SVM

Jaymeen Gandhi

5/7/2020

require(tidyverse)

## Loading required package: tidyverse

## -- Attaching packages ----------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts -------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

social\_data = read\_csv("C:/Users/jayme/Downloads/social\_ads.csv")

## Parsed with column specification:  
## cols(  
## `User ID` = col\_double(),  
## Gender = col\_character(),  
## Age = col\_double(),  
## EstimatedSalary = col\_double(),  
## Purchased = col\_double()  
## )

social\_ready = social\_data %>% select(-`User ID`) %>% mutate(Purchased = ifelse(Purchased==1 , TRUE, FALSE)) %>% mutate\_if(is.numeric,scale)

set.seed(123)  
social\_ready = social\_ready %>% mutate(id = row\_number())  
train\_sd = social\_ready %>% sample\_frac(0.75)  
test\_sd = anti\_join(social\_ready,train\_sd,by='id')

require(e1071)

## Loading required package: e1071

fm =formula('Purchased ~ .-id')  
sd\_classifier = svm(formula = fm ,data = train\_sd ,type = 'C-classification',kernel='linear')  
summary(sd\_classifier)

##   
## Call:  
## svm(formula = fm, data = train\_sd, type = "C-classification", kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
##   
## Number of Support Vectors: 118  
##   
## ( 60 58 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## FALSE TRUE

my\_predictions = predict(sd\_classifier,newdata=test\_sd[-4])

require(caret)

## Loading required package: caret

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

confusionMatrix(my\_predictions,reference = as.factor(test\_sd$Purchased))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 63 11  
## TRUE 5 21  
##   
## Accuracy : 0.84   
## 95% CI : (0.7532, 0.9057)  
## No Information Rate : 0.68   
## P-Value [Acc > NIR] : 0.0002249   
##   
## Kappa : 0.6132   
##   
## Mcnemar's Test P-Value : 0.2112995   
##   
## Sensitivity : 0.9265   
## Specificity : 0.6562   
## Pos Pred Value : 0.8514   
## Neg Pred Value : 0.8077   
## Prevalence : 0.6800   
## Detection Rate : 0.6300   
## Detection Prevalence : 0.7400   
## Balanced Accuracy : 0.7914   
##   
## 'Positive' Class : FALSE   
##

p\_sd\_classifier = svm(formula = fm ,data = train\_sd ,type = 'C-classification',kernel='polynomial',cost=2)  
my\_predictions2 = predict(p\_sd\_classifier,newdata=test\_sd[-4])  
confusionMatrix(my\_predictions2,reference = as.factor(test\_sd$Purchased))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 64 4  
## TRUE 4 28  
##   
## Accuracy : 0.92   
## 95% CI : (0.8484, 0.9648)  
## No Information Rate : 0.68   
## P-Value [Acc > NIR] : 9.698e-09   
##   
## Kappa : 0.8162   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9412   
## Specificity : 0.8750   
## Pos Pred Value : 0.9412   
## Neg Pred Value : 0.8750   
## Prevalence : 0.6800   
## Detection Rate : 0.6400   
## Detection Prevalence : 0.6800   
## Balanced Accuracy : 0.9081   
##   
## 'Positive' Class : FALSE   
##

r\_sd\_classifier = svm(formula = fm ,data = train\_sd ,type = 'C-classification',kernel='radial',cost =10)  
my\_predictions3 = predict(p\_sd\_classifier,newdata=test\_sd[-4])  
confusionMatrix(my\_predictions3,reference = as.factor(test\_sd$Purchased))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 64 4  
## TRUE 4 28  
##   
## Accuracy : 0.92   
## 95% CI : (0.8484, 0.9648)  
## No Information Rate : 0.68   
## P-Value [Acc > NIR] : 9.698e-09   
##   
## Kappa : 0.8162   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9412   
## Specificity : 0.8750   
## Pos Pred Value : 0.9412   
## Neg Pred Value : 0.8750   
## Prevalence : 0.6800   
## Detection Rate : 0.6400   
## Detection Prevalence : 0.6800   
## Balanced Accuracy : 0.9081   
##   
## 'Positive' Class : FALSE   
##