Metodos de regresión Big Data AEPS

Centro Internacional de Agricultura Trópical grupo de Big-Data AEPS

24/02/2015

Introducción

A continuación se muestra una breve descripción del script "REGRESSION-METHODS-AEPS-BD.R", el cual fue desarrollado en CIAT con el propósito de analizar datos de agricultura; las siguientes indicaciones muestran cómo funciona el Script y cómo puede ser usado para analizar datos con estructura similar al ejemplo. Los métodos están basados en modelos de regresión cuando la respuesta es cuantitativa; las entradas del modelo pueden ser tanto categóricas como numéricas; sin embargo es recomendable hacer una preselección de variables que valga la pena y sea razonable analizar luego. El principal objetivo del script es utilizar distintos métodos de análisis que nos permitan identificar dentro de un conjunto de variables de entrada (Inputs, predictores,...) cuales son relevantes para el explicar la variación de una variable de salida (Output, respuesta,...), también es de interés conocer el tipo de relaciones que se dan.

Los datos utilizados para el ejemplo hacen parte de una muestra de información comercial del cultivo de maíz, extraída de la plataforma AEPS; donde se procesaron 16 variables relacionadas con el cultivo y se escogió como respuesta el rendimiento comercial.

Preparación de datos

Antes de iniciar la ejecución de los modelos y el análisis exploratorio; es necesario realizar una preparación de la base de datos; es recomendable como primer paso identificar con un ID cada unidad de análisis que debe estar expresada en una fila de nuestra matriz; el paso siguiente será acomodar las variables de entrada (Que estarían representadas en las columnas de la matriz) al principio de la base de datos; el orden sugerido es: primero colocar las variables cuantitativas, luego las variables cualitativas y de último el rendimiento. Normalmente se tiene una variable clasificatoria o de segmentación que convierte los datos originales en subgrupos que luego van a ser analizados; por ejemplo variedad, departamento, municipio, etc...; En caso de NO tener alguna variable de este tipo entonces es necesario crearla, dejando como una única opción la categoría "All", tal como es el caso de la base de datos de ejemplo. Después de haber ubicado apropiadamente las columnas de nuestro conjunto de datos lo guardamos en un archivo csv (Separado por comas ","); dentro de una carpeta de trabajo donde van a ser almacenados los archivos de salida. Se recomienda utilizar un punto "." Como separador decimal. Otro aspecto relevante a tener en cuenta es que las variables cualitativas que estén escritas originalmente como numéricas; tal como serían: el año, niveles de severidad de una plaga, etc,.., deben ser convertidas previamente, para que R las reconozca como factor, esto se puede hacer simplemente agregando algo de texto a cada una de las observaciones de la misma variable. Ejemplo: año (2007 ,2007 ,2008 ,...) puede convertirse en (A2007, A2007 , A2008,...), para que R reconozca que son de tipo categórico.

```
##   ID_EVENTO pob20dias pendiente ph ferFrecQuim canFerQuim profEfectiva
## 1      1656     73000       2.0  5          11        750           20
## 2      1689     74320       3.0  5           5        700           22
## 3      1660     72250       2.0  6           7        500           25
## 4      1670     85500       2.5  6           7        500           23
## 5      1658     75000       3.0  6           8        600           23
## 6      1662     75000       2.0  5           9        650           26
##   ctrMalQui ctrPlaQui frecLabPreSul freqMonitor drenaje semillTrataEndo
## 1         4         1             1           2      Si              Si
## 2         1         0             3           1      No              No
## 3         1         1             1           1      No              Si
## 4         3         2             1           4      No              Si
## 5         6         1             1           4      No              Si
## 6         6         2             1           3      No              Si
##   colorEndosper textura materiaOrganica  materialGenetico Variety  RDT
## 1        Blanco      FA           Media            DK 234     All 5500
## 2      Amarillo     FAr            Baja            DK7088     All 7400
## 3      Amarillo      FA           Media              Otro     All 6500
## 4      Amarillo      FA           Media PIONEER 30F35 HRR     All 5200
## 5      Amarillo     FAr            Baja          PAC 105     All 5500
## 6      Amarillo     FAr            Baja PIONEER 30F35 HRR     All 6000
```

Instalación de paquetes en R

Para la ejecución de los distintos modelos y procesamiento de datos es necesario que R tenga instalado los siguientes paquetes:

```
install.packages("gtools")
install.packages("gridBase")
install.packages("relaimpo")
install.packages("caret")
install.packages("party")
install.packages("randomForest")
install.packages("snowfall")
install.packages("earth")
```

La función `install.packages` descargará los paquetes necesarios para la ejecución del Script; este paso solo se realiza la primera vez que se va a utilizar el código o cuando se cambie de versión de R.

