ITBA Escuela de INNOVACIÓN
EVOLUCIÓN CONTINUA

Certificación Profesional en Inteligencia Artificial ITBA

Trabajos Prácticos

Profesora Dra Marcela Riccillo

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 3 archivos pdf deberá figurar la siguiente información:

- Carátula con Nombre y Número de DNI
- Agregar además un encabezado o pie de página con el nombre.
- Texto 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). No es necesario poner resultados en el Anexo sino solamente el código.

En el nombre de cada archivo pdf colocar “Nombre IA TP1.pdf”, “Nombre IA TP2.pdf” y “Nombre IA TP3.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

Primera Parte

Ejercicio 1 – Regresión

Los casos de Regresión se caracterizan por tener una variable cuantitativa para predecir.

Seleccione un dataset con un caso de **Regresión**. El dataset debe ser obtenido de alguna librería de R o de una página web pública (no incluir datos confidenciales).

Por ejemplo, se podría utilizar:

- Datasets de R como: mtcars de base, iris de base, cheddar de faraway, etc.
- datasets de UCI (Universidad de California) https://archive.ics.uci.edu
- datasets de Kaggle https://www.kaggle.com/
- datasets de ISLR https://www.statlearning.com/resources-second-edition

El dataset debe contener al menos 3 variables y una de ellas debe ser numérica. (*Nota: este dataset es solamente para este ejercicio y no se espera ser utilizado en otros ejercicios*).

1) Indique el nombre del dataset, y la librería de R o la página web fuente del mismo.

2) ¿De qué trata la base?

3) ¿Cuántos registros tiene la base? ¿Cuántas variables? ¿De qué tipo son las variables?
   Podría utilizar dim(base) str(base) summary(base)

4) Realice un histograma de la variable numérica seleccionada. ¿En qué rango se encuentran los valores?
   hist(variable,main="Título",col="color")

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

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

   c) Indique el código R utilizado.

Segunda Parte

Ejercicio 2 – Clasificación

En este ejercicio se pide comparar modelos de Machine Learning para predecir tipos de vidrio. 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, después con estos conjuntos se modelan dos Árboles de Decisión. Finalmente se compara la performance de cada modelo.

Parte A – Análisis Exploratorio de Datos

1) Abra en R la base Glass de la librería mlbench.
   library(mlbench)
   data(Glass)

   Muestre dim(Glass) ¿cuántos ejemplos de vidrios tiene la base?

2) Escriba en R ?Glass y copie aquí la Descripción (Description) que aparece en la página web.

3) Abra la página web de Glass en UCI (Universidad de California)
   https://archive.ics.uci.edu/ml/datasets/glass+identification

   Busque la sección “Additional Variable Information” y luego Class Labels. Copie aquí los tipos de vidrio de la base.
   ¿Cuál tipo de vidrio no está representado en la base (none in this database)?

4) La variable a predecir es Type que representa los tipos de vidrios. Muestre un summary de la base summary(Glass).

5) Indique cuántos vidrios hay **de cada clase**.
   summary(Glass$Type)

6) A grandes rasgos, los tipos de vidrio serían (hay 2 tipos de ventana de edificio):

   - Ventana de edificio
   - Ventana de automóvil
   - Recipiente
   - Vajilla
   - Faro de Auto

   Busque y muestre una imagen de cada tipo de vidrio. *Indique para cada imagen la página web origen de la misma.*

7) Renombre cada categoría y transforme la variable Type a factor. Luego renombre Type como TipoDeVidrio.

```
Glass$Type=as.character(Glass$Type)
Glass$Type[Glass$Type=="1"]="VentanaTipo1"
Glass$Type[Glass$Type=="2"]="VentanaEdificio"
Glass$Type[Glass$Type=="3"]="VentanaAuto"
Glass$Type[Glass$Type=="5"]="Recipiente"
Glass$Type[Glass$Type=="6"]="Vajilla"
Glass$Type[Glass$Type=="7"]="FaroAuto"
Glass$Type=factor(Glass$Type)
names(Glass)[names(Glass)=="Type"]="TipoDeVidrio"
```

Muestre un summary(Glass) para ver cómo quedó.

