Trabajo Práctico Final SIA



*Agustin Chaud - 63253*

**Ejercicio 1 – Aprendizaje Supervisado**

**Parte A – Análisis Exploratorio de los Datos**

1. **Considere el archivo phoneme.data de la web del libro The Elements of**

**Statistical Learning (Hastie, Tibshirani, Friedman)**

2) **Abra Info y copie los ejemplos de palabras en inglés que el autor usa para cada fonema.**

Los ejemplos de palabras en inglés que el autor usa para cada fonema :

"sh" --> "she", "dcl" --> "dark", "iy" --> "she", "aa" --> "dark", "ao" --> "water"

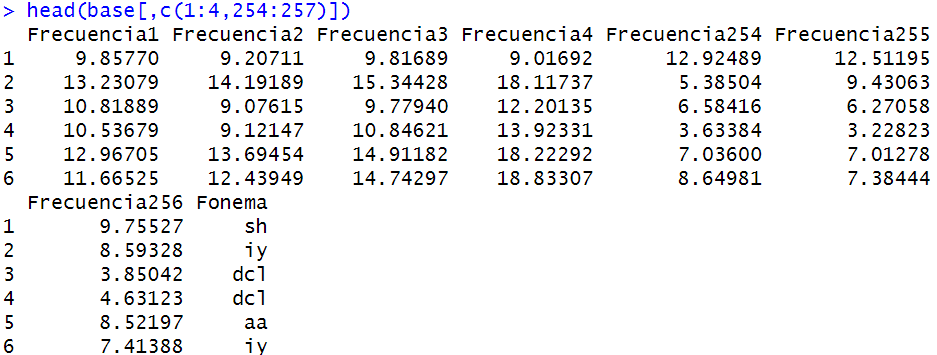
3) **¿Para qué podría servir predecir qué fonema se está diciendo?**

Podría servir para sistemas de vigilancia que escuchen las conversaciones de criminales a través de sus celulares, y ayuden a predecir delitos. Otra utilidad podría ser generar subtítulos automáticos a los videos.

4) **Abra el archivo "phoneme.data" en R. Muestre dim(base)**



5) **Mostrar el head truncado de la base con los cambios**



6) **Indique cuántos fonemas de cada tipo de fonema tiene el dataset.**

Hay 695 fonemas de tipo “aa”, 1022 de tipo “ao”, 757 de tipo “dcl”, 1163 de tipo “iy” y 872 de tipo “sh”

7) **¿Cuál es el fonema que tiene más elementos en el dataset?**

El fonema con más elementos en el dataset es "iy"

8) **Realice un gráfico de barras de la variable a predecir.**

plot(base$Fonema,col="steelblue4",main="Gráfico de Agustin Chaud")

9) **Considere los últimos 3 números de su LEGAJO (3numLEGAJO) y muestre los valores del fonema de la base en esa fila.**

Tiene el fonema "ao"

**Parte B – Conjuntos**

1. **Para el seteo de semilla considere su LEGAJO (completo) y particione la base en un conjunto de entrenamiento y uno de testeo, utilizando la instrucción createDataPartition de la librería caret.**

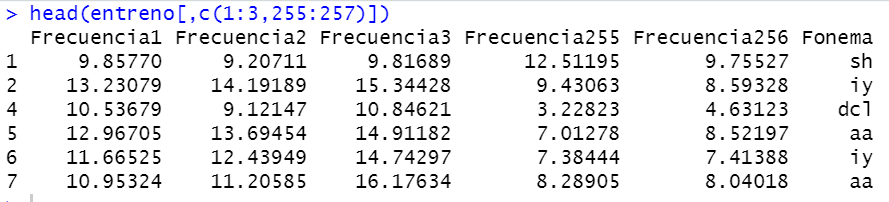
library(caret)

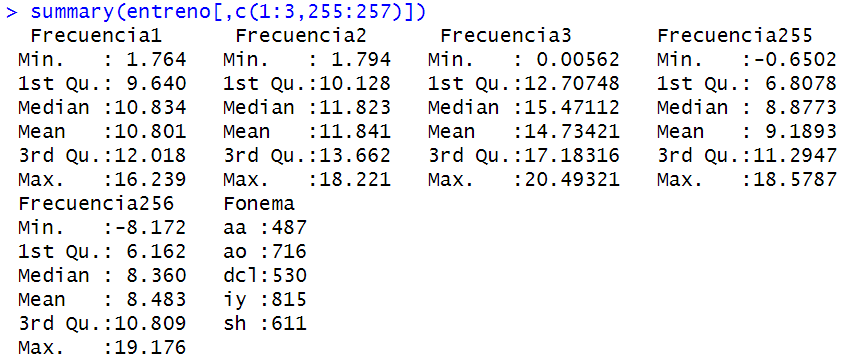
set.seed(63253);particion=createDataPartition(y=base$Fonema,p=0.70,list=FALSE)

