Chapter 9 - Applied Exercise 4

Sean Powell, Koby Arndt, Casey Clark, Nick Ohlheiser

4) *Generate a simulated two-class data set with 100 observations and two features in which there is a visible but non-linear separation between the two classes. Show that in this setting, a support vector machine with a polynomial kernel (with degree greater than 1) or a radial kernel will outperform a support vector classifier on the training data. Which technique performs best on test data? Make plots and report training and test error rates in order to back up your assertions.*

First we set up the R environment.

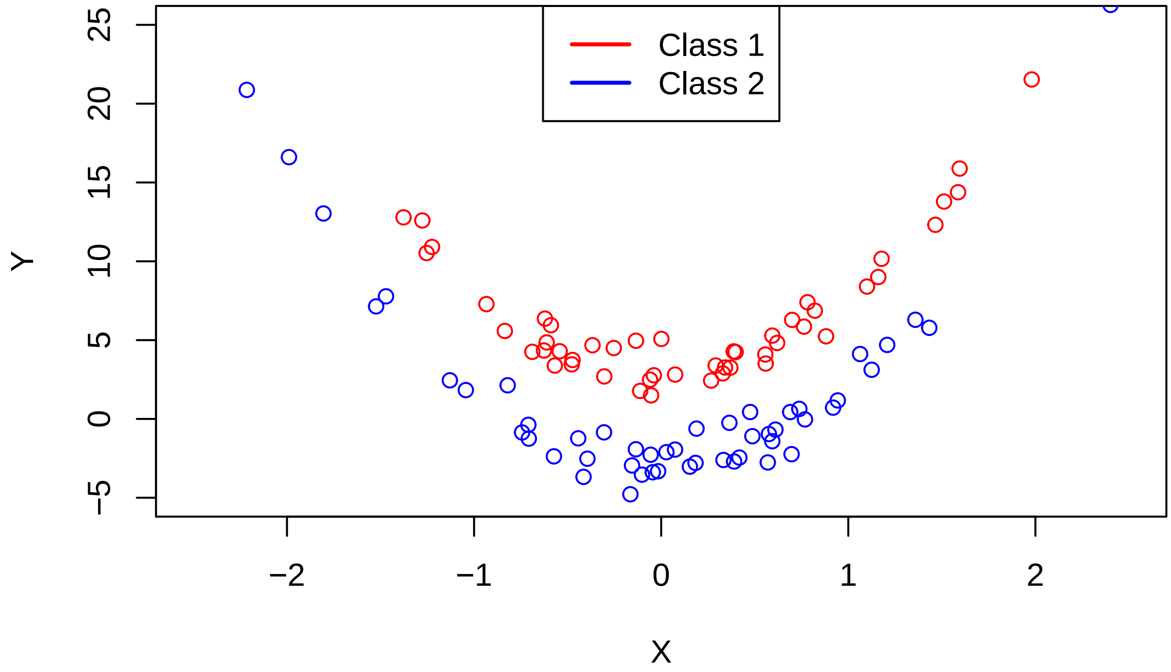
```r
#Remove all pre-existing variables
rm(list=ls())
#Load the e1071 library
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.3
```

```r
#Set seed to ensure same random numbers are generated every time
set.seed(1)
```

Create data set with 100 observations with non-linear separation between the 2 classes.

```r
#Create 100 random X values
x <- rnorm(100)
#Create equation to produce 100 Y values from X
#Y is parabolic equation (x^2), therefore will have non-linear separation between classes
y <- 5*x^2+rnorm(100)
#Randomly select 50 values between 1 and 100
#The 50 numbers will signal which Y values to be placed in Class 1
class1 <- sample(100, 50)
#Time to separate the data set into 2 separate groups
#For every Y value in Class 1, add 3
y[class1] <- y[class1]+3
#For every Y value not in Class 1, subtract 3
y[-class1] <- y[-class1]-3
#Plot the X and Y values to show the separation between the 2 classes
plot(x[class1], y[class1], col="red", xlab="X", ylab="Y", xlim=c(-2.5, 2.5), ylim=c(-5, 25))
points(x[-class1], y[-class1], col="blue")
legend("top", legend=c("Class 1", "Class 2"), col=c("red", "blue"), lty=1, lwd=2, cex=1)
```