8) Realice un gráfico de barras de la variable Glass$TipoDeVidrio. Amplíe el gráfico para ver los nombres de los tipos de vidrio.

```
plot(Glass$TipoDeVidrio,main="Título",col="COLOR")
```

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

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

c) Indique el código R utilizado.

Parte B – Conjuntos

1) Con la librería caret, particione la base en entrenamiento y test. Considere para el seteo de la semilla su DNI (completo).
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=Glass$TipoDeVidrio,p=asignado,list=FALSE)
entreno=Glass[particion,]
testeo=Glass[-particion,]
```

Indique cómo quedó el código R utilizado.

2) Muestre un head y un summary del conjunto de entrenamiento y del conjunto de testeo.

head(entreno)

summary(entreno)

head(testeo)

summary(testeo)

3) ¿Cuántos registros quedaron en cada conjunto (entrenamiento y testeo) en total?
dim(Glass);dim(entreno);dim(testeo)

4) ¿Cuántos registros de cada número quedaron en cada conjunto?

table(Glass$TipoDeVidrio);table(entreno$TipoDeVidrio);table(testeo$TipoDeVidrio)

Parte C – Árbol de Decisión

1) Cargue la librería rpart y cree un Árbol de Decisión para modelar el problema planteado
arbol=rpart(TipoDeVidrio~.,entreno,method="class")

Cargue la librería rpart.plot y grafique el Árbol de Decisión resultante (utilice el parámetro cex=0.8 para una mejor visualización).

rpart.plot(arbol,extra=1,type=5,cex=0.8)

2) ¿Cuántas hojas tiene el Árbol de Decisión?

3) Según el Árbol de Decisión creado, ¿cuándo un vidrio es del tipo “FaroAuto”? (Indique las reglas siguiendo las ramas desde el nodo raíz hasta las hojas “FaroAuto”).

4) 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(arbol,testeo,type="class")

confusionMatrix(pred,testeo$TipoDeVidrio)

5) 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.

6) Vea la tabla Statistics by Class debajo de la matriz de confusión e indique cuál clase presenta menor sensibilidad.

7) Con la instrucción Glass[numVidrio,] podemos obtener los datos de un vidrio de la base #base[filas,columnas]

Considere los 2 últimos números de su DNI (2numDNI) y busque el vidrio de esa fila

