Memoria Competición Kaggel Preprocesamiento

Francisco Pérez Hernández

09/03/2017

Contents

1 Introducción al problema y a Kaggel **3**
1.1 Lectura del dataset accidentes . . . 3
1.2 Primera prueba con un modelo . . . 5
1.3 Creación del archivo de salida y subida a kaggel . . . 6

2 Análisis del dataset **7**
2.1 Eliminación de valores perdidos . . . 7
2.2 Prueba del modelo con eliminación de variables . . . 10

3 Vusualización del dataset **11**
3.1 Análisis de las variables actuales . . . 11
3.2 Análisis de variables eliminadas sin valores perdidos . . . 15

4 Visión preeliminar de los datos **18**

5 Imputación de valores perdidos **19**
5.1 Imputación de variables . . . 19
5.2 Prueba del modelo con imputación de valores perdidos . . . 21

6 Detección de anomalias **21**
6.1 Uso del paquete outliers . . . 21
6.2 Paquete mvoutlier . . . 24
6.3 Eliminación de valores anómalos . . . 24
6.4 Prueba del modelo con imputación de valores perdidos . . . 27

7 Transformación de los datos **28**
7.1 Transformando los datos . . . 28
7.2 Prueba del modelo con transformación de los datos . . . 28

8 Discretización **28**

9 Selección de características **28**
9.1 Paquete FSelector . . . 29
9.2 Paquete caret . . . 36
9.3 Paquete Boruta . . . 39

10 Detección de ruido **47**

11 Modelo SVM Radial **49**

12 SVM (Support Vector Machine) **51**

13 Métodos ensamble de construcción de conjuntos de modelos **51**
13.1 Bagging . . . 51
13.2 Random forest . . . 52
13.3 Boosting . . . 55

14 Arboles de clasificación **56**

15 Primeras Conclusiones **57**

16 Prueba de modelos **58**
16.1 Prueba del primer modelo random forest 1000 58
16.2 Prueba del segundo modelo random forest 1000 65
16.3 Prueba del tercer modelo random forest 1000 67
16.4 Prueba del cuarto modelo random forest 1000 69
16.5 Prueba del quinto modelo random forest 1000 71
16.6 Prueba del sexto modelo random forest 1000 73
16.7 Prueba del primer modelo random forest multihebra 75
16.8 Prueba del primer modelo random forest 5000 76
16.9 Prueba del segundo modelo random forest multihebra 78
16.10Prueba del tercer modelo random forest multihebra 80
16.11Prueba del cuarto modelo random forest multihebra 81
16.12Prueba del quinto modelo random forest multihebra 83
16.13Prueba del sexto modelo random forest multihebra 83
16.14Prueba del setimo modelo random forest multihebra 85
16.15Prueba del octavo modelo random forest multihebra 87
16.16Prueba del noveno modelo random forest multihebra 88
16.17Prueba del decimo modelo random forest multihebra 88
16.18Prueba del modelo 11 de random forest multihebra 90

17 Balanceo de los datos **91**

18 Gráfica de resultados **94**

1 Introducción al problema y a Kaggel

Lo primero que se pretende realizar en este apartado es leer el dataset que nos han dado y realizar una subida a la plataforma Kaggel para obtener una primera puntuación. Mi usuario en Kaggel es "PacoPollos".

1.1 Lectura del dataset accidentes

Vamos a leer tanto los archivos de train como test dados.

```
accidentes.train.original <- read.csv("accidentes-kaggle.csv")
accidentes.test.original <- read.csv("accidentes-kaggle-test.csv")
```