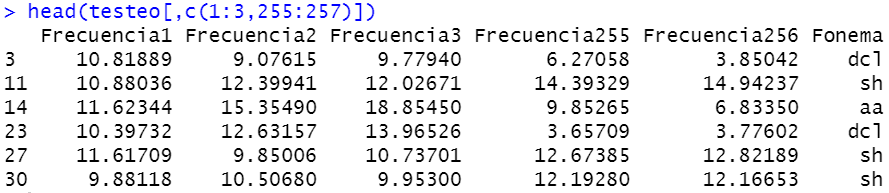
entreno=base[particion,]

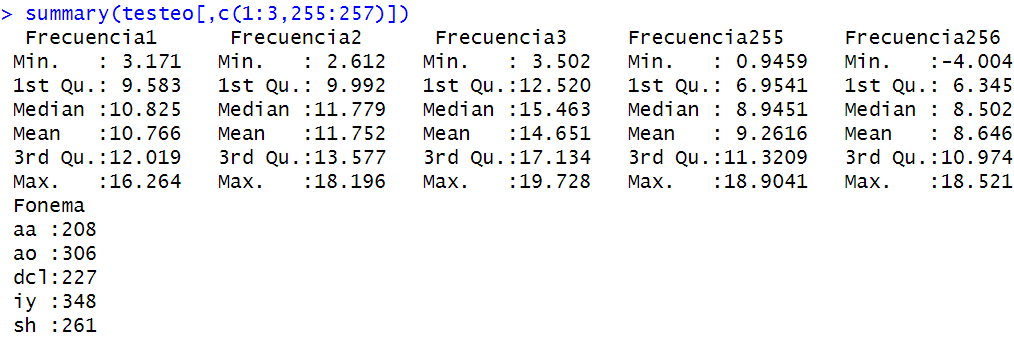
testeo=base[-particion,]

1. **Muestre un head truncado y un summary truncado del conjunto de entrenamiento. También un head truncado y un summary del conjunto de testeo**









1. **Haga una tabla comparando por fonema cuántos quedaron de cada uno en la base, en entrenamiento y en testeo.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | aa | ao | dcl | iy | sh |
| base | 695 | 1022 | 757 | 1163 | 872 |
| entreno | 487 | 716 | 530 | 815 | 611 |
| testeo | 208 | 306 | 227 | 348 | 261 |

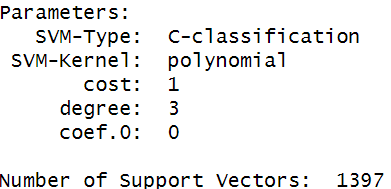
**Parte C – SVM1**

1. **Cargue la librería e1071 y modele el problema planteado con una Support Vector Machine con un kernel = (asignado) y los parámetros restantes con los valores por defecto sin cambiar**

library(e1071)

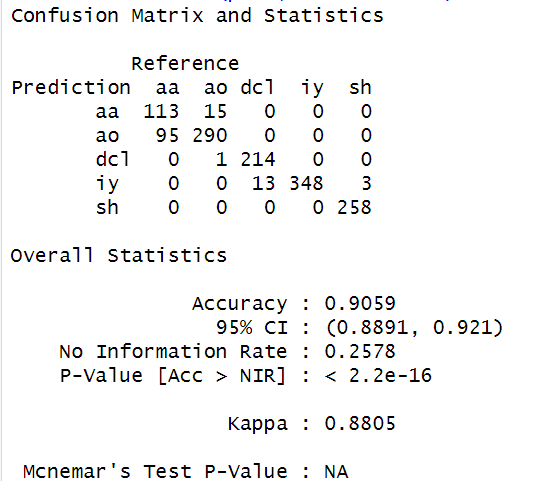
vm=svm(Fonema~.,entreno,kernel="polynomial")

