Profesora Dra Marcela Riccillo

Técnicas y Algoritmos de Aprendizaje Automático

Especialización en Ciencia de Datos ITBA

Trabajo Práctico 1 y Trabajo Práctico 2

Instrucciones:

Cada Trabajo Práctico deberá entregarse en un único archivo pdf. No se aceptarán otros formatos de archivo que no sean pdf.

En cada uno de los 2 archivos pdf deberá figurar la siguiente información:

- Carátula con Nombre del alumno y Número de DNI.
- Agregar además un encabezado o pie de página con Nombre del alumno.
- Informe con los enunciados de los ejercicios y los resultados de los análisis.
- Anexo con el código en R utilizado (copiar el código en el pdf, no enviar archivos R).

En el nombre de cada archivo pdf colocar “NombreDelAlumno TyAdeAA TP1.pdf” y “NombreDelAlumno TyAdeAA TP2.pdf”

Aclaraciones:

En el caso de agregar capturas de pantalla, las mismas tienen que estar recortadas a fin de mostrar solamente lo pedido en cada ejercicio. Por ejemplo, para indicar un gráfico se puede insertar una captura de pantalla, pero no de toda la pantalla, sino solamente de dicho gráfico.

En el caso de agregar capturas de pantalla, igualmente las preguntas deben ser respondidas en forma escrita (no se aceptarán respuestas que remitan a sectores remarcados en las imágenes). Por ejemplo, si se pide el accuracy se espera el accuracy escrito y no la frase “ver respuestas en la imagen de la matriz de confusión”.

No remitir a un anexo para ver resultados, gráficos o imágenes, completar las consignas en cada sección que corresponda.

Aunque se indique el código R utilizado en las respuestas a los ejercicios, igualmente se espera un anexo con todo el código utilizado (dentro del mismo archivo).

Trabajo Práctico 1

Aprendizaje Supervisado

Ejercicio1 – Parte teórica

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 dichos modelos.

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.

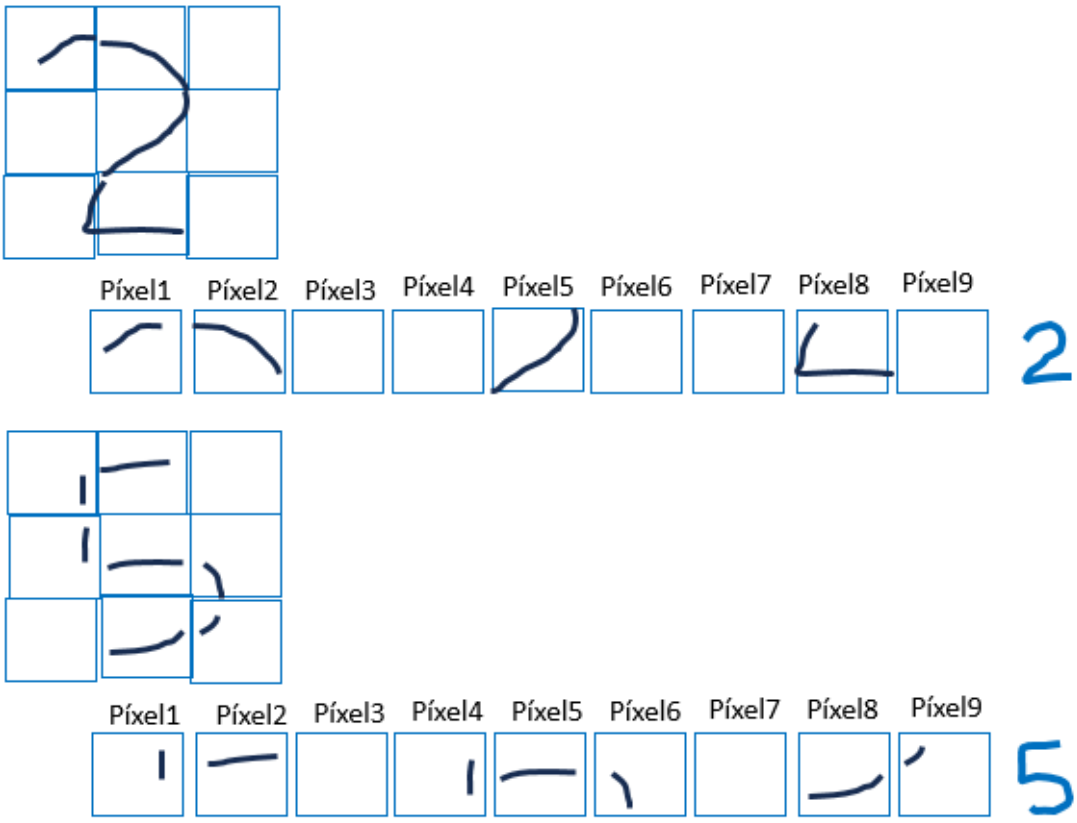
2) Baje el archivo zip de Download. Extraiga fuera del zip al archivo optdigits.tra en el directorio de trabajo de R.
Abra la base en R

```
base=read.table("optdigits.tra",sep=",")
```

