the log of the odds of having diabetoes when other variables are constant (The odds of the probability of having diabeties is 1.134736 for a unit change in Pregnancies when other variables held constant).

Blood Pressure: Coefficient -0.0096 implies that a one-unit change in Blood Pressure results in a 0.0096 unit decrease in the log of the odds when other variables are constant (The odds of the probability of having diabeties is 0.9904459 for a unit change in Blood Pressure when other variables held constant).

Training error= 0.216

Question 02

a)

	Value
Error rate	0.216
Sensitivity	0.5673
Specificity	0.8967

Table 2: Measures for classification

b)

```
## b)

LOOCV<-function(dataset){

  n<-length(dataset[,1])
  lr.pred.fit<-c()

  for (i in 1:n) {

    newdata<-dataset[-i,]
    testdata<-dataset[i,]
    fit <- glm(Outcome ~ ., family = binomial, data = newdata)
    lr.prob.fit <- predict(fit, testdata, type = "response")
    lr.pred.fit[i] <- ifelse(lr.prob.fit >= 0.5, 1, 0)
  }

Test.Error<- 1 - mean(lr.pred.fit == dataset$Outcome)

return(list(Test.Error=Test.Error))
}
Error<-LOOCV(dataset=diabetes)
Error
```

```
## $Test.Error
## [1] 0.2195
```

c) Use caret package. Results in part b and c are perfectly matching to each other.

```
## c)

library(caret)

train_control <- trainControl(method="LOOCV")
fit1 <- train(Outcome ~ . ,data=diabetes, trControl = train_control, method = "glm",family=binomial(link = logit

# Test Error
Pack.Error<-as.numeric(1-fit1$results[2])
Pack.Error
```

```
## [1] 0.2195
```

d) The test error for the logistic regression that I have proposed in question 1 is 0.2185 (By using caret package).

e) The test error for the LDA that I have proposed in question 1 is 0.219 (By using caret package).Note that this is very close to part d answer.

f) The test error for the QDA that I have proposed in question 1 is 0.2375 (By using caret package)

g) K=1 gives the lowest error (0.001), but this is more flexible to the data. The third-lowest error is given by K=6. Since there is not much error difference with other two K values, select K=6 as the optimal K value.

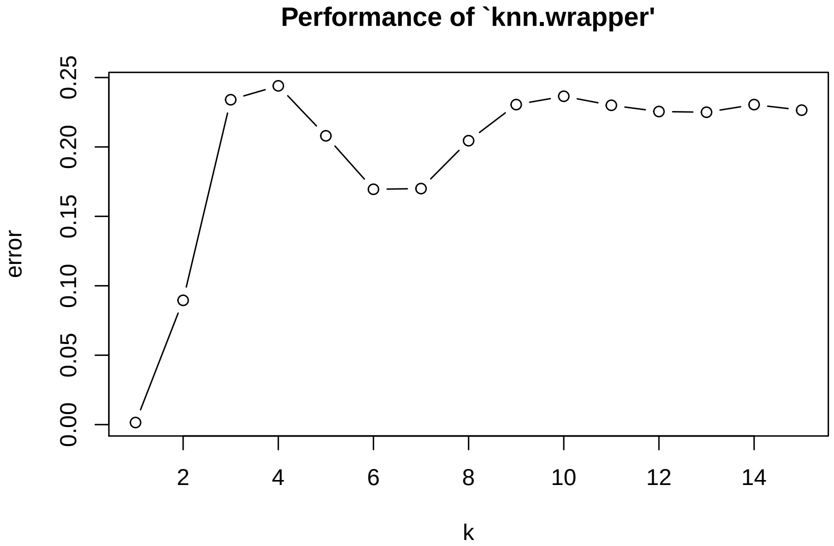

Figure 2: Test error rates for various values of K using knn

	1	2	3	4	5	6	7	8	9	10	11	12
Error rate	0.0010	0.0900	0.2325	0.2230	0.2030	0.1615	0.1740	0.2045	0.2195	0.2275	0.2210	0.2180

Table 3: Error rates for different k values

h) Since KNN has the lowest test error, I would recommend KNN as the classifier.