Una vez leídos vamos a realizar un summary para ver como están compuestos los datos.

```
summary(accidentes.train.original)
```

```
##       ANIO              MES             HORA           DIASEMANA
##  Min.   :2008   Julio     : 2757   14     : 1965   DOMINGO  :3597
##  1st Qu.:2009   Junio     : 2649   19     : 1847   JUEVES   :4351
##  Median :2010   Mayo      : 2605   13     : 1823   LUNES    :4349
##  Mean   :2010   Octubre   : 2600   17     : 1749   MARTES   :4343
##  3rd Qu.:2012   Septiembre: 2491   18     : 1726   MIERCOLES:4394
##  Max.   :2013   Diciembre : 2448   12     : 1713   SABADO   :4000
##                 (Other)   :14452   (Other):19179   VIERNES  :4968
##      PROVINCIA                COMUNIDAD_AUTONOMA           ISLA
##  Barcelona: 6238   Cataluna             :8208     NO_ES_ISLA  :28476
##  Madrid   : 4735   Madrid, Comunidad de:4735     MALLORCA    :  608
##  Valencia : 1658   Andalucia            :4412     TENERIFE    :  436
##  Sevilla  :  977   Comunitat Valenciana:2653     GRAN CANARIA:  199
##  Cadiz    :  887   Pais Vasco           :1594     IBIZA       :  117
##  Girona   :  814   Castilla y Leon      :1505     LANZAROTE   :   53
##  (Other)  :14693   (Other)              :6895     (Other)     :  113
##   TOT_VICTIMAS     TOT_MUERTOS      TOT_HERIDOS_GRAVES TOT_HERIDOS_LEVES
##  Min.   : 1.000   Min.   :0.00000   Min.   :0.0000     Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.:0.00000   1st Qu.:0.0000     1st Qu.: 1.00
##  Median : 1.000   Median :0.00000   Median :0.0000     Median : 1.00
##  Mean   : 1.429   Mean   :0.02447   Mean   :0.1453     Mean   : 1.26
##  3rd Qu.: 2.000   3rd Qu.:0.00000   3rd Qu.:0.0000     3rd Qu.: 1.00
##  Max.   :19.000   Max.   :7.00000   Max.   :9.0000     Max.   :18.00
##
##  TOT_VEHICULOS_IMPLICADOS          ZONA                 ZONA_AGRUPADA
##  Min.   : 1.000           CARRETERA  :13278   VIAS INTERURBANAS:13335
##  1st Qu.: 1.000           TRAVESIA   :  241   VIAS URBANAS     :16667
##  Median : 2.000           VARIANTE   :   57
##  Mean   : 1.738           ZONA URBANA:16426
##  3rd Qu.: 2.000
##  Max.   :21.000
##
##    CARRETERA
##  A-7    :  294
##  A-2    :  278
##  AP-7   :  229
##  N-340  :  229
##  A-4    :  184
```

```
##  (Other):12098
##  NA's   :16690
##                                                RED_CARRETERA
##  OTRAS TITULARIDADES                                  :  318
##  TITULARIDAD AUTONOMICA                               : 3890
##  TITULARIDAD ESTATAL                                  : 4021
##  TITULARIDAD MUNICIPAL                                :19077
##  TITULARIDAD PROVINCIAL (DIPUTACION, CABILDO O CONSELL): 2696
##
##
##             TIPO_VIA
##  OTRO TIPO       :15527
##  VIA CONVENCIONAL:10044
##  AUTOVIA         : 2941
##  AUTOPISTA       :  723
##  CAMINO VECINAL  :  519
##  RAMAL DE ENLACE :  101
##  (Other)         :  147
##                                       TRAZADO_NO_INTERSEC
##  CURVA FUERTE CON MARCA Y SIN VELOCIDAD MARCADA:  559
##  CURVA FUERTE CON MARCA Y VELOCIDAD MARCADA    :  872
##  CURVA FUERTE SIN MARCAR                       :  481
##  CURVA SUAVE                                   : 2875
##  ES_INTERSECCION                               :11038
##  RECTA                                         :14177
##
##            TIPO_INTERSEC
##  EN T O Y          : 3350
##  EN X O +          : 4714
##  ENLACE DE ENTRADA :  421
##  ENLACE DE SALIDA  :  223
##  GIRATORIA         : 2006
##  NO_ES_INTERSECCION:18983
##  OTROS             :  305
##                                                 ACOND_CALZADA
##  CARRIL CENTRAL DE ESPERA                              :  193
##  NADA ESPECIAL                                         : 4645
##  OTRO TIPO                                             :  791
##  PASO PARA PEATONES O ISLETAS EN CENTRO DE VIA PRINCIPAL:  397
##  RAQUETA DE GIRO IZQUIERDA                             :  109
##  SOLO ISLETAS O PASO PARA PEATONES                     :  168
##  NA's                                                  :23699
##                  PRIORIDAD           SUPERFICIE_CALZADA
##  NINGUNA (SOLO NORMA) :13495   SECA Y LIMPIA  :25236
##  SEMAFORO             : 1778   MOJADA         : 3895
##  SEÆAL DE STOP        : 1750   OTRO TIPO      :  327
##  SOLO MARCAS VIALES   : 1659   UMBRIA         :  165
##  SEÆAL DE CEDA EL PASO: 1629   GRAVILLA SUELTA:  150
##  (Other)              : 1569   ACEITE         :   83
##  NA's                 : 8122   (Other)        :  146
##                           LUMINOSIDAD      FACTORES_ATMOSFERICOS
##  CREPUSCULO                     : 1330   BUEN TIEMPO  :25852
##  NOCHE: ILUMINACION INSUFICIENTE: 1067   LLOVIZNANDO  : 2524
##  NOCHE: ILUMINACION SUFICIENTE  : 4793   OTRO         :  715
```

```
##  NOCHE: SIN ILUMINACION         : 1815   LLUVIA FUERTE:  499
##  PLENO DIA                      :20997   VIENTO FUERTE:  156
##                                          NIEBLA LIGERA:   83
##                                          (Other)      :  173
##               VISIBILIDAD_RESTRINGIDA         OTRA_CIRCUNSTANCIA
##  SIN RESTRICCION          :16982      NINGUNA          :24967
##  CONFIGURACION DEL TERRENO:  989      OTRA             :  942
##  OTRA_CAUSA               :  491      OBRAS            :  263
##  FACTORES ATMOSFERICOS    :  374      FUERTE DESCENSO  :  227
##  EDIFICIOS                :  229      CAMBIO DE RASANTE:  100
##  (Other)                  :  252      (Other)          :  264
##  NA's                     :10685      NA's             : 3239
##           ACERAS         DENSIDAD_CIRCULACION          MEDIDAS_ESPECIALES
##  NO HAY ACERA:21416   CONGESTIONADA:  308     CARRIL REVERSIBLE :   17
##  SI HAY ACERA: 5437   DENSA        : 1479     HABILITACION ARCEN:    8
##  NA's        : 3149   FLUIDA       :17505     NINGUNA MEDIDA    :21024
##                       NA's         :10710     OTRA MEDIDA       :  278
##                                               NA's              : 8675
##
##
##             TIPO_ACCIDENTE
##  Atropello         : 3642
##  Colision_Obstaculo:  952
##  Colision_Vehiculos:16520
##  Otro              : 1807
##  Salida_Via        : 6013
##  Vuelco            : 1068
##
```