3) Muestre un head(base,1) para ver cómo son las variables.

4) 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)).



5) 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.

Optativo: ¿cómo invertiría los colores del número?

6) 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.

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

7) Vamos a hacer un preprocesamiento de la base.

a) 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}
```

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

```
names(base)[names(base)=="V65"]="NumeroMostrado"
base$NumeroMostrado=as.factor(base$NumeroMostrado)
```

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

8) Realice un gráfico de barras de la variable a predecir.
plot(base$NumeroMostrado,main="Título",col="color")

a) Para el título ingrese **su nombre**, como “Gráfico de Marcela”.

b) Elija un color para el gráfico. Tenga en cuenta que si ingresa colors() en R verá que hay +500 colores posibles.

c) Indique el código R utilizado.

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

summary(base$NumeroMostrado)

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$NumeroMostrado,p=asignado,list=FALSE)
entreno=base[particion,]
testeo=base[-particion,]
```

Indique cómo quedó el código R utilizado.

2) 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[,1:5]);head(entreno[,61:65])
summary(entreno[,1:5]);summary(entreno[,61:65])

head(testeo[,1:5]);head(testeo[,61:65])
summary(testeo[,1:5]);summary(testeo[,61:65])
```

3) ¿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?

```
table(entreno$NumeroMostrado)
table(testeo$NumeroMostrado)
```

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=30.

```
set.seed(DNI);red=nnet(NumeroMostrado~.,entreno,size=30,maxit=20000,MaxNWts=20000)
```

   El entrenamiento puede durar aproximadamente 1 minuto. Indique el código R utilizado.

2) 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).

3) Escriba red<enter> y muestre una captura de pantalla de la información que aparece.

4) Indique la cantidad de pesos y la cantidad de iteraciones resultantes.

5) Dibuje la Red Neuronal
   ```
   library(NeuralNetTools)
   plotnet(red)
   ```

6) Calcule la matriz de confusión utilizando la instrucción confusionMatrix de la librería caret. Muestre una captura de pantalla de la matriz **y las tablas de sensibilidades y especificidades.**

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

7) Calcule el accuracy según la cantidad de registros bien clasificados (indique con números la fórmula que usó) y verifique que coincida con el accuracy obtenido por confusionMatrix.

   Diagonal (aciertos) / Total

   Optativo: asigne la matriz de confusión a una variable. Por ejemplo
   cm=confusionMatrix(pred,testeo$NumeroMostrado)
   ¿Escriba names(cm). ¿Cómo mostraría solamente la matriz? ¿Cómo podría conseguir la suma de la diagonal de la matriz?

8) ¿Cuál categoría presenta mayor sensibilidad y cuál es dicho valor?

9) 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.

digitoAsignado=base[2numDNI,]
digitoAsignado

¿Qué dígito se corresponde con ese registro?

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

¿Coincide la predicción con lo esperado?

Parte D – Comparación de modelos

1) Calcule nuevamente la Red Neuronal, pero con set.seed(123). Muestre una captura de pantalla de la matriz de confusión **y las tablas de sensibilidades y especificidades.** Indique el código R utilizado.

2) ¿Mejoraron los resultados de la matriz de confusión con respecto a la Red Neuronal anterior? (recordar comparar el accuracy de cada modelo, pero también las sensibilidades y especificidades por categoría). No promedie los valores de las sensibilidades y especificidades.

Trabajo Práctico 2

Aprendizaje No Supervisado

Ejercicio 1 – Segmentación de una imagen

1) Elegir una imagen pública de formato jpg. No pueden aparecer personas en la imagen y la misma debe respetar los principios éticos y de confidencialidad básicos. No utilizar imágenes usadas en clase.

   Muestre la imagen seleccionada e Indique el link de dónde fue obtenida la imagen.

   ```
   library(jpeg)
   imagen=readJPEG("archivo.jpg")
   ```

   Optativo: Si quiere, explique bervemente por qué eligió esa imagen.

2) Transformar la imagen a tonos de grises. Muestre la imagen transformada.