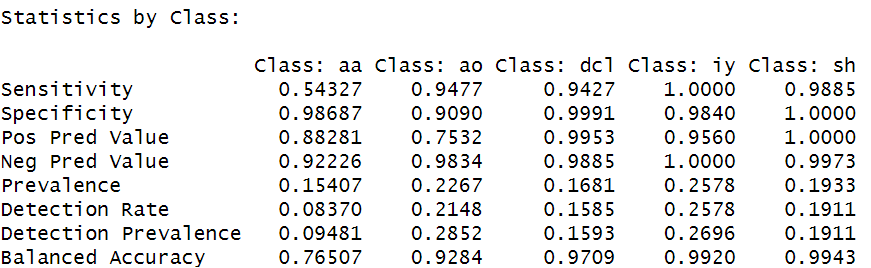
1. **Escriba svm y muestre una captura de pantalla de la información que aparece. ¿Cuántos vectores soporte aparecen? ¿Cuál es el SVM Type?**



Aparecen 1397 vectores soporte y el SVM-type es C-classification

**Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de los resultados completos**

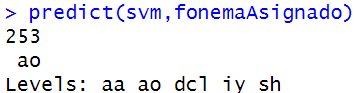




1. **¿Cuál fue el accuracy?**

El accuracy fue de 0.9059

1. **Prediga con la SVM el registro de los 3 últimos números de su LEGAJO y muestre la predicción**



Efectivamente, la predicción coincide con lo esperado

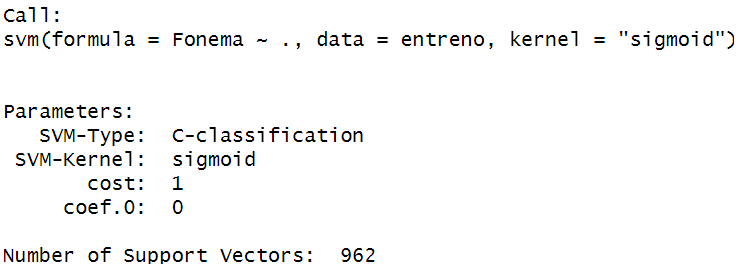
**Parte D – SVM2**

1. **Cargue la librería e1071 y modele el problema planteado con una Support Vector Machine con alguna variación de la SVM del punto anterior**

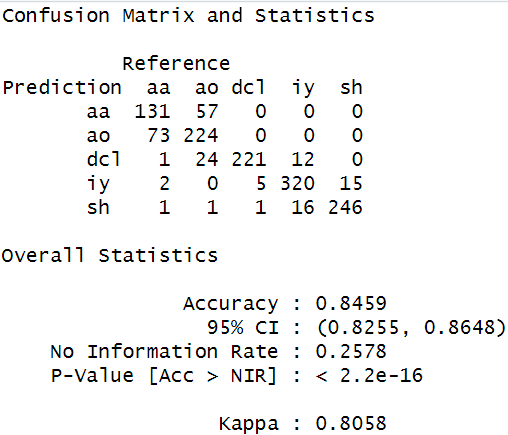
library(e1071)

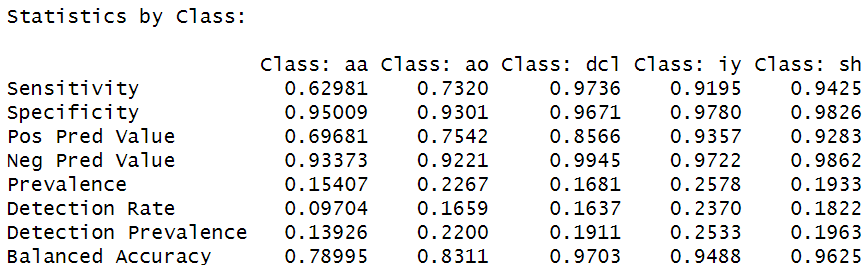
vm2=svm(Fonema~.,entreno,kernel="sigmoid")

1. **Escriba svm y muestre una captura de pantalla de la información que aparece.**

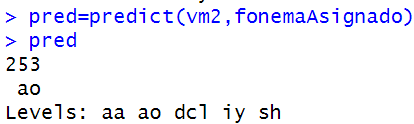


1. **Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de los resultados completos**





