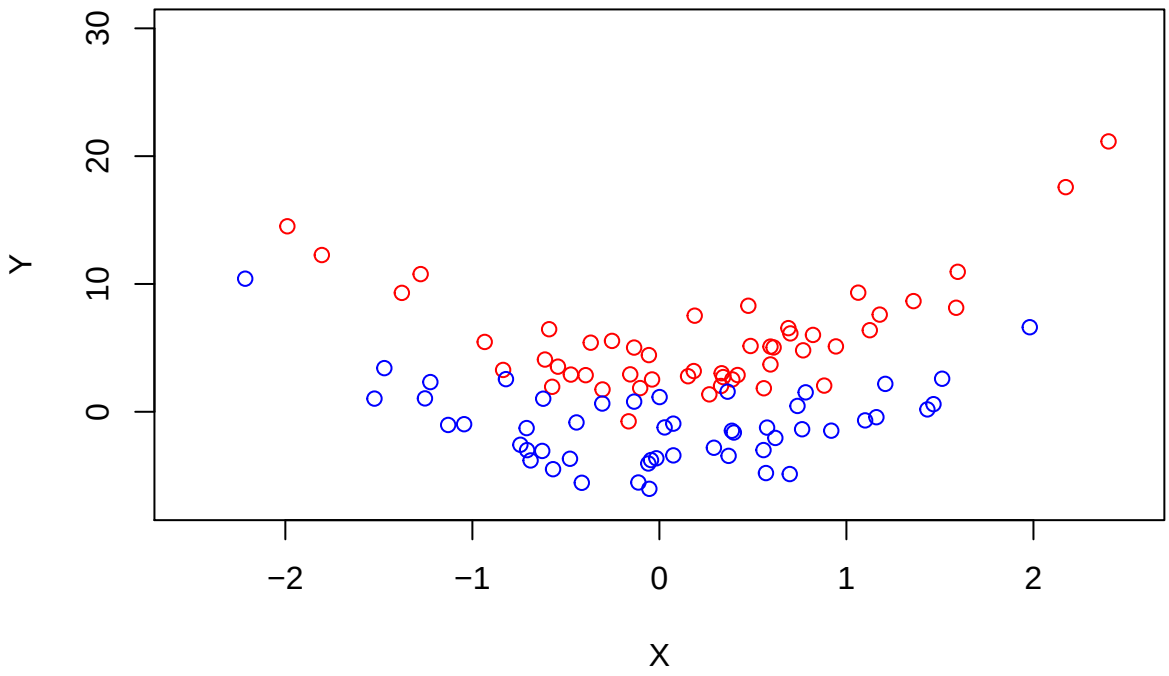
Chapter 9 Problem 4

Andira Putri

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 or a radial kernel will outperform a support vector classifier on the training data. Which technique performs best on the test data? Make plots and report training/test error rates to back up your assertions.

I generate a two-class data set by using $Y = 3x^2$ as a base for the points to hang around, and then shifting the curve up and down to create the separation.