vidrioAsignado=Glass[2numDNI,]
vidrioAsignado

Transcriba los datos del vidrio de ese número. ¿Qué tipo de vidrio es?

8) Prediga con el Árbol de Decisión el vidrio de los 2 últimos números de su DNI
predict(arbol,vidrioAsignado,type="class")

¿Coincide la predicción con lo esperado?

Parte D – Optimización del modelo

1) Con la instrucción arbol$control se pueden ver las restricciones en la creación del Árbol de Decisión. Cree un Árbol de Decisión sin algunas de las restricciones con la siguiente instrucción.

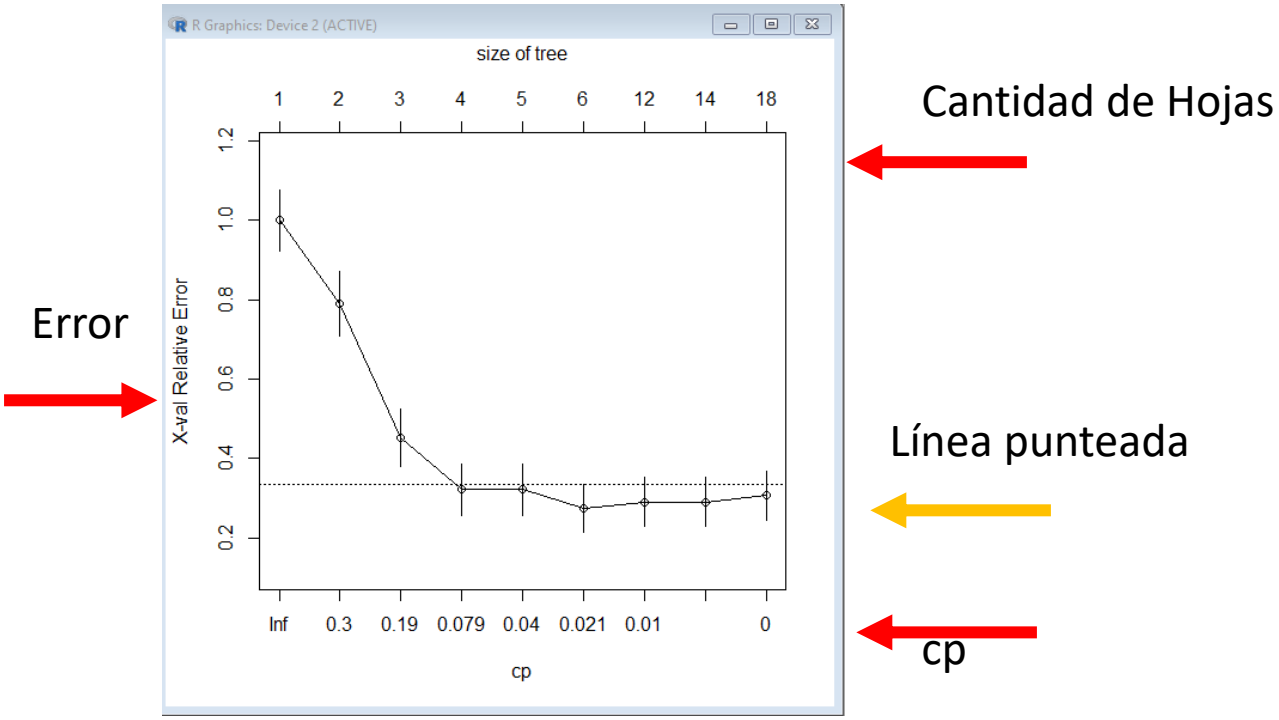
arbolGrande=rpart(TipoDeVidrio~.,entreno,method="class",cp=0,minsplit=0)

2) Dibuje el Árbol de Decisión.
rpart.plot(arbolGrande,extra=1,type=5,cex=0.6)

3) Muestre el gráfico plotcp(arbolGrande). Amplíe el gráfico para ver más cps.

Cada parte de este gráfico es la siguiente:

- En la parte inferior del gráfico figura el **cp** (parámetro de complejidad) de cada Árbol de Decisión
- En la parte superior la cantidad de **hojas**
- En el eje vertical el **error** de cada modelo (calculado por cross validation).



4) Elija un cp del gráfico, cuyo error (eje vertical) esté por debajo de la línea punteada (el cp elegido debe figurar en el gráfico). Pode el Árbol de Decisión por el cp elegido

   arbolPodado=prune(arbolGrande,cp=cpElegido)

   Indique el código R utilizado.

5) Muestre una captura de pantalla de los resultados completos (la matriz de confusión, accuracy y tablas).del Árbol de Decisión podado. ¿Mejoró el modelo?

   pred=predict(arbolPodado,testeo,type="class")
   confusionMatrix(pred,testeo$TipoDeVidrio)

   Verifique accuracy, y sensibilidades y especificidades de cada categoría. Tal vez el accuracy no mejoró, pero alguna categoría mejoró la sensibilidad.

Parte E – Parte Teórica

Resuma en un párrafo lo realizado en este TP.

Trabajo Práctico 2

Ejercicio 1 – Segmentación de una imagen

1) Elegir una imagen de formato jpg. La imagen debe ser **pública** de una página web (no de un drive). 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")
   plot(as.raster(imagen))
   ```

2) Optativo: explique brevemente por qué eligió la imagen.

