SVM Notes

library(datasets)  
library(tidyverse)

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --  
## v dplyr 1.1.2 v readr 2.1.4  
## v forcats 1.0.0 v stringr 1.5.0  
## v ggplot2 3.4.2 v tibble 3.2.1  
## v lubridate 1.9.2 v tidyr 1.3.0  
## v purrr 1.0.1   
## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(grid)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(rattle)

## Loading required package: bitops  
## Rattle: A free graphical interface for data science with R.  
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(cluster)  
library(e1071)  
library(kernlab)

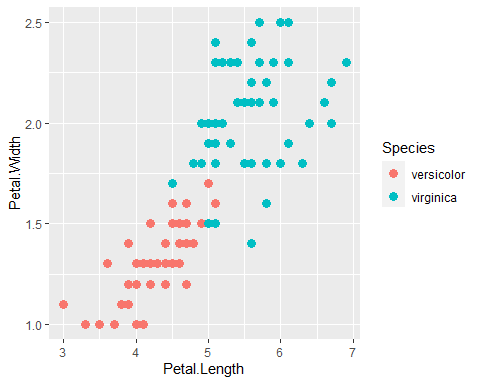
##   
## Attaching package: 'kernlab'  
##   
## The following object is masked from 'package:purrr':  
##   
## cross  
##   
## The following object is masked from 'package:ggplot2':  
##   
## alpha

library(caTools)

## Support Vector Machine

set.seed(3920)  
myiris2 = subset(iris, Species != "setosa")  
myiris2$Species = factor(myiris2$Species)  
svmsplit = sample.split(myiris2$Species, SplitRatio = 0.75)  
training.set = subset(myiris2, svmsplit==TRUE)  
test.set = subset(myiris2, svmsplit==FALSE)

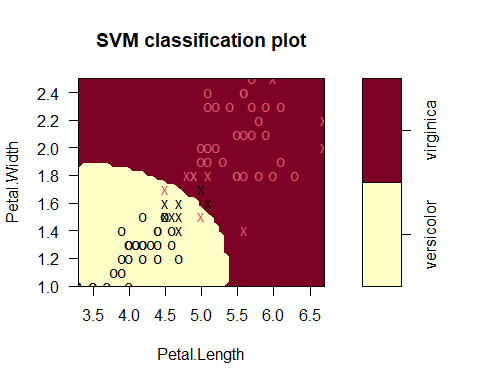
ggplot(data=myiris2,aes(x=Petal.Length,y=Petal.Width,color=Species))+geom\_point(size=3)



svmmodel = svm(factor(Species)~Petal.Width+Petal.Length, data=training.set, method="C-classification", kernal = "linear")  
svmmodel

##   
## Call:  
## svm(formula = factor(Species) ~ Petal.Width + Petal.Length, data = training.set,   
## method = "C-classification", kernal = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
##   
## Number of Support Vectors: 19

newtrain = subset(training.set, select = c(Petal.Length, Petal.Width, Species))  
plot(svmmodel, data = newtrain, formula=Petal.Width~Petal.Length)



pred.training = predict(svmmodel,training.set)  
mean(pred.training==training.set$Species)

## [1] 0.9473684

kern\_svm=ksvm(Species~Petal.Width+Petal.Length, training.set, kernel="vanilladot")

## Setting default kernel parameters

kern\_svm

## Support Vector Machine object of class "ksvm"   
##   
## SV type: C-svc (classification)   
## parameter : cost C = 1   
##   
## Linear (vanilla) kernel function.   
##   
## Number of Support Vectors : 14   
##   
## Objective Function Value : -10.412   
## Training error : 0.052632

pred.test = predict(kern\_svm, training.set)  
mean(pred.test == training.set$Species)

## [1] 0.9473684

plot(kern\_svm, data = training.set, formula = Petal.Width ~ Petal.Length)