	Test error
Logistic	0.2195
LDA	0.2190
QDA	0.2375
KNN (K=6)	0.1615

Table 4: est Error values for different classifie

Question 03

a) The two methods fairly agree with each other as the scatterplot roughly passes through the 45-degree line. The Boxplot of absolute differences confirms that conclusion because the mean absolute difference is around 1 and also there is not much deviation of absolute differences.

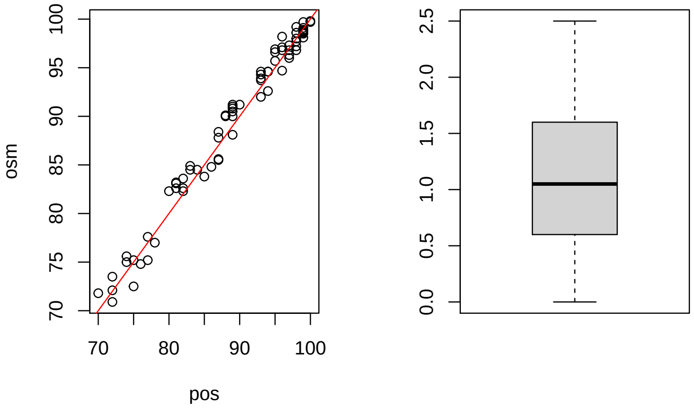

Figure 3: Scatter plot and Boxplot

b) θ is the p^{th} quantile of the absolute value of the deviation of two methods. θ is smaller means, (90%) of the total deviations are small. This means two methods give similar values and so they fairly agree with each other. Therefore smaller values for θ imply better agreement.

c) $\hat{\theta} = 2.$ is the point estimate of θ.

d) The expected value of the bootstrap distribution deviates from the true value by-0.002. The standard deviation of the distribution is 0.131. The upper bound 95% percentile bootstrap CI is 2.2. Therefore 95% of the data values fall below 2.2. Note that $\hat{\theta^*} = 1.998$.

```
## d)

theta_hat<-c()

for(i in 1:10000){
  set.seed(i)
  abs.D<- sample(abs(D),72, replace = T)
  theta<-quantile(abs.D, probs =0.9)

  theta_hat[i]<-theta
}

mean.est<-mean(theta_hat)
bias<- mean.est-true.theta[[1]]
std.err<- sd(theta_hat)
upper.CI<-quantile(theta_hat,probs = 0.95)
```

	Value
Bias	-0.001991
Standard error	0.1309216
0.95 upper confidence bound	2.2

Table 5: Bootstrap estimated values

e) Part e and part d values are very close to each other. Therefore both methods give similar values.

	Value
Bias	0.000522
Standard error	0.1301934
0.95 upper confidence bound	2.2

Table 6: Bootstrap estimated values using boot package

f) The two methods fairly agree with each other since the absolute differences of the two methods are very close to each other (around zero). Therefore I would say that the methods agree well enough to be used interchangeably in practice.