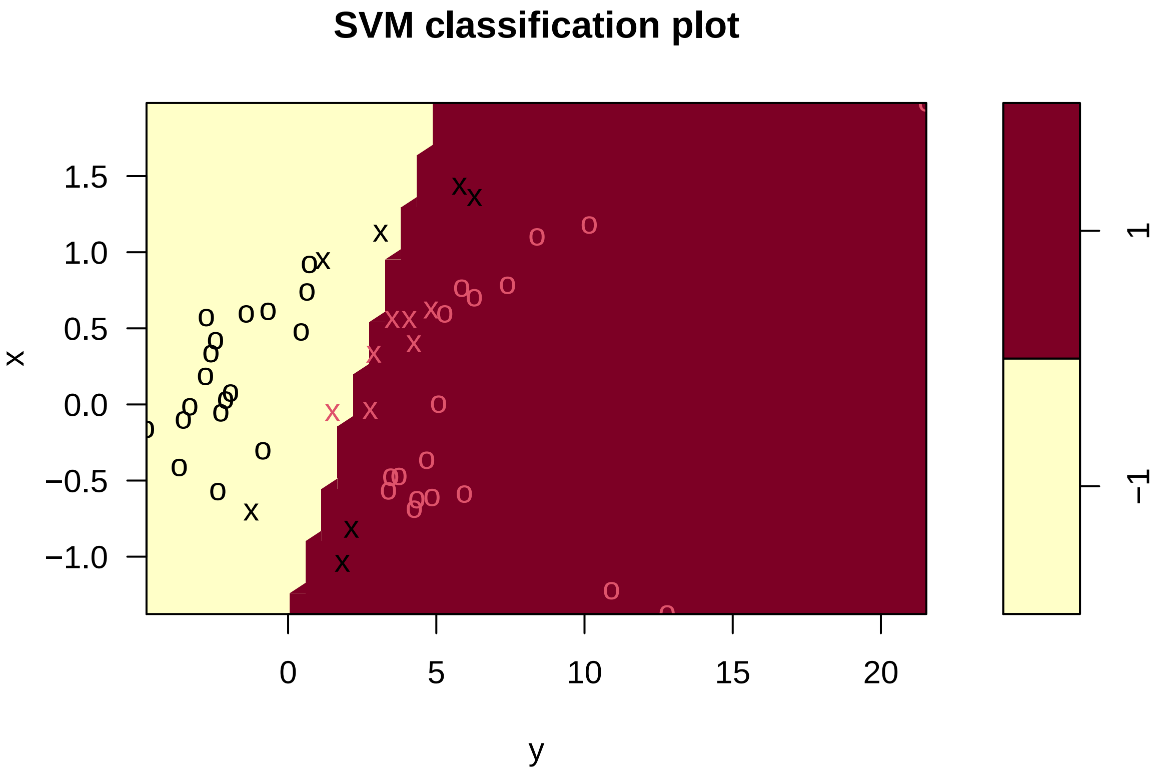


Create the training and testing data sets.

```
#Create Z component for every X and Y pair (preset all Z's equal to -1)
z <- rep(-1, 100)
#Change the Z values in Class 1 to equal 1
z[class1] <- 1
#Create data frame with X, Y, and Z values to split into test/train sets
#Z variables to be factors (this will be what we try to predict)
data <- data.frame(x = x, y = y, z = as.factor(z))
#Produce random 50 values between 1 to 100
#Will be used to select which rows go into the test & train data sets
dataSet <- sample(100, 50)
#Save selected rows into the training data set
train <- data[dataSet, ]
#Save the unselected rows into the testing data set
test <- data[-dataSet, ]
```

Use the training data to fit a linear Support Vector Classifier.

```
linear <- svm(z~., data=train, kernel="linear", cost=10)
plot(linear, train)
```