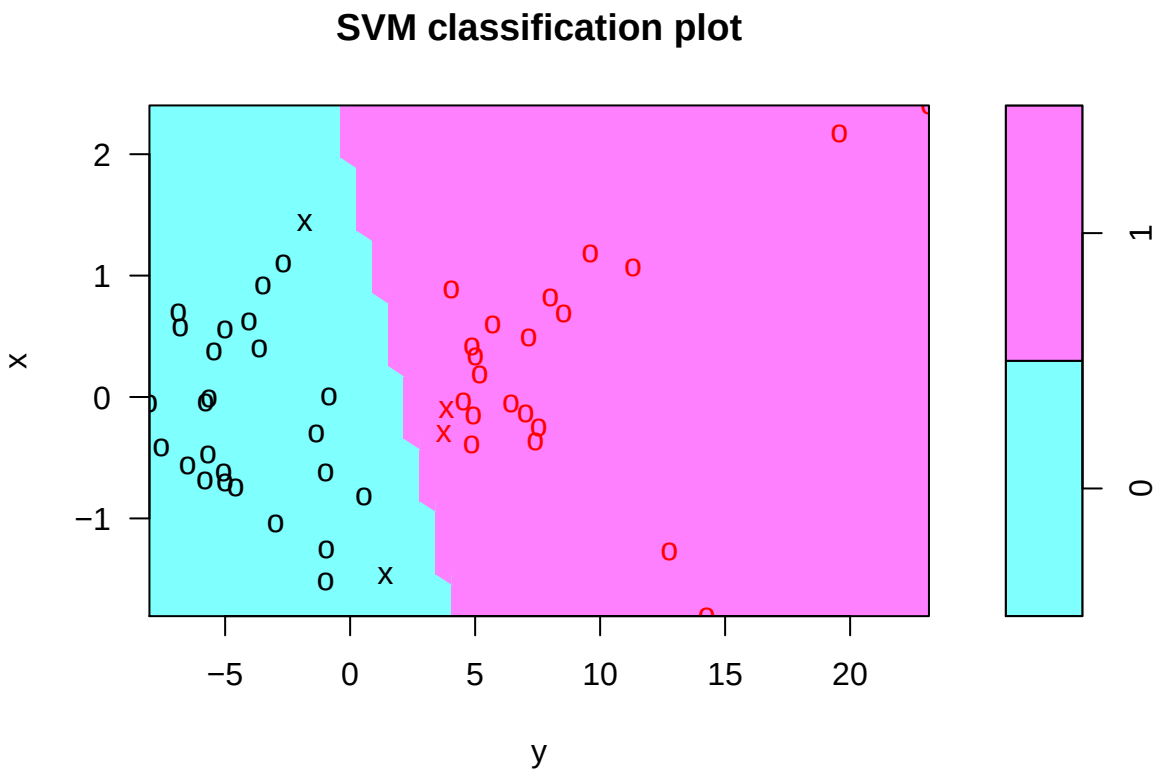
```
set.seed(1)
#create initial set of points
x=rnorm(100)
eps=rnorm(100,0,2)
y=(3*x^2)+eps
#create separation, 50 points per separation
#separation creates the two different classes
class=sample(100,50)
#shift up
y1=y[class]+3
#shift down
y0=y[-class]-3
#class separation
Y0=y
Y0[class]=y[class]+5
Y0[-class]=y[-class]-5
#plot the two set of points
plot(x[class],y1,col="red",xlab="X",ylab="Y",ylim=c(-7,30),xlim=c(-2.5,2.5))
points(x[-class],y0,col="blue")
```



Now, I fit a support vector classifier. Unlike maximal margin classifier, a support vector classifier allows observations to be on the wrong side of the hyperplane. The decision boundary in this classifier is linear.

