Laboratorio 4

Nota: **Resuelva todos los ejercicios utilizando primero rattle y luego implementando por línea de comando. Incluya en el informe un anexo con las líneas de código generadas.**

Ejercicio 1: Esta pregunta utiliza los datos sobre muerte del corazón en Sudáfrica (SAheart.csv). La variable a predecir es chd, un indicador de muerte coronaria. La predicción se construirá basándose en algunas variables predictivas (factores de riesgo) como son el ser fumador, la obesidad, el consumo de bebidas alcohólicas, entre otras.

1. Use el método de Máquinas de soporte vectorial para generar un modelo predictivo para la tabla SAheart.csv usando el 75% de los datos para la tabla de aprendizaje y un 25% para la tabla de prueba.

2. Calcule para los datos de prueba la matriz de confusión, la precisión global, la precisión positiva, la precisión negativa, los falsos positivos, los falsos negativos.

3. Grafique la curva ROC para el modelo generado en 1. Comente.

4. Genere un Modelo Predictivo usando Máquinas de Soporte Vectorial con los siguientes núcleos: Radial Basis, Polynomial, Linear, HyperbolicTangent, Laplacian, Bessel, Anova RBF y Spline ¿Cuál produce los mejores resultados?

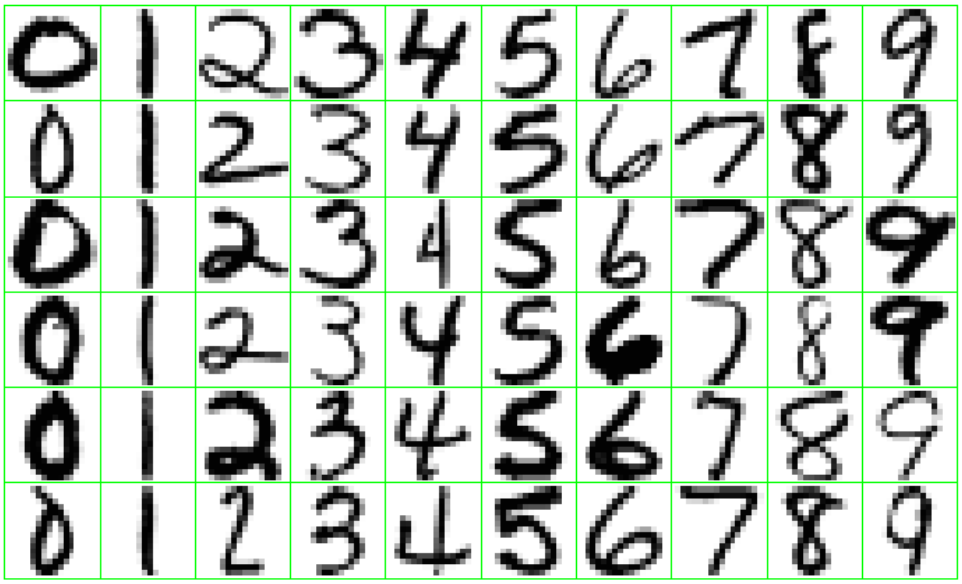
Ejercicio 2: Esta pregunta utiliza los datos "CompraBicicletas.csv", que contienen información de 11 variables (predictores) y de la variable"PurchasedBike" que da cuenta de si un cliente compró o no una bicicleta.

1. Use el método de máquinas de soporte vectorial para generar un modelo predictivo para la tabla PurchasedBike.csv usando 70% de los datos para tabla de aprendizaje y un 30% para la tabla de prueba. ¿Cuántos vectores de soporte requirió el modelo?

2. Calcule para los datos de prueba y para toda la tabla la matriz de confusión y la precisión global. Interprete la calidad de los resultados.

3. Grafique la curva ROC para este modelo.

Ejercicio 3: El objetivo de este ejercicio es utilizar el método de árboles de decisión para predecir números escritos a mano (Hand WrittenDigitRecognition). Las tablas de aprendizaje y prueba ya han sido previamente preparadas y se encuentran en los archivos "ZipDataTrainCod.csv" y "ZipDataTestCod.csv".