```
gris=(imagen[,,1]+imagen[,,2]+imagen[,,3])/3
plot(as.raster(gris))
```

optativo: writeJPEG(gris,"nombre.jpg")

3) Vuelva a la imagen original.
   Considere su DNI (completo) para setear la semilla y mediante un agrupamiento k-means segmentar la imagen en 3 colores.

```
rojo=as.vector(imagen[,,1])
verde=as.vector(imagen[,,2])
azul=as.vector(imagen[,,3])
base=data.frame(rojo,verde,azul)
set.seed(DNI);km=kmeans(base,3)
```

Cada centroide es el color promedio entre los colores de cada grupo. Muestre los centroides obtenidos con km$centers.

4) Realice y muestre un gráfico con el color de cada centroide. Fíjese que points permite agregar puntos al primer gráfico.

```
plot(10,10,pch=19,cex=10,col=rgb(km$center[1,1],km$center[1,2],km$center[1,3]))
points(11,11,pch=19,cex=10,col=rgb(km$center[2,1],km$center[2,2],km$center[2,3]))
points(12,12,pch=19,cex=10,col=rgb(km$center[3,1],km$center[3,2],km$center[3,3]))
```

5) Muestre la imagen segmentada coloreada con los 3 centroides (que son los colores del punto anterior).

```
#Reconstruir imagen
segmR=rojo
segmV=verde
segmA=azul

    segmR[km$cluster==1]=km$center[1,1]
    segmV[km$cluster==1]=km$center[1,2]
    segmA[km$cluster==1]=km$center[1,3]

    segmR[km$cluster==2]=km$center[2,1]
    segmV[km$cluster==2]=km$center[2,2]
    segmA[km$cluster==2]=km$center[2,3]

    segmR[km$cluster==3]=km$center[3,1]
    segmV[km$cluster==3]=km$center[3,2]
    segmA[km$cluster==3]=km$center[3,3]


segmentada=imagen
segmentada[,,1]=segmR
segmentada[,,2]=segmV
segmentada[,,3]=segmA
plot(as.raster(segmentada))
```

optativo: writeJPEG(segmentada,"nombre.jpg")

6) Segmente la imagen en 4 y 5 grupos coloreadas por los centroides. Muestre las 2 imágenes que quedaron.

    4 Grupos

```
rojo=as.vector(imagen[,,1])
verde=as.vector(imagen[,,2])
azul=as.vector(imagen[,,3])
base=data.frame(rojo,verde,azul)
set.seed(DNI);km=kmeans(base,4)

#Reconstruir imagen
segmR=rojo
segmV=verde
segmA=azul

    segmR[km$cluster==1]=km$center[1,1]
    segmV[km$cluster==1]=km$center[1,2]
```

```
    segmA[km$cluster==1]=km$center[1,3]

    segmR[km$cluster==2]=km$center[2,1]
    segmV[km$cluster==2]=km$center[2,2]
    segmA[km$cluster==2]=km$center[2,3]

    segmR[km$cluster==3]=km$center[3,1]
    segmV[km$cluster==3]=km$center[3,2]
    segmA[km$cluster==3]=km$center[3,3]

    segmR[km$cluster==4]=km$center[4,1]
    segmV[km$cluster==4]=km$center[4,2]
    segmA[km$cluster==4]=km$center[4,3]


segmentada=imagen
segmentada[,,1]=segmR
segmentada[,,2]=segmV
segmentada[,,3]=segmA
plot(as.raster(segmentada))
```

7) ¿Se parecen las segmentadas a la imagen original? Si es así, tenga en cuenta que la original tenía muchos colores y la segmentada solamente 3, 4 o 5 colores.

Optativo: Segmentar la imagen en más de 5 grupos o cambie los colores de un grupo.

Ejercicio 2 – Parte teórica

1) ¿Para qué podría servir segmentar imágenes?
2) ¿Para qué podría servir kmeans?