Establecer rutas y cargar paquetes

El siguiente paso consiste en utilizar la función `requiere` la cual cargará los paquetes en R, esto se hace al ejecutar las primeras líneas del Script, después es necesario indicar la ruta donde se encuentra el archivo All-Functions-AEPS_BD.RData; este archivo lo puede descargar una vez y almacenarlo en un lugar fijo en su computador o puede mantenerlo en su carpeta de trabajo.

```
load("D:/Tobackup/Documents/Mis funciones en R/All-Functions-AEPS_BD.RData")
```

En el ejemplo he almacenado el archivo .RDATA dentro de la carpeta “Mis funciones en R” la cual se encuentra en documentos. Una vez ejecutada esta línea desde R studio, en la sección de Environment de nuestra consola debe aparecer las funciones que utilizaremos más adelante.

Después se procede a indicar la dirección de trabajo en la cual está alojada nuestra base de datos; es importante recordar que las direcciones de carpetas deben estar separada por un slach "/" normal y no un contra slach "\"

```
dirFol  <- "D:/Tobackup/Documents/HUGO_ANDRES_FILES/PRUEBA_CODIGO_INDEPENDIENTE"

setwd(dirFol)
```

Al final la función `setwd` permitirá indicar a R cual será nuestro directorio de trabajo

Lectura y estructura de la base de datos

El siguiente paso consiste en indicar el nombre del archivo que contiene la base de datos; éste debe ser escrito entre comillas después de la intrusión `dataNam <-`. Lo siguiente será identificar las columnas de los inputs, el output y la variable de partición, esto lo podemos consultar directamente de nuestra base de datos.

```
datNam  <- "BASE_PROCESADA.csv"

inputs  <- 1:16   #inputs columns
segme   <- 17      #split column
output  <- 18     #output column

dataSet   <- read.csv(datNam,row.names=1)

namsDataSet <- names(dataSet)
```

El objeto creado como `namsDataSet`, simplemente almacenará los nombres de las columnas contenidas en la base de datos.

Variable de partición del conjunto de datos

En esta parte del script se procede con la creación de una variable que contiene las particiones de nuestro conjunto de datos, tal como variedad, localidad. . . , ó lo que hayamos especificado. El proceso está configurado para captar solo las categorías que se repitan más de 30 veces, sin embargo se puede modificar. Por otro lado se recomienda no reducir este valor dado que los modelos se aproximan más a la realidad en la medida que tengan mayor cantidad de información valiosa.

```
contVariety <- table(dataSet[,segme])
variety0    <- names(sort(contVariety[contVariety>=30]))


if(length(variety0)==1){variety = variety0 }else{variety = factor(c(variety0,"All"))}
```

Dentro del proceso también se utilizará un conjunto con la totalidad de los datos el cual tendrá el nombre de `All`.

Creación de carpetas, análisis descriptivos y procesamiento de datos

La función `createFolders` creará las carpetas donde van a ser alojados los archivos resultado de los modelos, bases de datos en Excel con matrices y descriptivos y todo el resto de resultados; una vez se ejecuta este comando en nuestra pantalla aparecerá una carpeta que se llama VARIETY_ANALYSIS y aquí adentro estarán carpetas asociadas a las categorías de la variable de partición.

```
createFolders(dirFol,variety)
```

`descriptiveGraphics` permitirá un análisis descriptivo univariado de cada una de las variables involucradas; también realiza un análisis exploratorio bivariado, en el que involucra cada una de las variable de entrada versus la variable de salida; respecto a los argumentos empleados dentro de la función `(variety,dataSet, inputs, segme, output)`, estos corresponden a objetos creados anteriormente. `smooth` es una opción que permite agregar una línea de suavización en los análisis bivariados, en caso de no querer generarla simplemente se debe modificar a `F`, `ylabel` sirve para especificar la etiqueta que se utilizará en gráficos para la variable de respuesta. `ghhr` permite escoger para las variables que son numéricas, dos gráficos que pueden ser: cajas de bigotes `box` o de dispersión donde en el fondo está representando todos los puntos del conjunto total de datos `varPoints` . Es normal que en ocasiones aparezca un error al intentar colocar una línea de suavización en unas de las variables que son numéricas y discretas; no obstante la función se ejecuta, entonces hacer caso omiso y revisar las carpetas de DESCRIPTIVE_ANALYSIS. Para las variables de entrada categóricas se utilizan cajas y bigotes y gráficos de puntos.

```
descriptiveGraphics(variety,dataSet,inputs = inputs,segme = segme,
                    + output = output,smooth=T,ylabel = "Rendimiento (kg/ha)",ghrp="box")
```