1. Prediga con la SVM el registro de los 3 últimos números de su LEGAJO y muestre la predicción

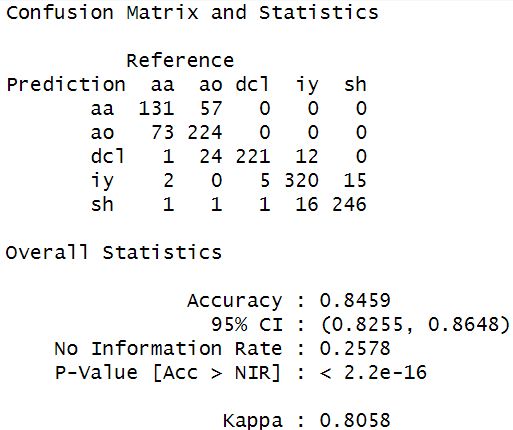


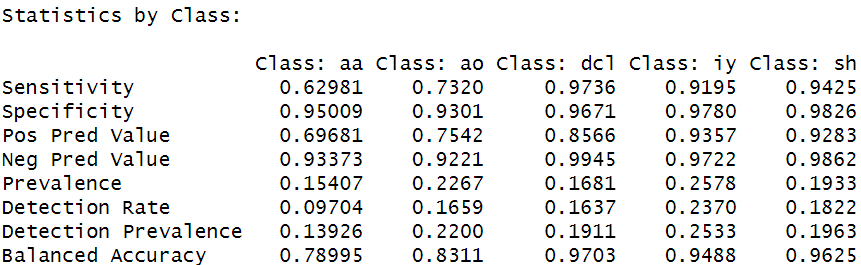
Efectivamente, la predicción coincide con lo esperado

**Parte E – Naive Bayes**

1) **Cargue la librería e1071 y modele el problema planteado utilizando Naive Bayes**

2) **Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de los resultados completos**

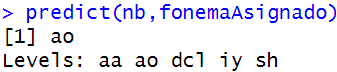




3) **¿Cuál fue el accuracy?**

El accuracy fue de 0.8459

4 ) **Prediga con Naive Bayes el registro de los 3 últimos números de su LEGAJO y muestre la predicción.**



Efectivamente, la predicción coincide con lo esperado

**Parte F – Comparación**

1) **Cree una tabla con el accuracy de cada uno de los 3 modelos, y la sensibilidad y especificidad de cada modelo por categoría**

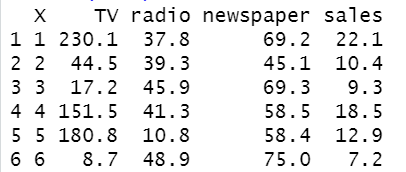
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mod | Sen aa | Sen ao | Sen dcl | Sen iy | Sen sh | Spe aa | Spe ao | Spe dcl | Spe iy | Spe sh | Accu |
| Bayes | 0.62981 | 0.7320 | 0.9736 | 0.9195 | 0.9425 | 0.95 | 0.93 | 0.9671 | 0.978 | 0.9826 | 0.8748 |
| Svm1 | 0.5432 | 0.9477 | 0.9427 | 1 | 0.9885 | 0.9868 | 0.9090 | 0.9991 | 0.9840 | 1 | 0.9059 |
| Svm2 | 0.6298 | 0.7320 | 0.9736 | 0.9195 | 0.9425 | 0.95 | 0.93 | 0.9671 | 0.9780 | 0.9826 | 0.8459 |

1. **Compare los resultados obtenidos (accuracy, sensibilidades y especificidades) con la SVM1, SVM2 y Naive Bayes. ¿Cuál modelo le parece que resultó mejor? ¿Según qué criterio?**

Resultó mejor el Svm 1 ya que tiene mayor accuracy y mayor sensibilidad y especifidad en la mayoría de los casos.