R Codes

```
knitr::opts_chunk$set(echo = TRUE}
## ----setup, include=FALSE-------------------------------------------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)


## ----include=FALSE--------------------------------------------------------------------------------------------
### Qustion 01

diabetes<-read.csv("diabetes.csv",header = T)
diabetes$Outcome<-as.factor(diabetes$Outcome)
str(diabetes)


## ----echo=FALSE,fig.align="center",fig.cap="Exploratory analysis plots",  out.width = "60%"-------------------
## a)

par(mfrow=c(2,4))
boxplot(diabetes[, 1] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Pregnancies",col=c(2,3))
boxplot(diabetes[, 2] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Glucose",col=c(2,3))
boxplot(diabetes[, 3] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Blood Pressure",col=c(2,3))
boxplot(diabetes[, 4] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "SkinThickness",col=c(2,3))
boxplot(diabetes[, 5] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Insulin",col=c(2,3))
boxplot(diabetes[, 6] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "BMI",col=c(2,3))
boxplot(diabetes[, 7] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "DPF",col=c(2,3))
boxplot(diabetes[, 8] ~ diabetes$Outcome , xlab= "Outcome" ,ylab = "Age",col=c(2,3))


## ----include=FALSE--------------------------------------------------------------------------------------------
## b)

# fit models

full <- glm(Outcome ~ ., family = binomial, data = diabetes)
summary(full)
red1<- glm(Outcome ~ Pregnancies.. +Glucose.. +BloodPressure.. +Insulin..+ BMI..+ DiabetesPedigreeFunction.. +Ag
summary(red1)
red2<- glm(Outcome ~ 1, family = binomial, data = diabetes)

## ----include=FALSE--------------------------------------------------------------------------------------------
# Check significance of the models

anova(red1, full, test = "Chisq")
anova(red2, red1, test = "Chisq")


## ----echo=FALSE, , results="asis"-----------------------------------------------------------------------------
## c)

s<-summary(red1)
Lower_Limit<-s$coefficients[,1]-1.96*s$coefficients[,2]
Upper_Limit<-s$coefficients[,1]+1.96*s$coefficients[,2]
Estimates<-s$coefficients[,1]
Standard_Err<-s$coefficients[,2]
data<-as.data.frame(cbind(Estimates,Standard_Err,Lower_Limit,Upper_Limit))
library(xtable)
options(xtable.comment=FALSE)
xtable(data,caption = "Summary of estimates with 0.95 CI",digits=c(0,4,4,4,4))


## ----include=FALSE--------------------------------------------------------------------------------------------
# Estimated probabilities for training data
```

```
lr.prob <- predict(red1, diabetes, type = "response")

# Predicted classes (using 0.5 cutoff)

lr.pred <- ifelse(lr.prob >= 0.5, 1, 0)

# training error rate

Training.Error <-1 - mean(lr.pred == diabetes[, "Outcome"])


## ----include=FALSE-------------------------------------------------------------------------------------------
###############################################################################################################

### Question 02

##a)

# Estimated probabilities for training data

lr.prob.full <- predict(full, diabetes, type = "response")

# Predicted classes (using 0.5 cutoff)

lr.pred.full <- ifelse(lr.prob.full >= 0.5, 1, 0)

# training error rate

Training.Error.full <-1 - mean(lr.pred.full == diabetes[, "Outcome"])

# Confusion matrix for training data
con.mat.full<-table(diabetes$Outcome,lr.pred.full)


## ----echo=TRUE, cache=TRUE----------------------------------------------------------------------------------
## b)

LOOCV<-function(dataset){

  n<-length(dataset[,1])

  lr.pred.fit<-c()

  for (i in 1:n) {

    newdata<-dataset[-i,]

    testdata<-dataset[i,]

    fit <- glm(Outcome ~ ., family = binomial, data = newdata)

    lr.prob.fit <- predict(fit, testdata, type = "response")

    lr.pred.fit[i] <- ifelse(lr.prob.fit >= 0.5, 1, 0)

  }

Test.Error<- 1 - mean(lr.pred.fit == dataset$Outcome)

return(list(Test.Error=Test.Error))
}
Error<-LOOCV(dataset=diabetes)
Error
```