Vemos como las variables TTO_VICTIMAS, TOT_MUERTOS, TOT_HERIDOS_GRAVES, TOT_HERIDOS_LEVES y TOT_VEHICULOS_IMPLICADOS son las únicas variables numéricas, por lo que nos quedaremos con ellas para la primera prueba, junto con la variable clasificadora TIPO_ACCIDENTE.

```
accidentes.train.solo.numericos <- accidentes.train.original[,c(8,9,10,11,12,30)]
accidentes.test.solo.numericos <- accidentes.test.original[,c(8,9,10,11,12)]
```

1.2 Primera prueba con un modelo

Lo primero es, con las variables numéricas únicamente, voy a realizar un primer modelo, que será un árbol, para predecir la clase del conjunto de test y comprobar el funcionamiento de Kaggel al no tener experiencia anterior.

```
set.seed(1234)
ct1 <- ctree(TIPO_ACCIDENTE ~., accidentes.train.solo.numericos)
testPred1 <- predict(ct1, newdata = accidentes.test.solo.numericos)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
ct1
```

```
##
##   Conditional inference tree with 14 terminal nodes
##
## Response:  TIPO_ACCIDENTE
## Inputs:  TOT_VICTIMAS, TOT_MUERTOS, TOT_HERIDOS_GRAVES, TOT_HERIDOS_LEVES, TOT_VEHICULOS_IMPLICADOS
```

```
## Number of observations:  30002
##
## 1) TOT_VEHICULOS_IMPLICADOS <= 1; criterion = 1, statistic = 14488.658
##   2) TOT_VICTIMAS <= 1; criterion = 1, statistic = 329.362
##     3) TOT_HERIDOS_GRAVES <= 0; criterion = 1, statistic = 38.228
##       4) TOT_HERIDOS_LEVES <= 0; criterion = 0.996, statistic = 21.181
##         5)*  weights = 256
##       4) TOT_HERIDOS_LEVES > 0
##         6)*  weights = 7696
##     3) TOT_HERIDOS_GRAVES > 0
##       7)*  weights = 1476
##   2) TOT_VICTIMAS > 1
##     8) TOT_VICTIMAS <= 2; criterion = 1, statistic = 47.735
##       9)*  weights = 1605
##     8) TOT_VICTIMAS > 2
##       10)*  weights = 550
## 1) TOT_VEHICULOS_IMPLICADOS > 1
##   11) TOT_HERIDOS_LEVES <= 1; criterion = 1, statistic = 99.886
##     12) TOT_HERIDOS_LEVES <= 0; criterion = 1, statistic = 49.242
##       13)*  weights = 1276
##     12) TOT_HERIDOS_LEVES > 0
##       14) TOT_VICTIMAS <= 1; criterion = 1, statistic = 34.382
##         15) TOT_VEHICULOS_IMPLICADOS <= 3; criterion = 1, statistic = 28.319
##           16) TOT_VEHICULOS_IMPLICADOS <= 2; criterion = 0.999, statistic = 24.207
##             17)*  weights = 10133
##           16) TOT_VEHICULOS_IMPLICADOS > 2
##             18)*  weights = 924
##         15) TOT_VEHICULOS_IMPLICADOS > 3
##           19)*  weights = 254
##       14) TOT_VICTIMAS > 1
##         20) TOT_VEHICULOS_IMPLICADOS <= 3; criterion = 0.965, statistic = 15.891
##           21)*  weights = 370
##         20) TOT_VEHICULOS_IMPLICADOS > 3
##           22)*  weights = 21
##   11) TOT_HERIDOS_LEVES > 1
##     23) TOT_VEHICULOS_IMPLICADOS <= 4; criterion = 0.994, statistic = 20.095
##       24) TOT_VEHICULOS_IMPLICADOS <= 2; criterion = 0.998, statistic = 22.592
##         25)*  weights = 4183
##       24) TOT_VEHICULOS_IMPLICADOS > 2
##         26)*  weights = 1124
##     23) TOT_VEHICULOS_IMPLICADOS > 4
##       27)*  weights = 134
```

1.3 Creación del archivo de salida y subida a kaggel

Vamos a escribir la salida del primer modelo para ver su puntuación en Kaggel.

```
salida.primer.modelo <- as.matrix(testPred1)
salida.primer.modelo <- cbind(c(1:(dim(salida.primer.modelo)[1])), salida.primer.modelo)
colnames(salida.primer.modelo) <- c("Id","Prediction")
write.table(salida.primer.modelo,file="predicciones/PrimeraPrediccion.txt",sep=",",quote = F,row.names =
```

1 Por lo que ya tenemos un fichero con la salida del conjunto de test. Lo único que tendremos que modificar es la primera linea del archivo para añadir "Id, Prediction". El resultado de este primer modelo para la

competición de Kaggel, subido el 11/02/2017 a las 19:54, con un total de 5 personas entregadas, se ha quedado en la posición 3 con una puntuación del 0.73246.

2 Análisis del dataset

Una vez realizada la primera prueba en Kaggel, vamos a analizar con detalle el dataset que nos han dado.

2.1 Eliminación de valores perdidos

Anteriormente en el summary, hemos visto que hay variables con valores perdidos, ya que por ejemplo, en la variable CARRETERA uno de los valores que más se repite es NA's. Por lo tanto, vamos a analizar que variables contienen valores perdidos.

```
porcentaje.de.valores.perdidos.por.columna.train <- apply(accidentes.train.original,2,function(x) sum(is
columnas.train.con.valores.perdidios <- (porcentaje.de.valores.perdidos.por.columna.train > 0)
columnas.train.con.valores.perdidios
```

```
##                     ANIO                      MES                     HORA
##                    FALSE                    FALSE                    FALSE
##                DIASEMANA                PROVINCIA       COMUNIDAD_AUTONOMA
##                    FALSE                    FALSE                    FALSE
##                     ISLA             TOT_VICTIMAS              TOT_MUERTOS
##                    FALSE                    FALSE                    FALSE
##       TOT_HERIDOS_GRAVES       TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                    FALSE                    FALSE                    FALSE
##                     ZONA             ZONA_AGRUPADA                CARRETERA
##                    FALSE                    FALSE                     TRUE
##            RED_CARRETERA                 TIPO_VIA      TRAZADO_NO_INTERSEC
##                    FALSE                    FALSE                    FALSE
##            TIPO_INTERSEC            ACOND_CALZADA                PRIORIDAD
##                    FALSE                     TRUE                     TRUE
##       SUPERFICIE_CALZADA              LUMINOSIDAD    FACTORES_ATMOSFERICOS
##                    FALSE                    FALSE                    FALSE
##  VISIBILIDAD_RESTRINGIDA       OTRA_CIRCUNSTANCIA                   ACERAS
##                     TRUE                     TRUE                     TRUE
##     DENSIDAD_CIRCULACION       MEDIDAS_ESPECIALES           TIPO_ACCIDENTE
##                     TRUE                     TRUE                    FALSE
```