3) 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")

4) 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.

5) 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]))
```

6) Muestre la imagen segmentada coloreada con los 3 centroides.

```
#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")

7) 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))
```

8) ¿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.

Ejercicio 2 – Aprendizaje No Supervisado

1) Abra la base crabs de la librería MASS

```
library(MASS)

data(crabs)
```

Escriba ?crabs y copie aquí la Descripción (Description) de la base.

Muestre la estructura de la base con str(crabs). Borre las variables cualitativas

```
crabs$index=NULL
crabs$sex=NULL
crabs$sp=NULL
```

2) Copie la información de qué significa cada variable.

3) Muestre un summary de la base summary(crabs).
4) Para el seteo de semilla considere su DNI (completo) y realice un agrupamiento kmeans.

   Además, si su DNI termina en 0, 1, 2 ó 3

   Cantidad de Grupos=3

   Si su DNI termina en 4, 5, 6 ó 7

   Cantidad de Grupos=4

   Si su DNI termina en 8 ó 9

   Cantidad de Grupos=5

   Deje el resto de los parámetros por defecto. Indique el código R utilizado.

   set.seed(DNI); km=kmeans(crabs,cantGrupos)

5) ¿Cuántos elementos quedaron en cada grupo? km$size
6) Muestre una captura de pantalla de los centroides. km$centers
7) Muestre km$cluster. ¿En qué grupo quedó el primer cangrejo de la base?
8) Determine alguna característica de alguno de los grupos con respecto a los otros grupos (tip: analice los centroides).

Trabajo Práctico 3

Ejercicio 1 – Modelado

En este ejercicio se pide comparar modelos de Machine Learning para predecir vocales (en inglés). 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, después con estos conjuntos se modela una SVM y una Red Neuronal. Finalmente se compara la performance de cada modelo.

Parte A – Análisis Exploratorio de Datos

1) Considere el archivo vowel.train de la web del libro The Elements of Statistical Learning (Hastie, Tibshirani, Friedman)

https://web.stanford.edu/~hastie/ElemStatLearn/

En la página web -> Busque Data en el menú -> Busque “Vowel: Info, Training and Test data.” -> en Test se encuentra el archivo vowel.test.

Bajar el archivo con <botón derecho> en el directorio de trabajo de R.

2) En Info se encuentra un resumen de la metodología. Busque la tabla 4.1 de la sección The Speech Data. Indique cuáles son las 11 vocales del estudio y qué palabra en inglés se da como ejemplo para su pronunciación.

3) La idea es que cuando la máquina escucha un sonido, tiene que predecir qué vocal es. ¿Para qué podría servir esto? (ayuda: piense en los asistentes de celulares que se activan por voz)

4) Abra el archivo "vowel.test" en R

base=read.table("vowel.test",sep=",",header=TRUE)

Muestre un head(base).

5) Borre la variable row.names

base$row.names=NULL

Muestre un head(base) sin la variable borrada.

6) La variable “y” es la variable a predecir que dice qué vocal está representada en cada registro. Por ejemplo, cuando dice 1 es la vocal “i”, si dice 2 es “I” y así sucesivamente.

   Renombre la variable y como “Sonido”.

names(base)[names(base)=="y"]="Sonido"

   Transforme la variable a predecir “y” en factor y los números en la vocal correspondiente.

base$Sonido=factor(base$Sonido,levels=c(1,2,3,4,5,6,7,8,9,10,11),labels=c("i", "I","E","A","a:","Y","O","C:","U","u:","3:"))

   Muestre un head de la base con head(base) para ver cómo quedaron las vocales.

7) ¿Cuántos registros tiene la base? ¿Cuántas variables? ¿De qué tipo son las variables? (ayuda: dim(base) y/o str(base))