**Ejercicio 2 – Aprendizaje No Supervisado**

1. **Considere el archivo Advertising.csv de la página web del libro “Introduction to Statistical Learning with Applications in R”**
2. **Muestre un head(base)**

****

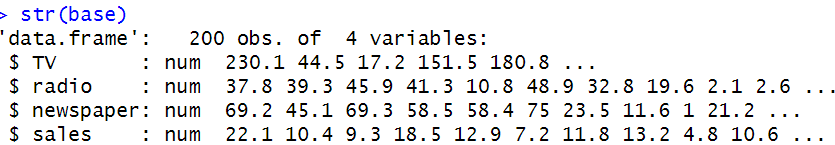
1. **¿Qué tiene como valores la variable X?**

La variable x es un identificador que se autoincrementa

1. **Borre la variable “X”**
2. **La variable a predecir es sales, ¿cuáles son las variables predictoras?**

Las variables predictoras son TV, RADIO y NEWSPAPER

1. **Muestre un str de la base.**

****

1. **¿Cuántos registros tiene la base?**

La base tiene 200 registros

1. **Elija una variable (que no sea sales). Realice un gráfico de dispersión con esa variable en el eje X y la variable a predecir sales en el eje Y. Indique el código R utilizado**

plot(base$TV,base$sales,main= "Gráfico de Dispersión de Agustín Chaud", col="steelblue4", pch=21)

**Parte B – Red Neuronal Kohonen**

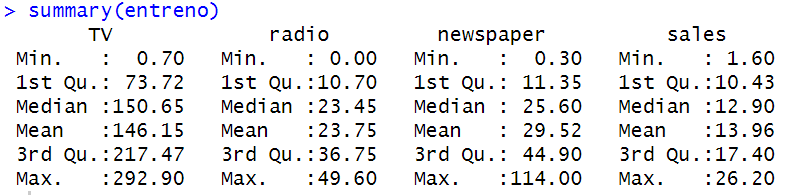
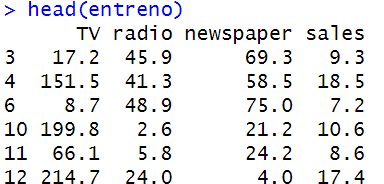
1. **Con la librería caret, particione la base en entrenamiento y test. Considere para el seteo de la semilla su número de Legajo (completo).**

set.seed(63253);particion=createDataPartition(y=base$sales,p=0.70,list=FALSE)

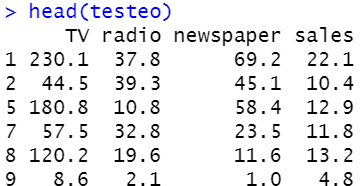
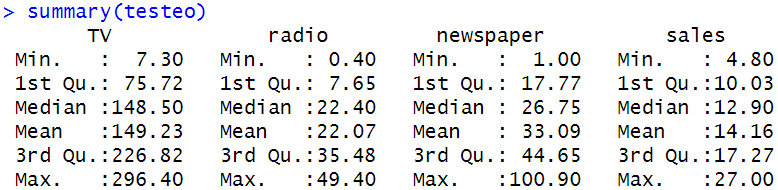
entreno=base[particion,]

testeo=base[-particion,]

1. **Muestre un head y un summary del conjunto de entrenamiento**

****

1. **Muestre un head y un summary del conjunto de testeo.**

****

1. **¿Cuántos registros quedaron en entrenamiento y testeo?**

**En entreno quedaron 142 registros y en testeo 58**

**Parte C – Modelado**

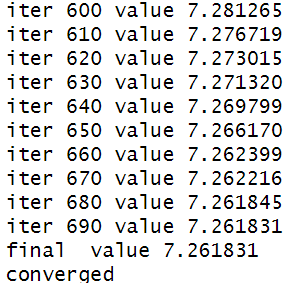
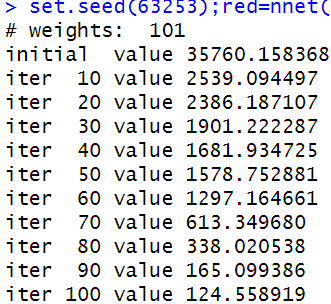
**Para la red neuronal**

1. **Setee la semilla=Legajo (completo) y cree una Red Neuronal con la librería nnet para modelar el problema planteado**
2. **Indique el código R utilizado**

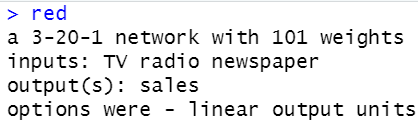
library(nnet)

set.seed(63253);red=nnet(sales~.,entreno,size=20,maxit=10000,linout=TRUE)

