```
"
  Title:      Using the Support Vector Machine functionality in the e1071 library to learn from
              the Mammographic Mass Data Set and make predictions using the learned model.

  Purpose:    Programming Assignment 1 for CECS 551 taught by Dr. Todd Ebert

  Author:     Varderes Barsegyan (CSULB ID: 016163470)

  Background: Although we have not tackeld SVM's in great detail, this exercise gave us the
              opportunity to become familiarized with the SVM functionality in the e1071 library.
              The library includes various machine learning algorithms, but for this exericise
              I used the svm and tune.svm function.  The tune.svm function was used to fine tune
              the 'cost' and 'gamma' parameters by seeing how the svm performed with a set of
              these parameters.
              I used the Mammographic Masses Data Set provided by UC Irvine by loading it as a
              dataframe in my R script.  I gave the user the option to rid the dataset of rows
              containing '?' or replacing '?' with -1. The results were not identical mainly
              because more data is available when using the -1 replacement.
              Finally, I ran svm using three different kernels: linear, 2nd order polynomial,
              and (for fun) a 3rd order polynomial. Cross validation was not performed and tests
              were done using the entire dataset. Finally, the accuracy was determined by inputting
              the dataset into the learned model and examining how well it performed by calculating
              the percentage of correct predictions.
              It is interesting to examine why the 2nd order polynomial kernel performs worse
              than a linear kernel.  A possible explanation is that a linear model works better
              with this particular dataset. Furthermore there is no correlation between accuracy
              and model complexity.  It is possible that the accuracy flucuates as the complexity
              of the model increases.  In a future class, we will learn how to use different
              models alongside cross validation to find the best model for the dataset of interest.

  Results:    A random run when removing rows with NA gave the following results:
                  Accuracy using linear classifier:  0.8277108
                  Accuracy using polynomial classifier with degree 2:  0.6566265
                  Accuracy using polynomial classifier with degree 3:  0.8192771

              A random run when replacing NA with -1:
                  Accuracy using linear classifier:  0.8324662
                  Accuracy using polynomial classifier with degree 2:  0.7585848
                  Accuracy using polynomial classifier with degree 3:  0.8345473


  Conclusion: It's interesting that the linear kernel performs better than the polynomial
              classifier. One would expect the contrary.  Yet, after doing some reading, I learned
              that a polynomial/nonlinear classifer is more likely to overfit, thus performing
              poorly on the test data.  This is termed 'generalization error'. A way this problem
              can be solved is by using more data. This leads me to the next point: it is possible
              that the data set is too small. There are 5 features for about 800 data points resulting
              in poor performance by a high-order kernel.

"

# SET WORKING DIRECTORY AND IMPORT NECESSARY LIBRARIES
setwd('C:/Users/barse/Google Drive/CSULB Spring 2017/CECS551/ProgrammingAssignments/Assignment1')
library(e1071)

# READ MAMMOGRAPHIC MASSES DATASET INTO A DATAFRAME, REPLACE '?' WITH NA
mammo_df <- read.csv(file='mammographic_masses.csv', header=FALSE, sep=',', na.strings="?")

# SET COLUMNS NAME OF DATAFRAME
column_names <- c('Birads','Age','Shape','Margin','Density','Severity')
colnames(mammo_df) <- column_names

# CHECK WHETHER TO REPLACE '?' WITH 'NA' OR '-1' (EXERCISE 1 AND 2 DISTINCTIONS)
exercise <- readline("Which exercise do you want to run? \n\n1: Rows with NA removed\n2: '?' replaced with -1\n")

if (exercise == "1") {
  # REMOVE ROWS WITH 'NA'
  mammo_df <- mammo_df[complete.cases(mammo_df),]
} else if (exercise == "2") {
  # REPLACE 'NA' WITH -1
  mammo_df[is.na(mammo_df)] <- -1
}

# LEARN MODEL USING LINEAR KERNEL
mammo_svm <- svm(Severity~., data=mammo_df, kernel='linear', type='C-classification')
print((mammo_svm))

# TEST MODEL AND SHOW SUMMARY
predictions <- predict(mammo_svm, mammo_df[,-6])
predictions_table <- table(predictions=predictions, truth=mammo_df[,6])

mean = mean(predictions==mammo_df[,6])[1]


# LEARN MODEL USING DEGREE 2 POLYNOMIAL
```

```
mammo_svm_poly <- svm(Severity~., data=mammo_df, kernel='polynomial',degree=2, type='C-classification')
print(summary(mammo_svm_poly))

predictions_poly <- predict(mammo_svm_poly, mammo_df[,-6])
predictions_table_poly <- table(predictions=predictions_poly, truth=mammo_df[,6])

mean_poly <- mean(predictions_poly==mammo_df[,6])


# JUST FOR FUN, LEARN MODEL USING DEGREE 3 POLYNOMIAL
mammo_svm_poly3 <- svm(Severity~., data=mammo_df, kernel='polynomial',degree=3, type='C-classification')
print(summary(mammo_svm_poly3))

predictions_poly3 <- predict(mammo_svm_poly3, mammo_df[,-6])
predictions_table_poly3 <- table(predictions=predictions_poly3, truth=mammo_df[,6])

mean_poly3 <- mean(predictions_poly3==mammo_df[,6])


# PRINT FINAL RESULTS
cat("Accuracy using linear classifier: ", mean, '\n')
cat("Accuracy using polynomial classifier with degree 2: ", mean_poly, '\n')
cat("Accuracy using polynomial classifier with degree 3: ", mean_poly3)
```