La función `dataSetProces` realiza algunos procesamientos previos de las matrices como: eliminar variables con varianza igual cero, convertir datos categóricos al formato de 0 y 1 y eliminar variables correlacionadas, estos pasos son necesarios dadas algunas restricciones que colocan unos de los modelos; como resultados de ejecutar esta función encontraremos dentro de las carpetas DATASETS, la matriz de correlación, una base de datos completa, otra reducida (quitando variables correlacionadas y haciendo el proceso de variables cualitativas) y dos archivos de texto uno corScheme2 que sirve para identificar como están correlacionados los inputs y otro llamado RemovedVariables que muestra las distintas variables que fueron eliminadas en una parte del proceso.

```
dataSetProces(variety,dataSet,segme,corRed="caret")
```

Modelos de regresión

En este paso final se estiman los modelos de regresión, cada uno corresponde con una de las carpetas generadas en el paso anterior y una vez ejecutado el modelo se podrá apreciar en primera instancia un gráfico de relevancia que permitirá identificar las variables que son más representativas al explicar la variación en rendimiento; para todos se generan gráficos de dispersión donde es posible distinguir una de aproximación en las relaciones que se dan entre las variables más relevantes y la salida; los modelos que son ejecutados en `caret` (Multilayer perceptron, randomForest y cForest) son entrenados en validación cruzada, donde se extrae un 70% para hacer las pruebas y se deja por fuera un 30% para validación en 100 repeticiones. Como son procesos identicos que pueden ser ejecutados independientes, existe la posibilidad de ejecutarlos en paralelo, por tanto se puede dar indicación en muchas de las funciones sobre la cantidad de procesadores que van a ser utilizados.

MULTIPLE REGRESSION En este proceso se generan resultados de una regresión lineal multiple clásica, adicional a esto se ejecuta un modelo alternativo de selección de variables con el algoritmo backward; por el momento solo funciona cuando todas las variables de entrada son de escala cuantitativa.

```
#LINEAR REGRESSION; only when all inputs are cuantitative;
lineaRegresionFun(variety,dirLocation=paste0(getwd(),"/"),ylabs="Yield (Kg/HA)")
```

MULTILAYER PERCEPTRON En este paso se utiliza el paquete caret de R, para ejecutar modelos basado en redes neuronales. `ncores` sirve para enunciar el número de servidores que serán utilizados y pertuRelevance es para especificar que se quiere generar métricas con el método de perturbación de variables

```
#MULTILAYER PERCEPTRON
multilayerPerceptronFun(variedad,dirLocation=paste0(getwd(),"/"),ylabs="Yield (Kg/HA)",
                        + pertuRelevance=T,ncores=3)
```

RANDOM FOREST En este paso se ejecuta el método de random forest, el cual consiste en una extensión de los arboles de clasificación y regresión dentro de un proceso aleatorio en el que intervienen tanto subconjuntos de observaciones al asar, como variables al azar; le es atribuido gran flexibilidad respecto a las características los datos de entrada, sin embargo en ocasiones puede ser costoso desde el punto de vista computacional. El argumento saveWS solo aplica para c forest y random forest; este argumento permite guardar el espacio de trabajo generado de los modelos ejecutados.

```
#RANDOM FOREST
randomForestFun("All",ncores = 3,nb.it=50,saveWS=F)
```

CONDITIONAL FOREST De manera similar a random forest se ejecuta cForest el cual es una extensión de este con la modificación de que se utilizan algoritmos de árboles pero esta vez insesgados, lo cual es más imparcial respecto a variables con muchas categorías o continuas, entre otras cosas, esta función utiliza relevancias incondicionales reconociendo un conjunto de variables importantes incluso si están correlacionadas con otras. Para mayor información de este método se recomienda. http://epub.ub.uni-muenchen.de/9387/1/techreport.pdf

```
#CONDITIONAL FOREST
conditionalForestFun("All", ncores= 3,nb.it=50,saveWS=F)
```

El desarrollo de este script, se hizo en el software estadístico R versión 3.1.2, para mayor información o sugerencias puede escribir a h.a.dorado@cgiar.org

Referencias

http://topepo.github.io/caret/index.html

http://dumas.ccsd.cnrs.fr/LM-ORSAY/hal-01096237v1

Carolin Strobl, Torsten Hothorn, Achim Zeileis, Party on! A New, Conditional Variable Importance Measure for Random Forests Available in the party Package *Technical Report Number 050, 2009* (epub.ub.uni-muenchen.de/9387/1/techreport.pdf)