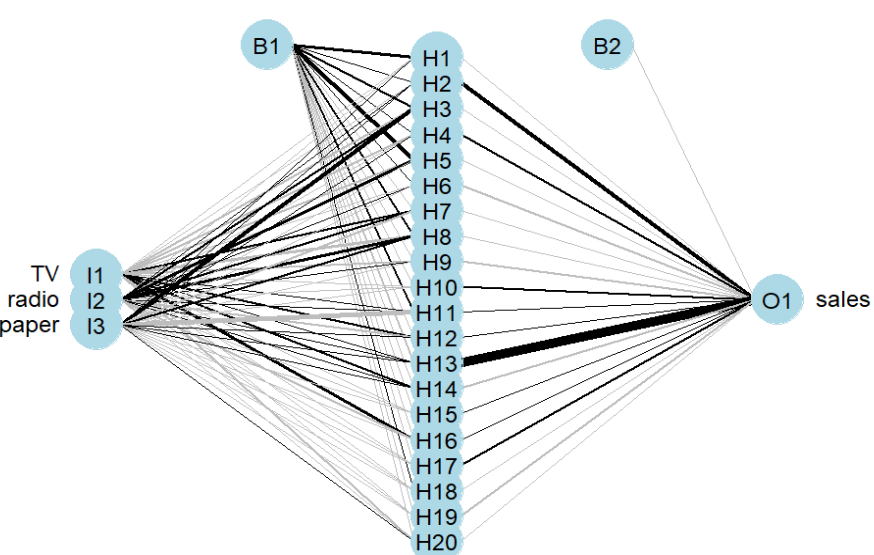
1. **Muestre una captura de pantalla de la lista de iteraciones de la Red Neuronal. Si son muchas, muestre las primeras y las últimas**

****

1. **Escriba el nombre de la red (por ejemplo red) y muestre una captura de pantalla de la información que aparece.**

****

1. **Dibuje la Red Neuronal.**

****

1. **Indique el Error Cuadrático Medio.**

El error cuadrático medio es de 0.1873

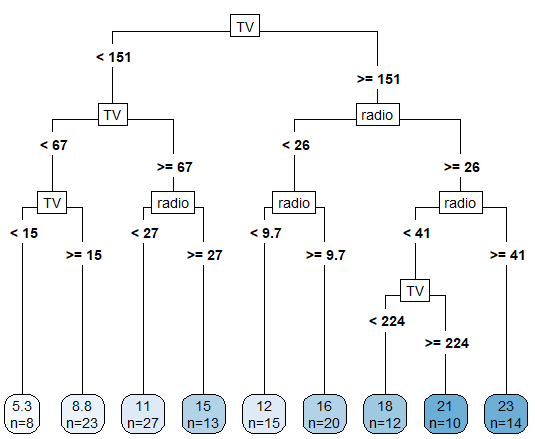
**Para el árbol de decisión**

1. **Cree el Árbol de Decisión con la librería rpart. Indique el código R utilizado (recuerde que es un caso de Regresión). No lo pode.**

library(rpart)

arbol=rpart(sales~., entreno)

1. **Dibuje el Árbol de Decisión obtenido**

****

1. **Indique el Error Cuadrático Medio**

El error cuadrático medio es de 2.6932

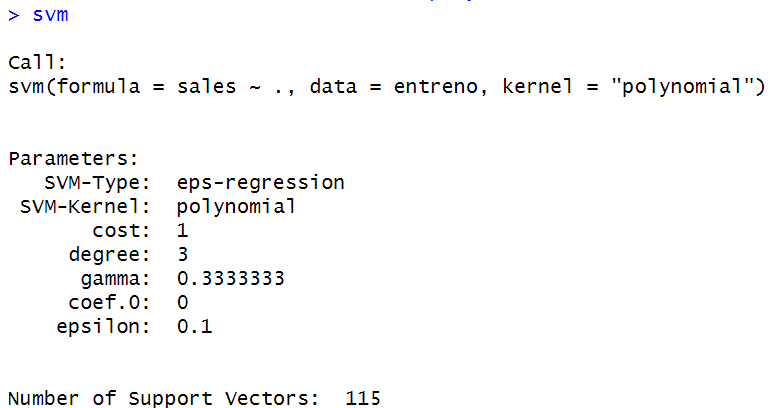
**Para la SVM**

1. **Cree la SVM con la librería e1071. Utilice el kernel asignado en el ejercicio anterior (la primera SVM1) y los parámetros restantes con los valores por defecto sin cambiar.**
2. **Indique el código R utilizado**

library(e1071)

svm=svm(sales~., entreno,kernel="polynomial")

