**Trabajo Práctico 1**

**Aprendizaje Supervisado**

**Ejercicio1 – Parte teórica**

No copie y pegue las respuestas, sino que deben ser de elaboración propia.

Responda la pregunta asignada en Campus. Copie aquí la pregunta y la respuesta.

**Ejercicio 2 – Modelado**

En este ejercicio se pide comparar modelos de Machine Learning para predecir dígitos escritos a mano. Primero se realiza un Análisis Exploratorio de los Datos para entender la base. Luego se particiona la base en un conjunto de entrenamiento y uno de testeo, y después **con estos conjuntos se modelan dos Redes Neuronales**. Finalmente se comparan dichosmodelos.

**Parte A – Análisis Exploratorio de Datos**

1. Abra la página web “Optical Recognition of Handwritten Digits Data Set” de

UCI (Universidad de California)

http://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits

Busque el “Data Set Information / Additional Information” y cópielo aquí. “We used preprocessing…”

1. Baje el archivo zip de Download. Extraiga fuera del zip al archivo optdigits.tra Abra la base en R: base=read.table("optdigits.tra",sep=",")
2. Muestre un head(base,1) para ver cómo son las variables.

Texto

Descripción generada automáticamente

1. Cada registro es una imagen de 8x8 píxeles de números del 0 al 9 escritos a mano. En la variable V65 se encuentra qué número está en dibujado en ese registro (variable a predecir).

¿Cuántos registros tiene la base? ¿Cuántas variables? (dim(base)).

Tiene 3823 registros y 64 variables

Diagrama

Descripción generada automáticamente

1. Cada píxel tiene un tono de gris de 0 a 16. Se quiere predecir a qué número corresponde una nueva imagen.

Dibuje el registro 14 de la base como jpg que corresponde a un número “5” siguiendo las siguientes instrucciones.

library(jpeg) #cargamos la librería

vector=base[14,] #tomamos la fila 14 por separado

vector #visualizamos que la fila sigue siendo un data.frame con títulos de columnas

#visualice en la columna V65 qué número se va a dibujar

vector=vector[-65] #sacamos la columna V65 (variable a predecir – valor 5) vector=as.numeric(vector) #transformamos el data.frame a vector numérico

vector #visualizamos que la fila ahora es un vector sin títulos de columnas

vector=vector/16

#transformamos los valores de (1:16) a (0:1)

vector

imagen=array(vector,dim=c(8,8))

#creamos la imagen de 8x8

imagen=t(imagen)

#rotamos la imagen

plot(as.raster(imagen))

¿Ve que la imagen es un “5”? Muestre la imagen resultante.

Dibujo en blanco y negro

Descripción generada automáticamente con confianza baja

**Optativo:** ¿cómo invertiría los colores del número?

plot(as.raster(1 - imagen))

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

1. Considere los 2 últimos números de su DNI (2numDNI).

En el punto anterior cambie el 14 de la instrucción vector=base[14,] por el número de los 2 últimos números de su DNI.

Dibujo en blanco y negro

Descripción generada automáticamente con confianza baja

Siguiendo las instrucciones anteriores, dibuje el registro que se corresponde con ese número y muestre el resultado.

1. Vamos a hacer un preprocesamiento de la base.
2. Cada variable es un píxel de la imagen. Como la imagen es de 8x8, son 64 píxeles y el número que forman en la variable 65.

Renombre las variables predictoras.

for (i in 1:64){

var=paste("V",i,sep="")

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

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

1. Transformar la variable V65 (variable a predecir) en factor renombrándola “Numero” con las siguientes instrucciones.

names(base)[names(base)=="V65"]="Numero"

base$Numero=as.factor(base$Numero)

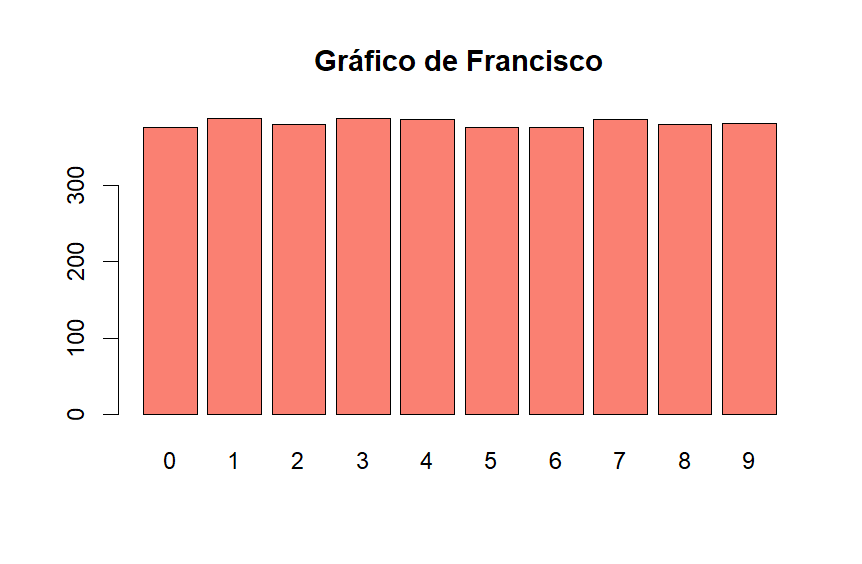
* 1. Muestre un head(base,1) para ver cómo quedaron las variables.

