Classify the Size\_Categorie using SVM

> library(caret)

> library(ggplot2)

> library(psych)

> library(kernlab)

> forest <- read.csv(file.choose())

> View(forest)

> str(forest)

'data.frame': 517 obs. of 31 variables:

$ month : Factor w/ 12 levels "apr","aug","dec",..: 8 11 11 8 8 2 2 2 12 12 ...

$ day : Factor w/ 7 levels "fri","mon","sat",..: 1 6 3 1 4 4 2 2 6 3 ...

$ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...

$ DMC : num 26.2 35.4 43.7 33.3 51.3 ...

$ DC : num 94.3 669.1 686.9 77.5 102.2 ...

$ ISI : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...

$ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...

$ RH : int 51 33 33 97 99 29 27 86 63 40 ...

$ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...

$ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...

$ area : num 0 0 0 0 0 0 0 0 0 0 ...

$ dayfri : int 1 0 0 1 0 0 0 0 0 0 ...

$ daymon : int 0 0 0 0 0 0 1 1 0 0 ...

$ daysat : int 0 0 1 0 0 0 0 0 0 1 ...

$ daysun : int 0 0 0 0 1 1 0 0 0 0 ...

$ daythu : int 0 0 0 0 0 0 0 0 0 0 ...

$ daytue : int 0 1 0 0 0 0 0 0 1 0 ...

$ daywed : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthapr : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthaug : int 0 0 0 0 0 1 1 1 0 0 ...

$ monthdec : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthfeb : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjan : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjul : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjun : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthmar : int 1 0 0 1 1 0 0 0 0 0 ...

$ monthmay : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthnov : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthoct : int 0 1 1 0 0 0 0 0 0 0 ...

$ monthsep : int 0 0 0 0 0 0 0 0 1 1 ...

$ size\_category: Factor w/ 2 levels "large","small": 2 2 2 2 2 2 2 2 2 2 ...

> table(forest$size\_category)

large small

139 378

#Creating dummies

> forest$month=as.integer(factor(forest$month,levels = c("jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"),labels = c(1,2,3,4,5,6,7,8,9,10,11,12)))

> forest$day=as.integer(factor(forest$day,levels = c("sun","mon","tue","wed","thu","fri","sat"),labels = c(1,2,3,4,5,6,7)))

> forest$size\_category=as.integer(factor(forest$size\_category,levels = c("large","small"),labels = c(1,0)))

> str(forest)

'data.frame': 517 obs. of 31 variables:

$ month : int 3 10 10 3 3 8 8 8 9 9 ...

$ day : int 6 3 7 6 1 1 2 2 3 7 ...

$ FFMC : num 86.2 90.6 90.6 91.7 89.3 92.3 92.3 91.5 91 92.5 ...

$ DMC : num 26.2 35.4 43.7 33.3 51.3 ...

$ DC : num 94.3 669.1 686.9 77.5 102.2 ...

$ ISI : num 5.1 6.7 6.7 9 9.6 14.7 8.5 10.7 7 7.1 ...

$ temp : num 8.2 18 14.6 8.3 11.4 22.2 24.1 8 13.1 22.8 ...

$ RH : int 51 33 33 97 99 29 27 86 63 40 ...

$ wind : num 6.7 0.9 1.3 4 1.8 5.4 3.1 2.2 5.4 4 ...

$ rain : num 0 0 0 0.2 0 0 0 0 0 0 ...

$ area : num 0 0 0 0 0 0 0 0 0 0 ...

$ dayfri : int 1 0 0 1 0 0 0 0 0 0 ...

$ daymon : int 0 0 0 0 0 0 1 1 0 0 ...

$ daysat : int 0 0 1 0 0 0 0 0 0 1 ...

$ daysun : int 0 0 0 0 1 1 0 0 0 0 ...

$ daythu : int 0 0 0 0 0 0 0 0 0 0 ...

$ daytue : int 0 1 0 0 0 0 0 0 1 0 ...

$ daywed : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthapr : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthaug : int 0 0 0 0 0 1 1 1 0 0 ...