Por lo que tenemos que las variables con valores perdidos son: CARRETERA, ACOND_CALZADA, PRIORIDAD, VISIBILIDAD_RESTRINGIDA, OTRA_CIRCUNSTANCIA, ACERAS, DENSIDAD_CIRCULACION y MEDIDAS_ESPECIALES. Veamos el resumen para esas variables.

```
summary(accidentes.train.original[c("CARRETERA","ACOND_CALZADA","PRIORIDAD", "VISIBILIDAD_RESTRINGIDA",
```

```
##    CARRETERA
##  A-7    :  294
##  A-2    :  278
##  AP-7   :  229
##  N-340  :  229
##  A-4    :  184
##  (Other):12098
##  NA's   :16690
##                                          ACOND_CALZADA
```

```
##  CARRIL CENTRAL DE ESPERA                                :  193
##  NADA ESPECIAL                                           : 4645
##  OTRO TIPO                                               :  791
##  PASO PARA PEATONES O ISLETAS EN CENTRO DE VIA PRINCIPAL:  397
##  RAQUETA DE GIRO IZQUIERDA                               :  109
##  SOLO ISLETAS O PASO PARA PEATONES                       :  168
##  NA's                                                    :23699
##                  PRIORIDAD                  VISIBILIDAD_RESTRINGIDA
##  NINGUNA (SOLO NORMA) :13495   SIN RESTRICCION          :16982
##  SEMAFORO             : 1778   CONFIGURACION DEL TERRENO:  989
##  SEÆAL DE STOP        : 1750   OTRA_CAUSA               :  491
##  SOLO MARCAS VIALES   : 1659   FACTORES ATMOSFERICOS    :  374
##  SEÆAL DE CEDA EL PASO: 1629   EDIFICIOS                :  229
##  (Other)              : 1569   (Other)                  :  252
##  NA's                 : 8122   NA's                     :10685
##          OTRA_CIRCUNSTANCIA          ACERAS         DENSIDAD_CIRCULACION
##  NINGUNA          :24967    NO HAY ACERA:21416   CONGESTIONADA:  308
##  OTRA             :  942    SI HAY ACERA: 5437   DENSA        : 1479
##  OBRAS            :  263    NA's        : 3149   FLUIDA       :17505
##  FUERTE DESCENSO  :  227                         NA's         :10710
##  CAMBIO DE RASANTE:  100
##  (Other)          :  264
##  NA's             : 3239
##           MEDIDAS_ESPECIALES
##  CARRIL REVERSIBLE :   17
##  HABILITACION ARCEN:    8
##  NINGUNA MEDIDA    :21024
##  OTRA MEDIDA       :  278
##  NA's              : 8675
##
##
```

Donde podemos ver que el valor más pequeño de NA's es para la variable ACERAS con 3149 instancias con valores perdidos, lo que sería un 10,49% de los datos. Un 25% de los datos de este train serían unas 7500 instancias, por lo que las variables que tienen más del 25% de valores perdidos son: CARRETERA, ACOND_CALZADA, PRIORIDAD, VISIBILIDAD_RESTRINGIDA, DENSIDAD_CIRCULACION y MEDIDAS_ESPECIALES. O lo que es lo mismo, me quedo con las variables OTRA_CIRCUNSTANCIA y ACERAS, del anterior grupo. Pero además voy a comenzar eliminando esas variables ya que a mi juicio pueden no tener demasiada importancia.

```
primeras.variables.eliminadas <- c("CARRETERA","ACOND_CALZADA","PRIORIDAD", "VISIBILIDAD_RESTRINGIDA",
accidentes.train.sin.variables.1 <- accidentes.train.original[,-c(15,20,21,25,26,27,28,29)]
accidentes.train.variables.eliminadas <- accidentes.train.original[,c(15,20,21,25,26,27,28,29)]
```