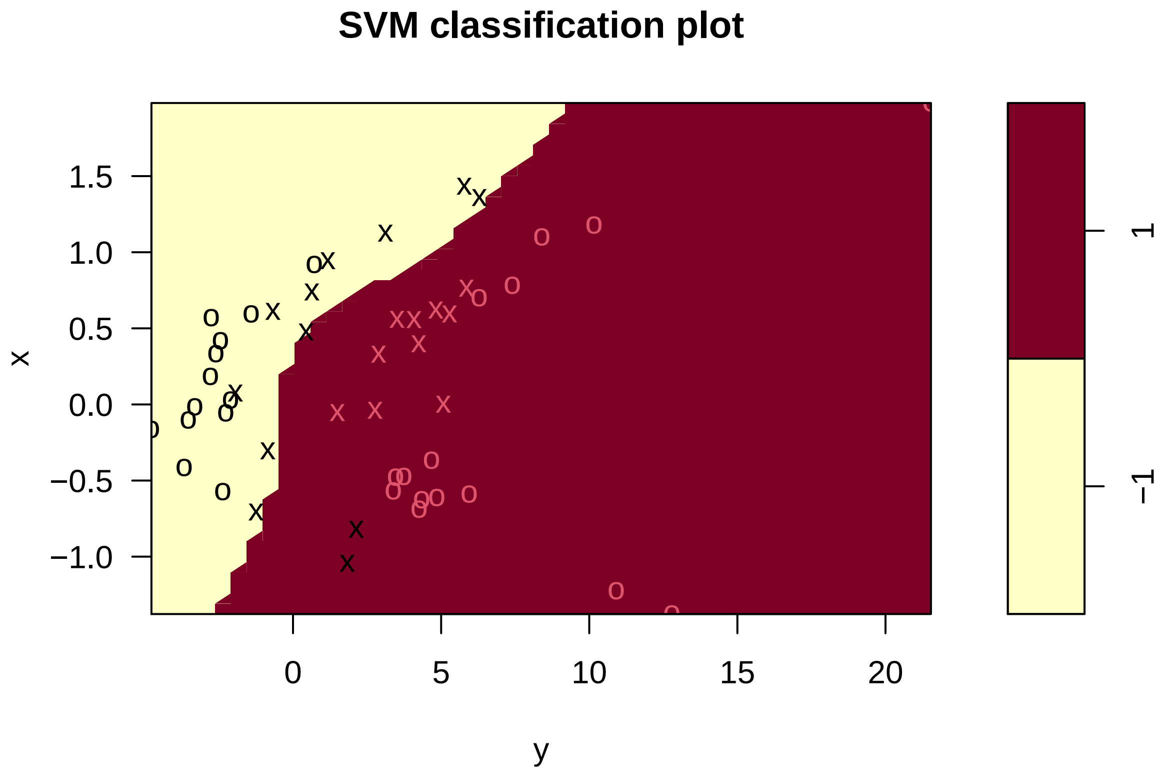
```r
table(linear$fitted, train$z)
```

```
##
##      -1  1
##   -1 21  1
##   1   4 24
```

The support vector classifier made 5 errors on the training data.

Use the training data to fit a support machine with a polynomial kernel.

```r
#Default number of degrees is 3
poly <- svm(z~., data=train, kernel="polynomial", cost=10)
plot(poly, train)
```