$ monthdec : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthfeb : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjan : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjul : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthjun : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthmar : int 1 0 0 1 1 0 0 0 0 0 ...

$ monthmay : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthnov : int 0 0 0 0 0 0 0 0 0 0 ...

$ monthoct : int 0 1 1 0 0 0 0 0 0 0 ...

$ monthsep : int 0 0 0 0 0 0 0 0 1 1 ...

$ size\_category: int 2 2 2 2 2 2 2 2 2 2 ...

#Normalising

> norm <- function(x){

return((x-min(x))/(max(x)-min(x)))

}

> forest\_norm <- as.data.frame(lapply(forest,norm))

> head(forest\_norm)

month day FFMC DMC DC ISI temp

1 0.1818182 0.8333333 0.8709677 0.08649207 0.10132520 0.09090909 0.1929260

2 0.8181818 0.3333333 0.9277419 0.11819435 0.77541926 0.11942959 0.5080386

3 0.8181818 1.0000000 0.9277419 0.14679531 0.79629412 0.11942959 0.3987138

4 0.1818182 0.8333333 0.9419355 0.11095796 0.08162308 0.16042781 0.1961415

5 0.1818182 0.0000000 0.9109677 0.17298415 0.11058989 0.17112299 0.2958199

6 0.6363636 0.0000000 0.9496774 0.29014473 0.56303507 0.26203209 0.6430868

RH wind rain area dayfri daymon daysat daysun daythu daytue

1 0.4235294 0.70000000 0.00000 0 1 0 0 0 0 0

2 0.2117647 0.05555556 0.00000 0 0 0 0 0 0 1

3 0.2117647 0.10000000 0.00000 0 0 0 1 0 0 0

4 0.9647059 0.40000000 0.03125 0 1 0 0 0 0 0

5 0.9882353 0.15555556 0.00000 0 0 0 0 1 0 0

6 0.1647059 0.55555556 0.00000 0 0 0 0 1 0 0

daywed monthapr monthaug monthdec monthfeb monthjan monthjul monthjun monthmar

1 0 0 0 0 0 0 0 0 1

2 0 0 0 0 0 0 0 0 0

3 0 0 0 0 0 0 0 0 0

4 0 0 0 0 0 0 0 0 1

5 0 0 0 0 0 0 0 0 1

6 0 0 1 0 0 0 0 0 0

monthmay monthnov monthoct monthsep size\_category

1 0 0 0 0 1

2 0 0 1 0 1

3 0 0 1 0 1

4 0 0 0 0 1

5 0 0 0 0 1

6 0 0 0 0 1

#Splitting of data to test and train

> trns <- createDataPartition(forest\_norm$area,p=0.8,list = F)

> tn <- forest\_norm[trns,]

> ts <- forest\_norm[-trns,]

#Model 1 : rbfdot

> modelrbfdot <- ksvm(area~.,data=tn,kernel="rbfdot")

> predrbfdot <- predict(modelrbfdot,newdata=ts)

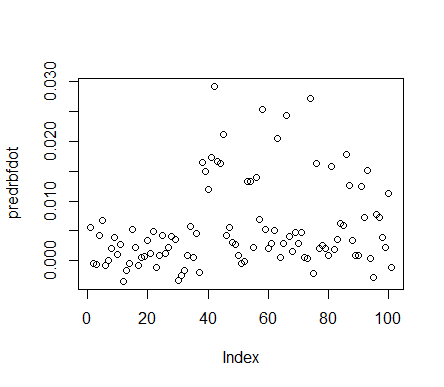
> cor(predrbfdot,ts$area)

[,1]

[1,] 0.3431461

Accuracy is 34.31%

> plot(predrbfdot)



#Model 2 : besseldot

> modelbessel <- ksvm(area~.,data=tn,kernel="besseldot")

Setting default kernel parameters

> predbessel <- predict(modelbessel,newdata=ts)