```
## ----echo=TRUE, message=FALSE, warning=FALSE, cache=TRUE-------------------------------------------------
## c)

library(caret)

train_control <- trainControl(method="LOOCV")
fit1 <- train(Outcome ~ . ,data=diabetes, trControl = train_control, method = "glm",family=binomial(link = logit

# Test Error
Pack.Error<-as.numeric(1-fit1$results[2])
Pack.Error


## ----message=FALSE, warning=FALSE, include=FALSE, cache=TRUE---------------------------------------------
## d)

library(caret)

train_control <- trainControl(method="LOOCV")
fit2  <- train(Outcome ~ . -SkinThickness.. ,data=diabetes, trControl = train_control, method = "glm",family=bin

# Test Error
Pack.Error.red<-as.numeric(1-fit2$results[2])
Pack.Error.red


## ----include=FALSE,cache=TRUE-----------------------------------------------------------------------------
## e)

library(MASS)
library(caret)

train_control <- trainControl(method="LOOCV")
fit.lda  <- train(Outcome ~ . -SkinThickness.. ,data=diabetes, trControl = train_control, method = "lda")

# Test Error
Pack.Error.red1<-as.numeric(1-fit.lda$results[2])
Pack.Error.red1



## ----include=FALSE, cache=TRUE----------------------------------------------------------------------------
## f)

library(MASS)
library(caret)

train_control <- trainControl(method="LOOCV")
fit.qda  <- train(Outcome ~ . -SkinThickness.. ,data=diabetes, trControl = train_control, method = "qda")

# Test Error
Pack.Error.red2<-as.numeric(1-fit.qda$results[2])
Pack.Error.red2



## ----echo=FALSE, cache=TRUE,fig.align="center",fig.cap="Test error rates for various values of K using knn",
## g)

# Optimal k (tune.knn)
library(e1071)
```

```r
object <- tune.knn(diabetes[,-9], diabetes[,-c(1:8)], k = 1:15, tunecontrol = tune.control(sampling = "cross",cr
plot(object)


## ----include=FALSE,cache=TRUE--------------------------------------------------------------------------
#error rate
train.control <- trainControl(method  = "LOOCV")
set.seed(6340)
fit.knn <- train(Outcome ~ . -SkinThickness.., method = "knn", tuneGrid   = expand.grid(k = 1:20), trControl  =
            metric  = "Accuracy", data = diabetes)


## ----echo=FALSE, results='asis', cache=TRUE------------------------------------------------------------
library(xtable)
set.seed(6340)
knn.res<- c(fit.knn$results[1],1-fit.knn$results[2])
knn.res1<-as.data.frame(knn.res)[1:12,2]
knn.res2<-t(knn.res1)
row.names(knn.res2)<-c("Error rate")
knn.xtab<-xtable(knn.res2,caption = "Error rates for different k values")
digits(knn.xtab)<-c(0,rep(4,12))
err.tab1<-print(knn.xtab,table.placement = "H")


## ----include=FALSE-------------------------------------------------------------------------------------
##########################################################################################################

### Question 03

O2_satu<-read.table("oxygen_saturation.txt", header = T)
str(O2_satu)


## ---- echo=FALSE, fig.align="center",fig.cap="Scatter plot and Boxplot", out.width = "50%"---------------
## a)

par(mfrow=c(1,2))

plot(O2_satu)
abline(0,1, col="red")

abD<-abs(O2_satu$pos-O2_satu$osm)
boxplot(abD)


## ----include=FALSE-------------------------------------------------------------------------------------
## b)

D<- O2_satu$pos-O2_satu$osm
true.theta<-quantile(abs(D), probs =0.9)


## ----echo=TRUE, cache=TRUE-----------------------------------------------------------------------------
## d)

theta_hat<-c()

for(i in 1:10000){
  set.seed(i)
  abs.D<- sample(abs(D),72, replace = T)
  theta<-quantile(abs.D, probs =0.9)

  theta_hat[i]<-theta
```

```r
}

mean.est<-mean(theta_hat)
bias<- mean.est-true.theta[[1]]
std.err<- sd(theta_hat)
upper.CI<-quantile(theta_hat,probs = 0.95)



## ----echo=TRUE,warning=FALSE,message=FALSE-------------------------------------------------------------------
## e)

quantile.fn <- function(x, indices) {
    result <- quantile(x[indices],probs = 0.9)
    return(result)
}

library(boot)

set.seed(1)
quantil.boot <- boot(abs(D), quantile.fn, R = 10000)

boot.CI<-boot.ci(quantil.boot, type = "perc")
```