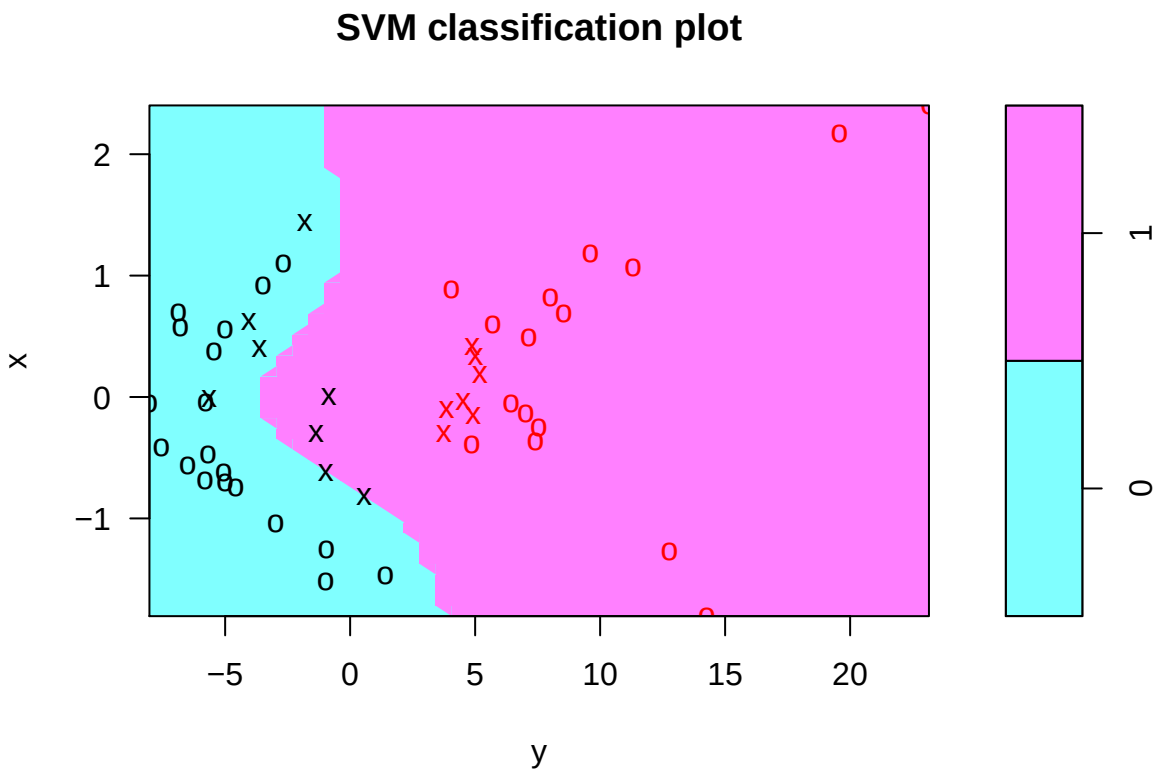
set.seed(1)

```
library(e1071)
set.seed(1)
#initialize vector with n=100 # of 0s
z = rep(0,100)
#denote top parabola as class 1 observations
z[class]=1
#combine to single data frame
data=data.frame(x=x,y=Y0,z=as.factor(z))
train=sample(100,50)
data.train=data[train,]
data.test=data[-train,]
svm.linear=svm(z~.,data=data.train,kernel="linear",cost=5)
plot(svm.linear,data.train)
```