Por lo que guardo en una variable las variables que he eliminado, y creo mi dataset sin variables con valores NA. Hago lo mismo para el test:

```
accidentes.test.sin.variables.1 <- accidentes.test.original[,-c(15,20,21,25,26,27,28,29)]
accidentes.test.variables.eliminadas <- accidentes.test.original[,c(15,20,21,25,26,27,28,29)]
accidentes.test.variables.eliminadas.copia <- accidentes.test.variables.eliminadas
```

Pensemos ahora que variables restantes pueden ser no interesantes.

```
summary(accidentes.train.sin.variables.1)
```

```
##       ANIO              MES             HORA           DIASEMANA
##  Min.   :2008   Julio     : 2757   14     : 1965   DOMINGO  :3597
```

```
##  1st Qu.:2009   Junio     : 2649   19     : 1847   JUEVES   :4351
##  Median :2010   Mayo      : 2605   13     : 1823   LUNES    :4349
##  Mean   :2010   Octubre   : 2600   17     : 1749   MARTES   :4343
##  3rd Qu.:2012   Septiembre: 2491   18     : 1726   MIERCOLES:4394
##  Max.   :2013   Diciembre : 2448   12     : 1713   SABADO   :4000
##                 (Other)   :14452   (Other):19179   VIERNES  :4968
##      PROVINCIA                COMUNIDAD_AUTONOMA           ISLA
##  Barcelona: 6238   Cataluna            :8208     NO_ES_ISLA  :28476
##  Madrid   : 4735   Madrid, Comunidad de:4735     MALLORCA    :  608
##  Valencia : 1658   Andalucia           :4412     TENERIFE    :  436
##  Sevilla  :  977   Comunitat Valenciana:2653     GRAN CANARIA:  199
##  Cadiz    :  887   Pais Vasco          :1594     IBIZA       :  117
##  Girona   :  814   Castilla y Leon     :1505     LANZAROTE   :   53
##  (Other)  :14693   (Other)             :6895     (Other)     :  113
##   TOT_VICTIMAS     TOT_MUERTOS      TOT_HERIDOS_GRAVES TOT_HERIDOS_LEVES
##  Min.   : 1.000   Min.   :0.00000   Min.   :0.0000     Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.:0.00000   1st Qu.:0.0000     1st Qu.: 1.00
##  Median : 1.000   Median :0.00000   Median :0.0000     Median : 1.00
##  Mean   : 1.429   Mean   :0.02447   Mean   :0.1453     Mean   : 1.26
##  3rd Qu.: 2.000   3rd Qu.:0.00000   3rd Qu.:0.0000     3rd Qu.: 1.00
##  Max.   :19.000   Max.   :7.00000   Max.   :9.0000     Max.   :18.00
##
##  TOT_VEHICULOS_IMPLICADOS          ZONA                 ZONA_AGRUPADA
##  Min.   : 1.000           CARRETERA  :13278   VIAS INTERURBANAS:13335
##  1st Qu.: 1.000           TRAVESIA   :  241   VIAS URBANAS     :16667
##  Median : 2.000           VARIANTE   :   57
##  Mean   : 1.738           ZONA URBANA:16426
##  3rd Qu.: 2.000
##  Max.   :21.000
##
##                                                 RED_CARRETERA
##  OTRAS TITULARIDADES                                   :  318
##  TITULARIDAD AUTONOMICA                                : 3890
##  TITULARIDAD ESTATAL                                   : 4021
##  TITULARIDAD MUNICIPAL                                 :19077
##  TITULARIDAD PROVINCIAL (DIPUTACION, CABILDO O CONSELL): 2696
##
##
##              TIPO_VIA
##  OTRO TIPO       :15527
##  VIA CONVENCIONAL:10044
##  AUTOVIA         : 2941
##  AUTOPISTA       :  723
##  CAMINO VECINAL  :  519
##  RAMAL DE ENLACE :  101
##  (Other)         :  147
##                                       TRAZADO_NO_INTERSEC
##  CURVA FUERTE CON MARCA Y SIN VELOCIDAD MARCADA:  559
##  CURVA FUERTE CON MARCA Y VELOCIDAD MARCADA    :  872
##  CURVA FUERTE SIN MARCAR                       :  481
##  CURVA SUAVE                                   : 2875
##  ES_INTERSECCION                               :11038
##  RECTA                                         :14177
##
```

```
##             TIPO_INTERSEC         SUPERFICIE_CALZADA
##  EN T O Y          : 3350   SECA Y LIMPIA  :25236
##  EN X O +          : 4714   MOJADA         : 3895
##  ENLACE DE ENTRADA :  421   OTRO TIPO      :  327
##  ENLACE DE SALIDA  :  223   UMBRIA         :  165
##  GIRATORIA         : 2006   GRAVILLA SUELTA:  150
##  NO_ES_INTERSECCION:18983   ACEITE         :   83
##  OTROS             :  305   (Other)        :  146
##                           LUMINOSIDAD      FACTORES_ATMOSFERICOS
##  CREPUSCULO                     : 1330   BUEN TIEMPO  :25852
##  NOCHE: ILUMINACION INSUFICIENTE: 1067   LLOVIZNANDO  : 2524
##  NOCHE: ILUMINACION SUFICIENTE  : 4793   OTRO         :  715
##  NOCHE: SIN ILUMINACION         : 1815   LLUVIA FUERTE:  499
##  PLENO DIA                      :20997   VIENTO FUERTE:  156
##                                          NIEBLA LIGERA:   83
##                                          (Other)      :  173
##             TIPO_ACCIDENTE
##  Atropello         : 3642
##  Colision_Obstaculo:  952
##  Colision_Vehiculos:16520
##  Otro              : 1807
##  Salida_Via        : 6013
##  Vuelco            : 1068
##
```

