library("readxl")  
iris <- read.csv("BigDataAnalytics/iris.csv")  
head(iris)  
# install.packages("e1071")  
library("e1071")  
iris.class <- iris$class   
iris.data <- iris[,1:4]  
head(iris.data)

# retrieve the model  
svm\_linear <- svm(iris.data, iris.class, kernel="linear")  
svm\_polynomial <- svm(iris.data, iris.class, kernel="polynomial")  
svm\_radial <- svm(iris.data, iris.class, kernel="radial")  
svm\_sigmoid <- svm(iris.data, iris.class, kernel="sigmoid")

# make prediction  
pred\_linear <- predict(svm\_linear,iris.data)  
pred\_polynomial <- predict(svm\_polynomial,iris.data)  
pred\_radial <- predict(svm\_radial,iris.data)  
pred\_sigmoid <- predict(svm\_sigmoid,iris.data)

# confusion matrix  
table(pred\_linear, iris.class)  
table(pred\_polynomial, iris.class)  
table(pred\_radial, iris.class)  
table(pred\_sigmoid, iris.class)

# accuracy  
sum(pred\_linear == iris.class)/length(iris.class)  
sum(pred\_polynomial == iris.class)/length(iris.class)  
sum(pred\_radial == iris.class)/length(iris.class)  
sum(pred\_sigmoid == iris.class)/length(iris.class)

# now dividing the data  
# compose training set  
index <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))  
iris.train\_data <- iris[index==1, 1:4]  
iris.test\_data <- iris[index==2, 1:4]  
iris.train\_class <- iris[index==1, 5]  
iris.test\_class <- iris[index==2, 5]

# retrieve the model  
svm\_linear <- svm(x=iris.train\_data, y=iris.train\_class, kernel="linear")  
svm\_polynomial <- svm(x=iris.train\_data, y=iris.train\_class, kernel="polynomial")  
svm\_radial <- svm(x=iris.train\_data, y=iris.train\_class, kernel="radial")  
svm\_sigmoid <- svm(x=iris.train\_data, y=iris.train\_class, kernel="sigmoid")

# make prediction  
pred\_linear <- predict(svm\_linear,iris.test\_data)  
pred\_polynomial <- predict(svm\_polynomial,iris.test\_data)  
pred\_radial <- predict(svm\_radial,iris.test\_data)  
pred\_sigmoid <- predict(svm\_sigmoid,iris.test\_data)

# confusion matrix  
table(pred\_linear, iris.test\_class)  
table(pred\_polynomial, iris.test\_class)  
table(pred\_radial, iris.test\_class)  
table(pred\_sigmoid, iris.test\_class)

# accuracy  
sum(pred\_linear == iris.test\_class)/length(iris.test\_class)  
sum(pred\_polynomial == iris.test\_class)/length(iris.test\_class)  
sum(pred\_radial == iris.test\_class)/length(iris.test\_class)  
sum(pred\_sigmoid == iris.test\_class)/length(iris.test\_class)

Output:

# install.packages("e1071")

> library("readxl")

> iris <- read.csv("iris.csv")

> head(iris)

sepallength sepalwidth petallength petalwidth class

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

> library("e1071")

> iris.class <- iris$class

> iris.data <- iris[,1:4]

> head(iris.data)

sepallength sepalwidth petallength petalwidth

1 5.1 3.5 1.4 0.2

2 4.9 3.0 1.4 0.2

3 4.7 3.2 1.3 0.2

4 4.6 3.1 1.5 0.2

5 5.0 3.6 1.4 0.2

6 5.4 3.9 1.7 0.4

>

> # retrieve the model

> svm\_linear <- svm(iris.data, iris.class, kernel="linear")

> svm\_polynomial <- svm(iris.data, iris.class, kernel="polynomial")

> svm\_radial <- svm(iris.data, iris.class, kernel="radial")

> svm\_sigmoid <- svm(iris.data, iris.class, kernel="sigmoid")

>

> # make prediction

> pred\_linear <- predict(svm\_linear,iris.data)

> pred\_polynomial <- predict(svm\_polynomial,iris.data)

> pred\_radial <- predict(svm\_radial,iris.data)

> pred\_sigmoid <- predict(svm\_sigmoid,iris.data)

>

> # confusion matrix

> table(pred\_linear, iris.class)

iris.class

pred\_linear setosa versicolor virginica

setosa 50 0 0

versicolor 0 46 1

virginica 0 4 49

> table(pred\_polynomial, iris.class)

iris.class

pred\_polynomial setosa versicolor virginica

setosa 50 0 0

versicolor 0 50 7

virginica 0 0 43

> table(pred\_radial, iris.class)

iris.class

pred\_radial setosa versicolor virginica

setosa 50 0 0

versicolor 0 48 2

virginica 0 2 48

> table(pred\_sigmoid, iris.class)

iris.class

pred\_sigmoid setosa versicolor virginica

setosa 49 0 0

versicolor 1 41 7

virginica 0 9 43

>

> # accuracy

> sum(pred\_linear == iris.class)/length(iris.class)

[1] 0.9666667

> sum(pred\_polynomial == iris.class)/length(iris.class)

[1] 0.9533333

> sum(pred\_radial == iris.class)/length(iris.class)

[1] 0.9733333

> sum(pred\_sigmoid == iris.class)/length(iris.class)

[1] 0.8866667

>

> # now dividing the data

> # compose training set

> index <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))

> iris.train\_data <- iris[index==1, 1:4]

> iris.test\_data <- iris[index==2, 1:4]

> iris.train\_class <- iris[index==1, 5]

> iris.test\_class <- iris[index==2, 5]

>

> # retrieve the model

> svm\_linear <- svm(x=iris.train\_data, y=iris.train\_class, kernel="linear")

> svm\_polynomial <- svm(x=iris.train\_data, y=iris.train\_class, kernel="polynomial")

> svm\_radial <- svm(x=iris.train\_data, y=iris.train\_class, kernel="radial")

> svm\_sigmoid <- svm(x=iris.train\_data, y=iris.train\_class, kernel="sigmoid")

>

> # make prediction

> pred\_linear <- predict(svm\_linear,iris.test\_data)

> pred\_polynomial <- predict(svm\_polynomial,iris.test\_data)

> pred\_radial <- predict(svm\_radial,iris.test\_data)

> pred\_sigmoid <- predict(svm\_sigmoid,iris.test\_data)

>

> # confusion matrix

> table(pred\_linear, iris.test\_class)

iris.test\_class

pred\_linear setosa versicolor virginica

setosa 15 0 0

versicolor 0 21 0

virginica 0 1 14

> table(pred\_polynomial, iris.test\_class)

iris.test\_class

pred\_polynomial setosa versicolor virginica

setosa 15 0 0

versicolor 0 22 5

virginica 0 0 9

> table(pred\_radial, iris.test\_class)

iris.test\_class

pred\_radial setosa versicolor virginica

setosa 15 0 0

versicolor 0 21 0

virginica 0 1 14

> table(pred\_sigmoid, iris.test\_class)

iris.test\_class

pred\_sigmoid setosa versicolor virginica

setosa 15 0 0

versicolor 0 15 0

virginica 0 7 14

>

> # accuracy

> sum(pred\_linear == iris.test\_class)/length(iris.test\_class)

[1] 0.9803922

> sum(pred\_polynomial == iris.test\_class)/length(iris.test\_class)

[1] 0.9019608

> sum(pred\_radial == iris.test\_class)/length(iris.test\_class)

[1] 0.9803922

> sum(pred\_sigmoid == iris.test\_class)/length(iris.test\_class)

[1] 0.8627451