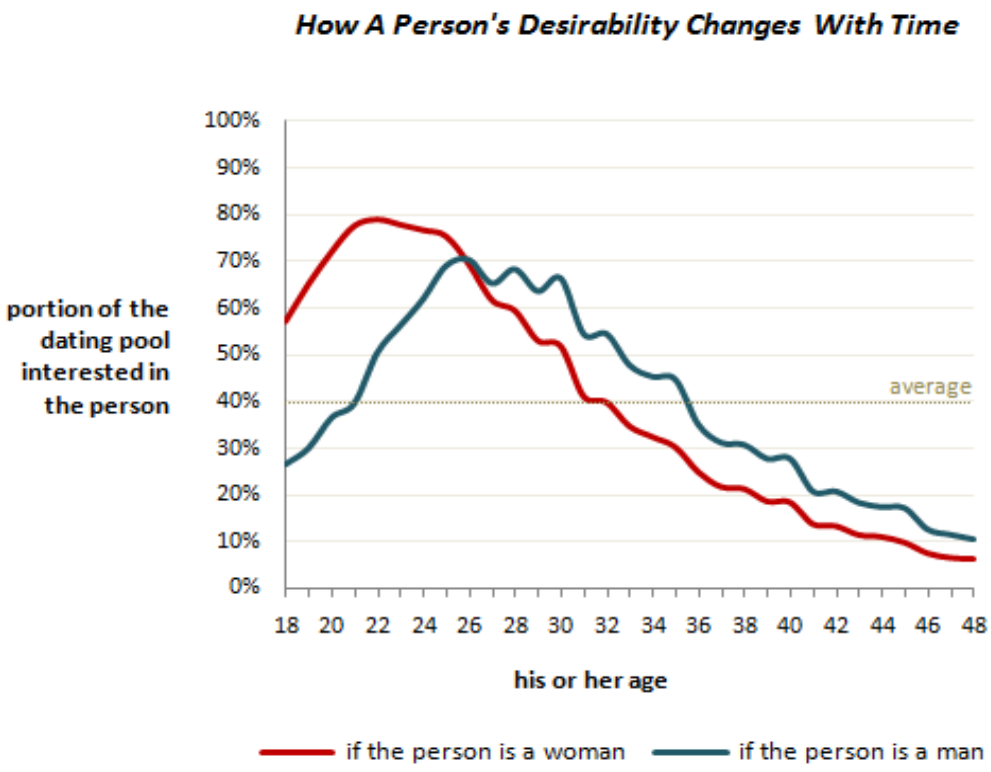
Universidad de Buenos Aires
Maestría en Data Mining
DM EyF ciclo 2019

Tareas para el Hogar

1. Lecturas ligeras para antes de dormir
   - https://medium.com/@worstonlinedater/tinder-experiments-ii-guys-unless-you-are-really-hot-you-are-probably-better-off-not-wasting-your-2ddf370a6e9a
   - https://quillette.com/2019/03/12/attraction-inequality-and-the-dating-economy
   - / https://hbr.org/2013/12/the-economics-of-online-dating

   - https://www.freecodecamp.org/news/will-the-sun-rise-tomorrow-255afc810682/
   - https://en.wikipedia.org/wiki/Additive_smoothing



2. La carpeta R del dropbox ha sido modificada, se han modificado programas para agregar "canaritos", por favor vover a cargarla

3. Tarea de Programación

   El objetivo es que el alumno construya un árbol de decision de 16 hojas, en forma semiautomatica, asistido por el codigo que se encuentra en `\R\elementary\corte_univariado_optimo.r` , cortando distinto a como corta rpart, haciendo los cortes optimos segun la funcion ganancia, y mida los resultados en training y testing.
   El alumnos debera agregar lineas al final del codigo de `corte_univariado_optimo.r`
   El alumno deberá traer dibujado ese árbol para la proxima clase y calculada la ganancia normalizada en testing.

   En el codigo se deben asignar las propias semillas aleatorias, cambiar en las lineas

```
ksemilla_azar1   <-  102191
ksemilla_azar2   <-  200177
```

   Dado que cada alumno utilizará sus propias semillas aleatorias, los arboles de los alumnos seran todos distintos, quizas corten por distintas variables y casi con certeza los cortes seran en distintos valores.

   Dado que testing es el 50%, para normalizar la ganancia que obtenga, debera multiplicarla por DOS .

En el dropbox está el nuevo script `\R\elementary\corte_univariado_optimo.r`

En dicho script está la funcion `dataset_mejorcorte` que dado un dataset devuelve el mejor corte.
El mejor corte viene dado por una variable por la que se corta, el valor por el que hay que cortar esa variable, la ganancia que se obtiene hacia la izquierda `(variable<=valor)`, la ganancia que se obtiene hacia la derecha `(variable>valor)`, y la ganancia que se obtiene cuando `is.na(variable)` es decir cuando la variable es nula.

Un ejemplo de la salida de la funcion `dataset_mejorcorte` es :

```
$`columna`
[1] "tmovimientos_ultimos90dias"

$valor
[1] "21"

$gan_left
[1] 2340500

$gan_right
[1] -35033000

$gan_na
[1] 0

$gan_total
[1] 2340500
```

Algunas consideraciones
Se elige el mejor corte siempre en training .

Se aplica el corte a todo el dataset, o sea a training y testing al mismo tiempo.

Se mide la ganancia en training y en testing, pero la que se tiene en cuenta es la de testing.

Se crea un campo `particion` en el dataset, que es la division entre training y testing. Notar que no tenemos en este caso dataset_training y dataset_testing.

1. Training es cuando particion=1
2. Testing es cuando particion=2

Se crea un campo llamado `nodo_arbol` , en donde se registra el nodo en forma jerarquica

4. nodo_arbol = “1”	
5. “11”	6. “12”

7. “111”	8. “112”	9. “121”	10. “122”

La siguiente instruccion, que es una sola, hace los conteos necesarios :

```
dataset[ , list(  train_cant = sum(  particion==1 ),
                  test_cant  = sum(  particion==2 ),
                  train_pos  = sum(  particion==1 & clase01==1 ),
                  test_pos   = sum(  particion==2 & clase01==1 ),
                  train_neg  = sum(  particion==1 & clase01==0 ),
                  test_neg   = sum(  particion==2 & clase01==0 ),
                  train_gan  = sum(  ifelse( particion==1,
ifelse( clase01, 19500, -500),  NA ), na.rm=TRUE),
                  test_gan   = sum(  ifelse( particion==2,
ifelse( clase01, 19500, -500),  NA ), na.rm=TRUE)
                )
         , by="nodo_arbol" ]
```

esta instruccion debe ser llamada cada vez que se quiera contabilizar
Tener en cuenta que para calcular la ganancia en testing de un arbol, se debe sumar solamente la ganancia de las hojas en testing pero unicamente cuando dicha hoja en training tiene ganancia positiva. O sea, es posible sumar una hoja con ganancia negativa en testing porque en training dicha hoja da ganancia positiva.

Debido a que los alumnos van a usar sus propias semillas aleatorias, DEBEN reescribir la parte final de codigo, y luego continuar cortando a mano los nodos.