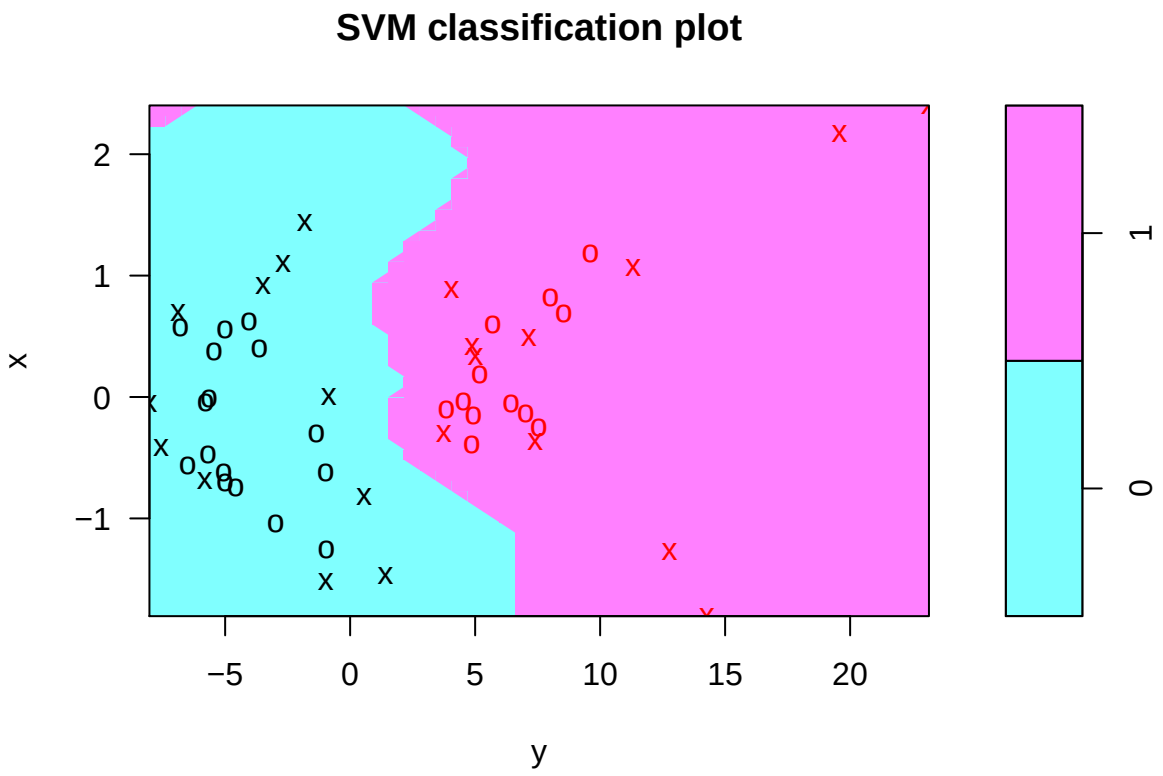
A support vector machine is essentially a support vector classifier with a non-linear boundary, dictated by a kernel. Here is a support vector machine with a polynomial kernel and decision boundary.

```
set.seed(1)
svm.quad=svm(z~.,data=data.train,kernel="polynomial",cost=5)
plot(svm.quad,data.train)
```



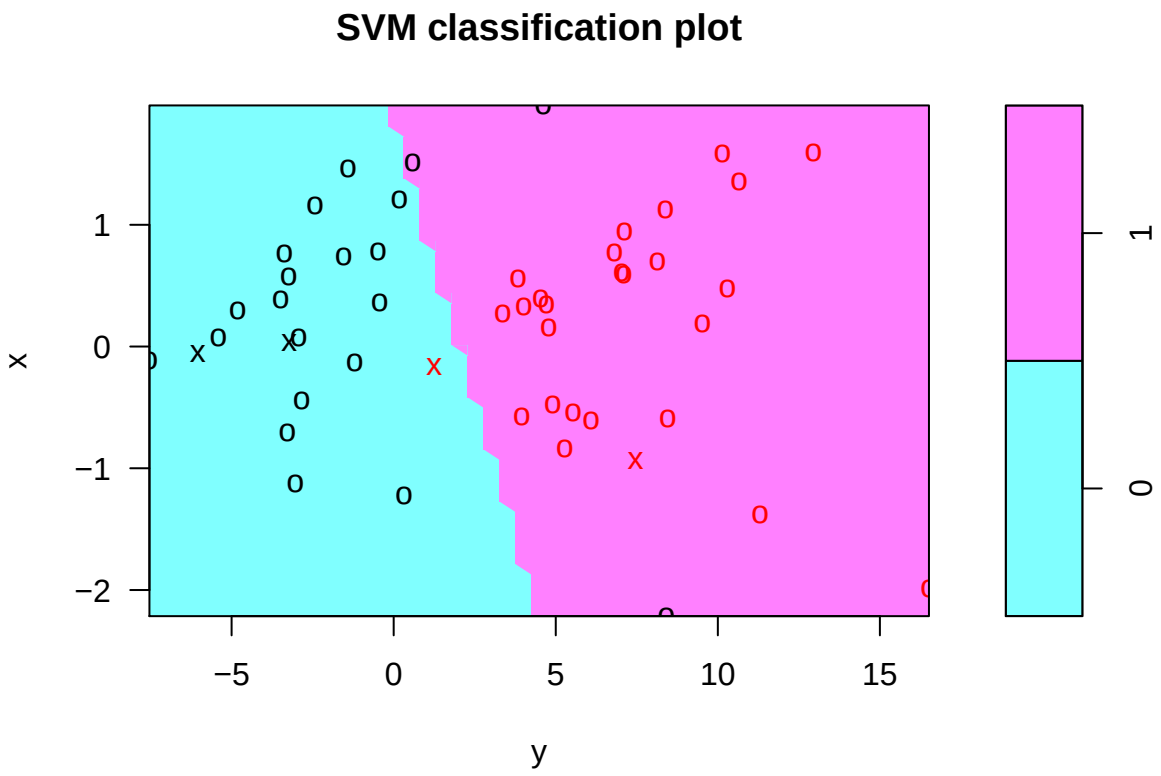
Finally, we use a radial kernel.

```
set.seed(1)
svm.rad=svm(z~.,data.train,kernel="radial",cost=5,gamma=2)
plot(svm.rad,data.train)
```