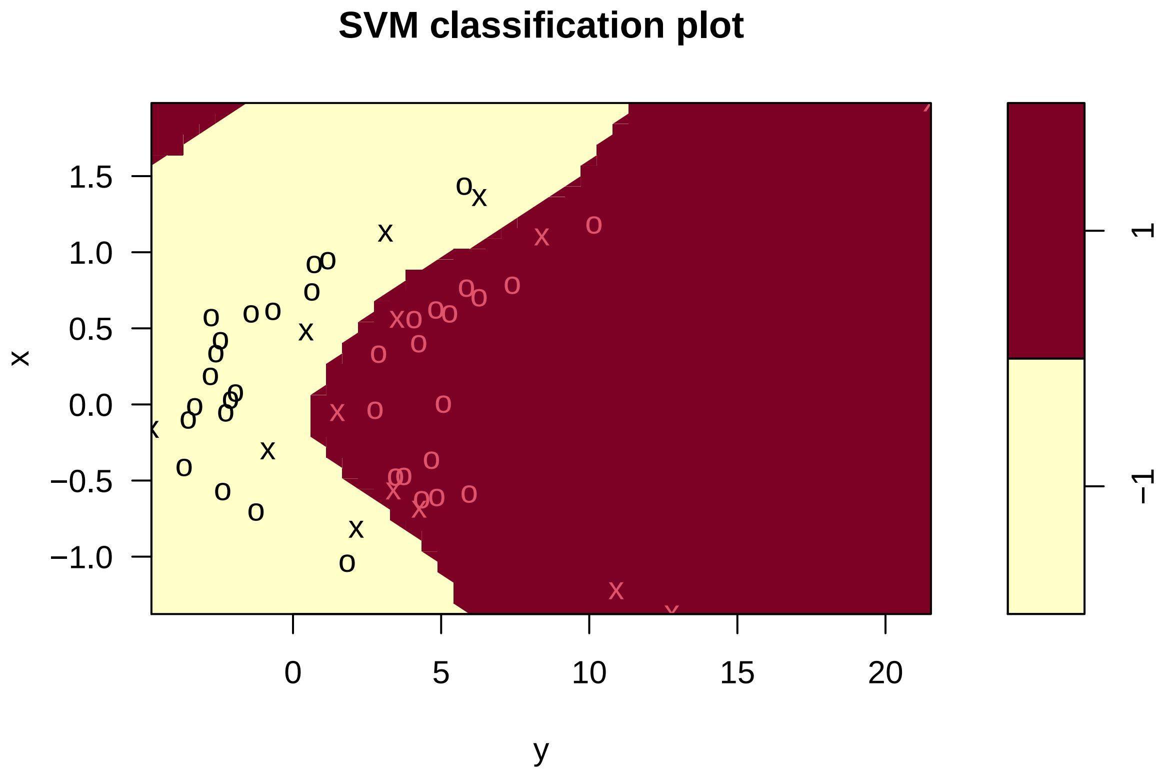
```r
table(poly$fitted, train$z)
```

```
##
##       -1  1
##   -1 22  0
##   1   3 25
```

The support vector machine with a polynomial kernel made 3 errors on the training data.

Use the training data to fit a support vector machine with a radial kernel.

```r
radial <- svm(z~., data=train, kernel="radial", gamma=1, cost=10)
plot(radial, train)
```



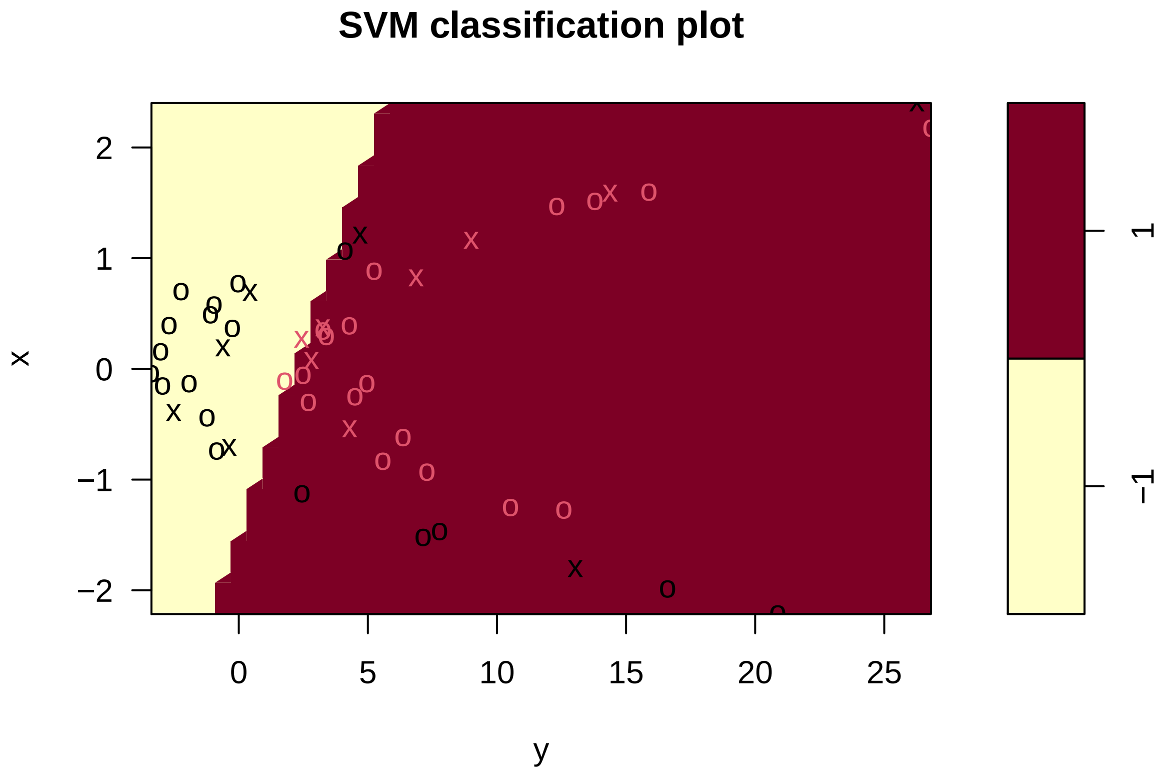
```r
table(radial$fitted, train$z)
```

```
##
##       -1  1
##   -1 25  0
##   1   0 25
```