Los datos corresponden a códigos postales escritos a mano en sobres del correo postal de EE.UU. Más precisamente, cada dígito está representado por una imagen de 16 × 16 en escala de grises, cada pıxel con intensidad de −1 a 1 (de blanco a negro). Las imágenes se han normalizado para tener aproximadamente el mismo tamaño y orientación. La tarea consiste en predecir, a partir de la matriz de 16×16 de intensidades de cada pıxel, la identidad de cada imagen (0, 1, . . . , 9) de forma rápida y precisa. De lograrse el cometido con una buena precisión, el algoritmo resultante se utilizará como parte de un procedimiento de selección automática para sobres. Este es un problema de clasificación para el cual la tasa de error debe mantenerse lo más baja posible de modo de evitar una mala distribución de correo. La columna 1 tiene la variable a predecir Número codificada como sigue: 0='cero'; 1='uno'; 2='dos'; 3='tres'; 4='cuatro'; 5='cinco';6='seis'; 7='siete'; 8='ocho' y 9='nueve', las demás columnas son las variables predictivas, cada fila de la tabla representa un bloque 16 × 16 por lo que la matriz tiene 256 variables predictivas.

Para este ejercicio realice lo siguiente:

1. Use el método de máquinas de soporte vectorial para generar un modelo predictivo para la tabla ZipDataTrainCod.csv usando 100 % de los datos para tabla aprendizaje. ¿Qué puede comentar con respecto al tiempo de ejecución del

algoritmo?

2. Para la tabla de prueba ZipDataTestCod.csv calcule la matriz de confusión y la precisión positiva para cada número ¿Qué tan buenos son los resultados? ¿Puede explicar por qué algunos números inducen una menor precisión?

3. Grafique la curva ROC para este modelo. ¿Qué observa?

Nota: En todos los casos, compare los resultados obtenidos con el método de máquinas de soporte vectorial con aquellos obtenidos mediante el método de árboles de decisión. En el contexto de cada problema, ¿cuál método elegiría?

Anexo

Ejercicio 1:

Cargamos la tabla de Datos,

```
Datos<-read.table('SAheart.csv', sep=";", dec=".", header=T)
#head(Datos)
```

Generamos una muestra donde el75%de los datos serán utilizados como tabla de aprendizaje y el25%como tabla de prueba (testing):

```
muestra <-sample(1:nrow(Datos),nrow(Datos)%/%(10/4))
ttesting<-Datos[muestra,]
taprendizaje<- Datos[-muestra,]
```

Cargamos las librerías necesarias para generar el modelo mediante el método de máquinas de soporte vectorial:

```
library(class)
library(e1071)
```

Generamos el modelo utilizando la función kernel lineal:

```
modelosvm<-svm(chd~.,data=taprendizaje, kernel="linear")
```

Realizamos las predicciones sobre la tabla de prueba:

```
prediccionsvm<-predict(modelosvm, ttesting)
```

Calculamos la matriz de confusión y la precisión global:

```
MCsvm<-table(ttesting[,ncol(Datos)],prediccionsvm)
MCsvm
PG<-(sum(diag(MCsvm)))/sum(MCsvm)
PG
```

El siguiente código permite construir la curva ROC sin la necesidad de recurrir a rattle:

```
library (ROCR)
modelosvm<-svm(chd~.,data=taprendizaje, kernel="linear",
probability=TRUE)
prediccionsvm<-predict(modelosvm, ttesting, probability=TRUE)

prediccionsvm.rocr <- prediction(attr(prediccionsvm,
"probabilities")[,2], ttesting$chd)
prediccionsvm.perf <- performance(prediccionsvm.rocr,"fpr",
"tpr")
plot(prediccionsvm.perf, main="Curva ROC")
```

El área bajo la curva, AUC, está dada por:

```
AUCsvm<-1-
as.numeric(slot(performance(prediccionsvm.rocr,"auc"),
"y.values"))
AUCsvm
```