MSc Business Analytics: **Quantitative Methods**

Exercises - SOLUTIONS

1. Randomly split the data into training set (80% for building a predictive model) and test set (20% for evaluating the model). Make sure to set seed for reproductibility.

```
data("PimaIndiansDiabetes2", package = "mlbench")
??PimaIndiansDiabetes2

PimaIndiansDiabetes2 <- na.omit(PimaIndiansDiabetes2)

# Split the data into training and test set
set.seed(123)
train.size <- round(0.8*(dim(PimaIndiansDiabetes2)[1]))
train <- sample(1:dim(PimaIndiansDiabetes2)[1], train.size)
test <- -train
train.data <- PimaIndiansDiabetes2[train, ]
test.data <- PimaIndiansDiabetes2[test, ]
```

2. Find the optimal value of lambda that minimizes the cross-validation error.

```
# Dummy code categorical predictor variables
x <- model.matrix(diabetes~., train.data)[,-1]

# Convert the outcome (class) to a numerical variable
y <- ifelse(train.data$diabetes == "pos", 1, 0)


library(glmnet)
set.seed(123)
cv.lasso <- cv.glmnet(x, y, alpha = 1, family = "binomial")
plot(cv.lasso)
```

The plot displays the cross-validation error according to the log of lambda. The left dashed vertical line indicates that the log of the optimal value of lambda is approximately -4.5, which is the one that minimizes the prediction error. This lambda value will give the most accurate model. The exact value of lambda can be viewed as follow:

```
cv.lasso$lambda.min
```

3. Using `lambda.min` as the best lambda, obtain the penalized regression coefficients

```
coef(cv.lasso, cv.lasso$lambda.min)
```

From the output above, `pressure` and `insuline` have coefficients exactly equal to zero.

4. Compute the final lasso model, make prediction on test data and calculate the model accuracy.

```
# Final model with lambda.min
lasso.model <- glmnet(x, y, alpha = 1, family = "binomial",
                      lambda = cv.lasso$lambda.min)

# Make prediction on test data
x.test <- model.matrix(diabetes ~., test.data)[,-1]
probabilities <- predict(lasso.model, newx = x.test, s ="lambda.min", type = "res
predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")

# Model accuracy
observed.classes <- test.data$diabetes
mean(predicted.classes == observed.classes)
```

5. Fit an unpenalized logistic model, make predictions, calculate the model accuracy and compare it with the accuracy of the penlized logistic regression.

```
# Fit unpenalized logistic model
logistic.model <- glm(diabetes ~., data = train.data, family = binomial)

# Make prediction on test data
probabilities <- predict(logistic.model, test.data, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")

# Model accuracy
observed.classes <- test.data$diabetes
mean(predicted.classes == observed.classes)
```

The model accuracy of the two approaches is the same.