The support vector machine with a radial kernel made 0 error on the training data.

Use the test data to see how accurate the models.

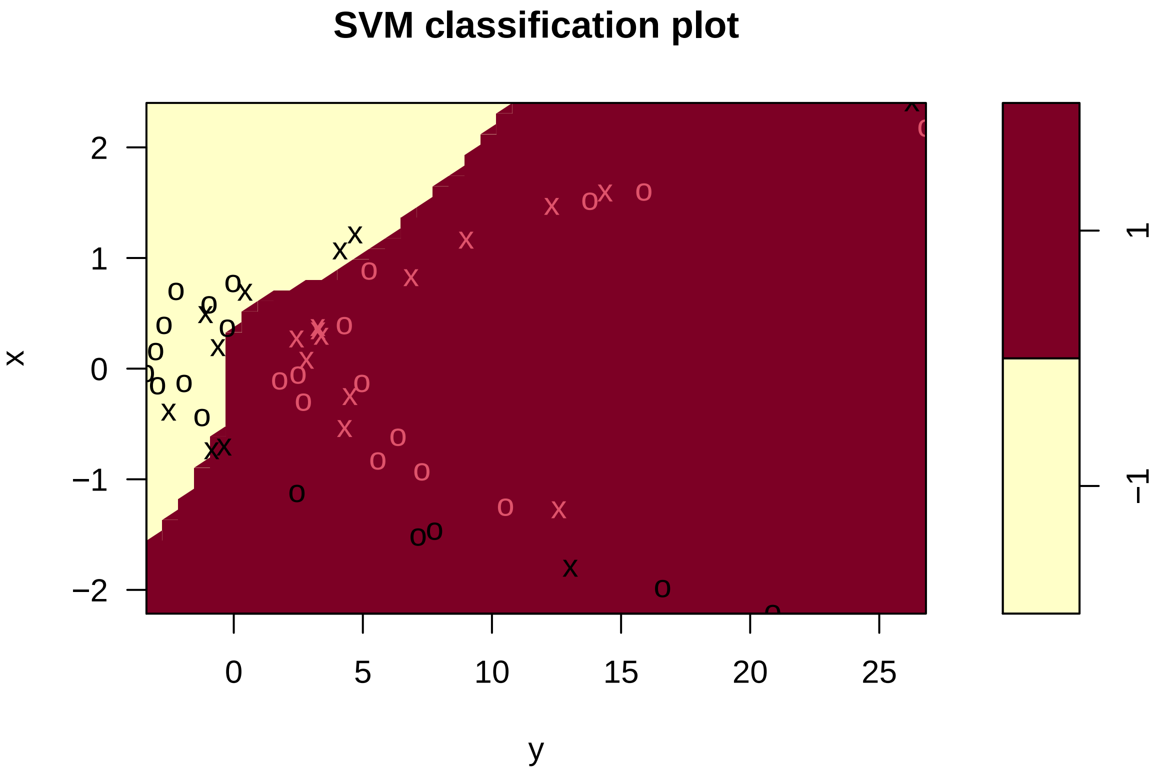
```r
#Support Vector Classifier
plot(linear, test)
```



```
table(predict(linear, test), test$z)
```

```
##
##       -1  1
##   -1 16  0
##   1   9 25
```