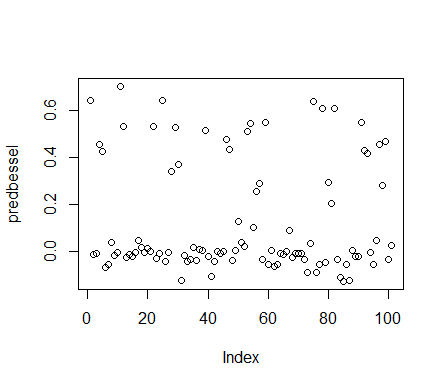
> cor(predbessel,ts$area)

[,1]

[1,] -0.04111227

Accuracy is 4.11%

> plot(predbessel)



#Model 3: polydot

> modelpoly <- ksvm(area~.,data=tn,kernel="polydot")

Setting default kernel parameters

> predpoly <- predict(modelpoly,newdata=ts)

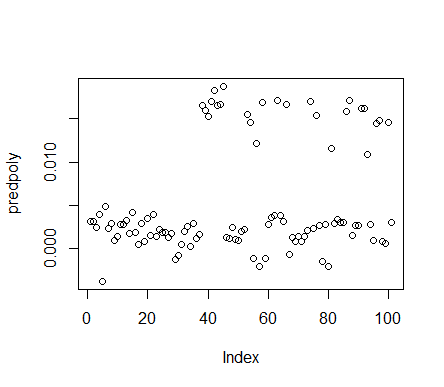
> cor(predpoly,ts$area)

[,1]

[1,] 0.3455455

Accuracy is 34.55%

> plot(predpoly)



1. **Model 4: vanilladot**

> modelvanilla <- ksvm(area~.,data=tn,kernel="vanilladot")

Setting default kernel parameters

> predvanilla <- predict(modelvanilla,newdata=ts)

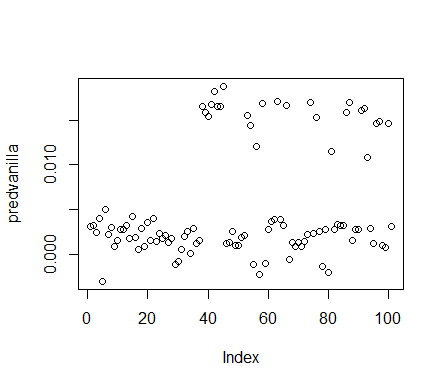
> cor(predvanilla,ts$area)

[,1]

[1,] 0.3464077

Accuracy is 34.64%

> plot(predvanilla)



1. **Model 5: tanhdot**

> modeltanh <- ksvm(area~.,data=tn,kernel="tanhdot")

Setting default kernel parameters

> predtanh <- predict(modeltanh,newdata=ts)

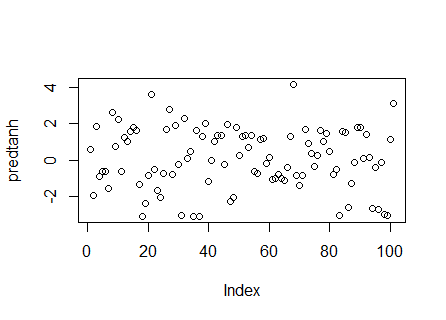
> cor(predtanh,ts$area)

[,1]

[1,] 0.005924511

Accuracy is 0.59%

> plot(predtanh)



**6. Model 6 : anovadot**

> modelanova <- ksvm(area~.,data=tn,kernel="anovadot")

> predanova <- predict(modelanova,newdata=ts)

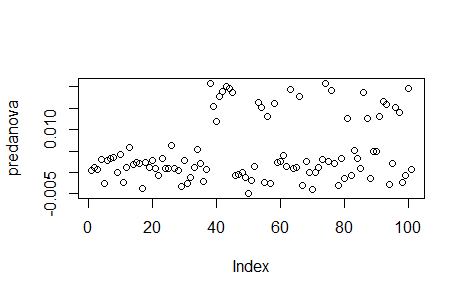
> cor(predanova,ts$area)

[,1]

[1,] 0.3368088

Accuracy is 33.68%

> plot(predanova)



**Kernel vanilladot method has highest accuracy with 34.64%**