8) Elija 2 variables (que no sea la variable Sonido). Realice un gráfico de dispersión entre las 2 variables y coloréelo por la variable “Sonido”.

   library(caret)

xyplot(varY~varX,base,groups=Sonido,auto.key=list(columns=3),main="Título",pch=num)

   a) Fíjese que en vez de auto.key=TRUE al utilizar auto.key=list(columns=3) la referencia de los colores queda ordenada en columnas.

   b) Para el título ingrese su nombre, como “Gráfico de dispersión de Marcela”.

   c) Indique el código R utilizado.

9) Con table(base$Sonido) indique cuántos ejemplos hay de cada vocal.

10) Con la instrucción base[num,] podemos ver los valores del elemento num del dataset.
    Considere los últimos 2 números de su DNI (2numDNI)
    y muestre los valores del elemento de la base en esa fila.

    vocal=base[2numDNI,]

    vocal

    ¿Qué vocal es?

Parte B - Conjuntos

1) Considere su DNI (completo) para el seteo de semilla y particione la base en un conjunto de entrenamiento y uno de testeo, utilizando la instrucción createDataPartition de 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$Sonido,p=asignado,list=FALSE)

entreno=base[particion,]

testeo=base[-particion,]

   Indique el código R utilizado.

2) Muestre un head y un summary del conjunto de entrenamiento y del conjunto de testeo.
3) Realice un table(base$Sonido) table(entreno$Sonido) y table(testeo$Sonido) ¿Cuántos registros quedaron **por vocal** en el conjunto de entrenamiento y en el de testeo?

Parte C – SVM

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.

   Para el kernel considere:
   Si su DNI termina en 0, 1, 2 ó 3
   kernel=”polynomial”
   Si su DNI termina en 4, 5, 6 ó 7
   kernel=”sigmoid”
   Si su DNI termina en 8 ó 9
   kernel=”radial”

   Indique el código R utilizado.
   svm=svm(Sonido~.,entreno,kernel="asignado")

2) Escriba svm<enter> y muestre una captura de pantalla de la información que aparece. ¿Cuántos vectores soporte aparecen?

3) 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(svm,testeo)
confusionMatrix(pred,testeo$Sonido)

4) ¿Cuál fue el accuracy?
5) ¿Cuál categoría presenta mayor sensibilidad?
6) Prediga con la SVM el registro de los 2 últimos números de su DNI.
   predict(svm,vocal)
   ¿Coincide la predicción con lo esperado?

Parte D – Red Neuronal

Nota: Para este ejercicio se utilizarán los mismos conjuntos de entrenamiento y testeo ya creados en la Parte B Conjuntos. No vuelva a particionar el dataset, sino que use los mismos entreno y testeo.

1) Considere su DNI para el seteo de semilla y cree una Red Neuronal (con librería nnet) para modelar el problema planteado con maxit=10000 y cantidad de neuronas en la capa oculta size=20.
   library(nnet)

   set.seed(DNI);red=nnet(Sonido~.,entreno,size=20,maxit=10000)

   Indique el código R utilizado.
2) Muestre una captura de pantalla de la lista de iteraciones de la Red Neuronal.
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)

   Optativo: Cambiar los colores del gráfico de la Red Neuronal.

6) 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).

pred2=predict(red,testeo,type="class")
confusionMatrix(factor(pred2),testeo$Sonido)

7) ¿Cuál fue el accuracy?

8) ¿Cuál categoría presenta mayor sensibilidad?
9) Prediga con la Red Neuronal el registro de los 2 últimos números de su DNI.
   predict(red,vocal,type="class")
   ¿Coincide la predicción con lo esperado?

Parte E - Comparación

1) Cree una tabla con el accuracy de cada modelo, y además la sensibilidad y especificidad de cada modelo por categoría. La tabla esperada no es necesario hacerla con R, sino una tabla tipo Word.

2) Compare los resultados obtenidos con la SVM y la Red Neuronal. ¿Cuál modelo le parece que resultó mejor? No promedie las sensibilidades y especificidades de cada categoría.