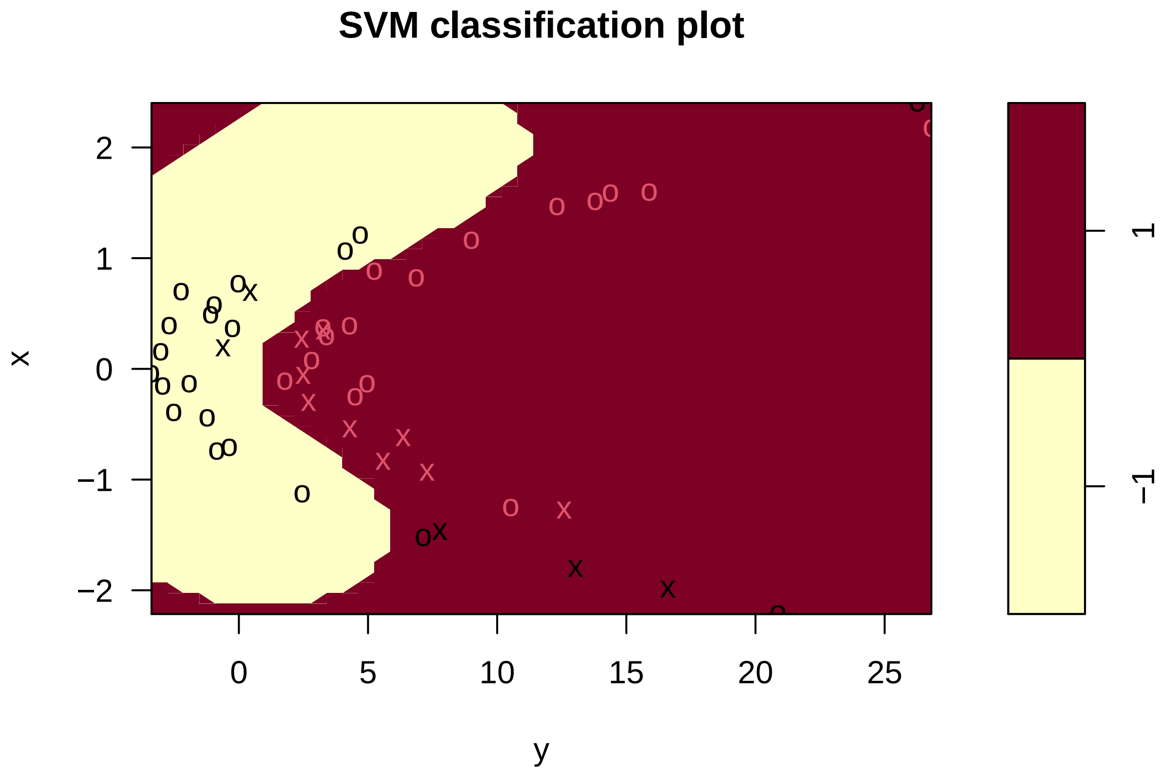
```
#Support Vector Machine with a Polynomial Kernel
plot(poly, test)
```



```
table(predict(poly, test), test$z)
```

```
##
##       -1  1
##   -1 15  0
##   1  10 25
```

```
#Support Vector Machine with a Radial Kernel
plot(radial, test)
```



```
table(predict(radial, test), test$z)
```

```
##
##       -1  1
##   -1 19  0
##   1   6 25
```

From the tables using the test data set, we can see that the linear model incorrectly classified 9 observations, the polynomial was wrong on 10 occasions, and the radial model classified 6 observations incorrectly.

The question asked us to show the polynomial model or the radial model would outperofrm the linear model. By these numbers, the radial model is the best option as it had the least amount of incorrect classifications.