1. **Escriba el nombre de la svm (por ejemplo svm) y muestre una captura de pantalla de la información que aparece. ¿De qué tipo es SVM-Type?**



1. **Indique el Error Cuadrático Medio**

El error cuadrático medio es de 11.53646

**Parte D – Comparación**

1. **Realizar una tabla comparando los errores cuadráticos medios de cada uno de los 3 modelos**

|  |  |  |  |
| --- | --- | --- | --- |
|  | RNN | ADD | SVM |
| ECM | 0.1873 | 2.6932 | 11.53646 |

1. **Compare los resultados obtenidos con la Red Neuronal, el Árbol de Decisión, y la Support Vector Machine. ¿Cuál modelo le parece que resultó mejor?**

La Red Neuronal es la mejor ya que tiene el menor error cuadrático medio

Anexo

base=read.table("phoneme.data",sep=",",header=TRUE)

#head(base)

dim(base)

base$speaker=NULL

base$row.names=NULL

dim(base)

for (i in 1:256){

var=paste("x.",i,sep="")

nuevo=paste("Frecuencia",i,sep="")

names(base)[names(base)==var]=nuevo}

names(base)[names(base)=="g"]="Fonema"

base$Fonema=factor(base$Fonema)

#head(base)

head(base[,c(1:4,254:257)])

table(base$Fonema) # Hay 695 fonemas de tipo aa, 1022 de tipo ao, 757 de tipo dcl, 1163 de tipo iy y 872 de tipo sh

# el fonema con mas elementos en el dataset es "iy"

plot(base$Fonema,col="steelblue4",main="Gráfico de Agustin Chaud")

fonemaAsignado=base[253,]

fonemaAsignado # tiene el fonema "ao"

# parte B

library(caret)

set.seed(63253);particion=createDataPartition(y=base$Fonema,p=0.70,list=FALSE)

entreno=base[particion,]

testeo=base[-particion,]

head(entreno[,c(1:3,255:257)])

summary(entreno[,c(1:3,255:257)])

head(testeo[,c(1:3,255:257)])

summary(testeo[,c(1:3,255:257)])

table(base$Fonema);table(entreno$Fonema);table(testeo$Fonema)

#---parte C--

library(e1071)

svm=svm(Fonema~.,entreno,kernel="polynomial")

svm #Aparecen 1397 vectores soporte... El SVM-type es C-classification

#library(caret)

pred=predict(svm,testeo)

confusionMatrix(pred,testeo$Fonema)

predict(svm,fonemaAsignado)

#---------parte D---------

library(e1071)

vm2=svm(Fonema~.,entreno,kernel="sigmoid")

vm2

pred=predict(vm2,testeo)

confusionMatrix(pred,testeo$Fonema)

pred=predict(vm2,fonemaAsignado)

#-------parte E---------

library(e1071)

nb=naiveBayes(Fonema~.,entreno)

pred=predict(nb,testeo)

confusionMatrix(pred,testeo$Fonema)

predict(nb,fonemaAsignado)

#------------ej 2------------

base=read.csv("Advertising.csv",sep=",",header=TRUE)

head(base)

base$X=NULL

head(base)

str(base)

dim(base)

plot(base$TV,base$sales,main= "Gráfico de Dispersión de Agusín Chaud", col="steelblue4", pch=21)

#----parte B

library(caret)

set.seed(63253);particion=createDataPartition(y=base$sales,p=0.70,list=FALSE)

entreno=base[particion,]

testeo=base[-particion,]

head(entreno)

summary(entreno)

dim(entreno)

head(testeo)

summary(testeo)

dim(testeo)

#--------parte C---

library(nnet)

set.seed(63253);red=nnet(sales~.,entreno,size=20,maxit=10000,linout=TRUE)

red

library(NeuralNetTools)

plotnet(red)

pred=predict(red, testeo) #sin type=“class”

ECMRN=mean((pred-testeo$sales)^2)

#ADD

library(rpart)

arbol=rpart(sales~., entreno)

library(rpart.plot)

rpart.plot(arbol,extra=1,type=5)

pred=predict(arbol, testeo) #sin type=“class”

ECMArbol=mean((pred-testeo$sales)^2)

#SVM

library(e1071)

svm=svm(sales~., entreno,kernel="polynomial")

svm

pred=predict(svm, testeo)

ECMSvm=mean((pred-testeo$sales)^2)