These are the resulting predictions and confusion matrices from each model.

```
set.seed(1)
#Support vector classifier
lin.pred=predict(svm.linear,data.test)
plot(svm.linear,data.test)
```

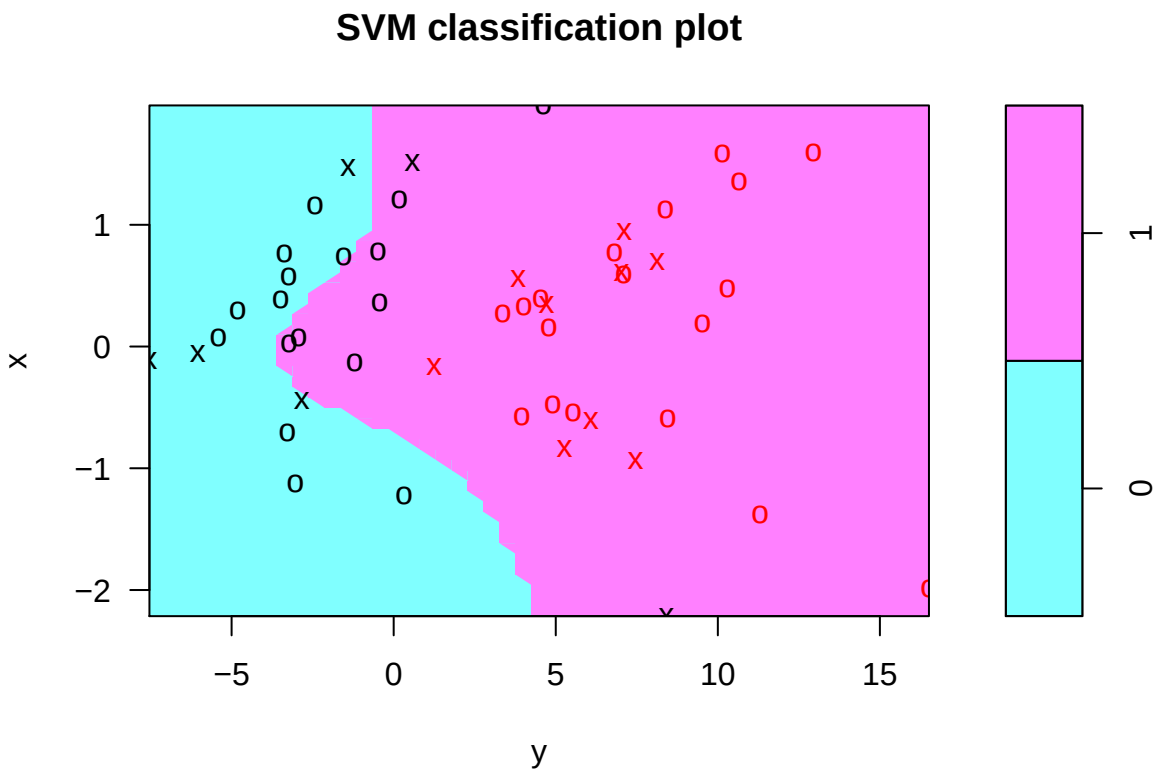



```
table(lin.pred,data.test$z)
```

```
##
## lin.pred  0  1
##        0 20  1
##        1  3 26
```