1. Realice un gráfico de barras de la variable a predecir. plot(base$Numero,main="Título",col="color")
   * 1. Para el título ingrese **su nombre**, como “Gráfico de Marcela” *(o sea,*

*“Gráfico de SuNombre”).*

* + 1. Elija un color para el gráfico. Tenga en cuenta que si ingresa colors() en R verá que hay +500 colores posibles.
    2. Indique el código R utilizado.

plot(base$Numero,main="Gráfico de Francisco",col="salmon")



1. Muestre una tabla con las cantidades de registros por número (o sea, cuántos registros de “1”, de “2”, etc) con summary(base$Numero)

**Optativo:** Verificar que las cantidades suman el total de registros.¿Cómo encontrar el valor mínimo y máximo de las cantidades por número?

**Parte B - Conjuntos**

1. Considere su DNI (completo) para el seteo de la semilla y particione la base en un conjunto de entrenamiento y uno de testeo con la librería caret. Además, si su DNI termina en 0, 1, 2 ó 3

Setee p=0.70

Si su DNI termina en 4, 5, 6 ó 7 Setee p=0.75

Si su DNI termina en 8 ó 9 Setee p=0.80

set.seed(DNI);particion=createDataPartition(y=base$Numero,p=asignado,list=FALSE) entreno=base[particion,]

testeo=base[-particion,]

Indique cómo quedó el código R utilizado.

1. Muestre un head y un summary del conjunto de entrenamiento y del conjunto de testeo. Dada la cantidad de columnas, mostrar los head y summary truncados a las primeras y últimas columnas.

head(entreno[,c(1:2,63:65)])

summary(entreno[,c(1:2,63:65)])

head(testeo[,c(1:2,63:65)])

summary(testeo[,c(1:2,63:65)])

1. ¿Cuántos registros quedaron en el conjunto de entrenamiento (dim(entreno)) y en el conjunto de testeo en total? ¿Cuántos registros de cada dígito quedaron en cada conjunto?

summary(entreno$Numero);summary(testeo$Numero)

**Parte C – Creación de la Red Neuronal**

1. Para el seteo de semilla considere su DNI (completo) y cree una Red Neuronal (con librería nnet) para modelar el problema planteado con maxit=20000 y cantidad de neuronas en la capa oculta size=40.

set.seed(DNI);red=nnet(Numero~.,entreno,size=40,maxit=20000,MaxNWts=20000)

El entrenamiento puede durar aproximadamente 1 minuto. Indique el código

R utilizado.

1. Muestre una captura de pantalla de la lista de iteraciones de la Red Neuronal (si son muchas, puede mostrar las primeras y las últimas iteraciones).
2. Escriba red<enter> y muestre una captura de pantalla de la información que aparece.
3. Indique la cantidad de pesos y la cantidad de iteraciones resultantes.
4. Dibuje la Red Neuronal library(NeuralNetTools) plotnet(red)
5. Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de los resultados completos (la matriz de confusión, accuracy y tablas).

pred=predict(red,testeo,type="class") confusionMatrix(factor(pred),testeo$Numero)

1. La cantidad de elementos de la matriz de confusión es igual a la cantidad de elementos de testeo (o sea dim(testeo)).

Sume la cantidad de elementos de la diagonal de la matriz de confusión y divida el resultado por dim(testeo).

Muestre la cuenta con números y muestre que es igual al accuracy.

**Optativo:** asigne la matriz de confusión a una variable. Por ejemplo cm=confusionMatrix(factor(pred),testeo$Numero)

¿Escriba names(cm). ¿Cómo mostraría solamente la matriz? ¿Cómo podría conseguir la suma de la diagonal de la matriz?

1. ¿Cuál categoría presenta mayor sensibilidad y cuál es dicho valor?
2. Con la instrucción base[num,] se puede obtener los datos de un registro de

la base #base[filas,columnas]

Considere los 2 últimos números de su DNI (2numDNI) y muestre el registro correspondiente.

numeroAsignado=base[2numDNI,]

numeroAsignado

¿Qué dígito se corresponde con ese registro?

1. Prediga el registro de 2numDNI hallado en el punto anterior. predict(red,numeroAsignado,type="class")

¿Coincide la predicción con lo esperado?