The linear decision boundary makes 4 misclassifications.

```
set.seed(1)
#SVM with polynomial kernel
quad.pred=predict(svm.quad,data.test)
plot(svm.quad,data.test)
```

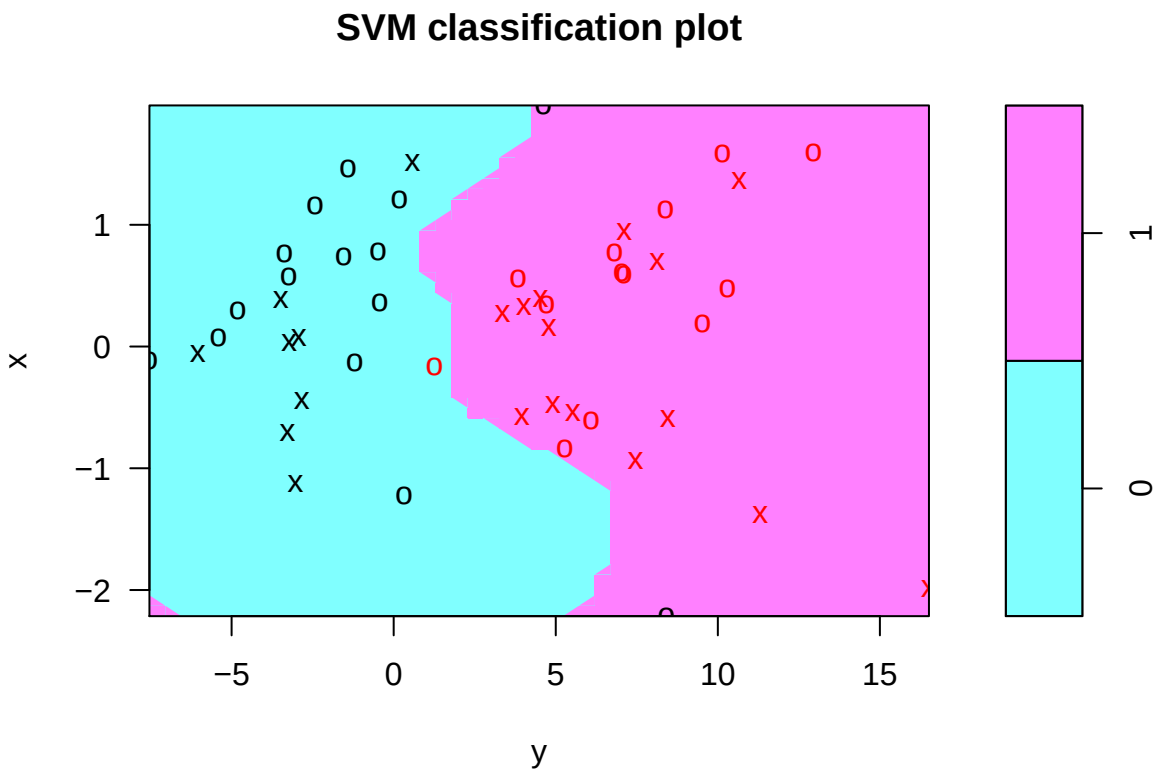



```
table(quad.pred,data.test$z)
```

```
##
## quad.pred  0  1
##         0 13  0
##         1 10 27
```

Our polynomial kernel has 10 misclassifications.

```
set.seed(1)
#SVM with radial kernel
rad.pred=predict(svm.rad,data.test)
plot(svm.rad,data.test)
```




```
table(rad.pred,data.test$z)
```

```
##
## rad.pred  0  1
##        0 21  1
##        1  2 26
```

Our radial SVM has only 3 misclassifications, making it the best-performing model for our simulated data set.