Podemos pensar que otras de las variables que puede que no nos sean de mucha utilidad pueden ser: ANIO, MES, HORA, DIASEMANA, PROVINCIA, COMUNIDAD_AUTONOMA, ISLA, ZONA_AGRUPADA, TIPO_VIA, TRAZADO_NO_INTERSEC, TIPO_INTERSEC, SUPERFICIE_CALZADA y LUMINOSIDAD. Ya que muchas de estas variables podrían no ser de vital importancia, de primera mano, para la obtención de la predicción del tipo de accidente. Por lo tanto, vamos a eliminarlas de momento para agilizar los modelos primeros.

```
segundas.variables.eliminadas <- c("ANIO", "MES", "HORA", "DIASEMANA", "PROVINCIA", "COMUNIDAD_AUTONOMA
accidentes.train.sin.variables.2 <- accidentes.train.sin.variables.1[,-c(1,2,3,4,5,6,7,14,16,17,18,19,2
accidentes.train.variables.eliminadasa <- cbind(accidentes.train.variables.eliminadas ,accidentes.train
accidentes.test.sin.variables.2 <- accidentes.test.sin.variables.1[,-c(1,2,3,4,5,6,7,14,16,17,18,19,20)]
accidentes.test.variables.eliminadas <- cbind(accidentes.test.sin.variables.2 ,accidentes.test.original
```

2.2 Prueba del modelo con eliminación de variables

Hagamos por lo tanto una prueba de como afecta la inclusión de estas variables con respecto a la primera prueba realizada.

```
set.seed(1234)
ct2 <- ctree(TIPO_ACCIDENTE ~., accidentes.train.sin.variables.2)
testPred2 <- predict(ct2, newdata = accidentes.test.sin.variables.2)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct2
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.segundo.modelo <- as.matrix(testPred2)
salida.segundo.modelo <- cbind(c(1:(dim(salida.segundo.modelo)[1])), salida.segundo.modelo)
```

```
colnames(salida.segundo.modelo) <- c("Id","Prediction")
write.table(salida.segundo.modelo,file="predicciones/SegundaPrediccion.txt",sep=",",quote = F,row.names
```

2 El resultado de este modelo para la competición de Kaggel, subido el 17/02/2017 a las 17:51, con un total de 14 personas entregadas, se ha quedado en la posición 9 con una puntuación del 0.81891.

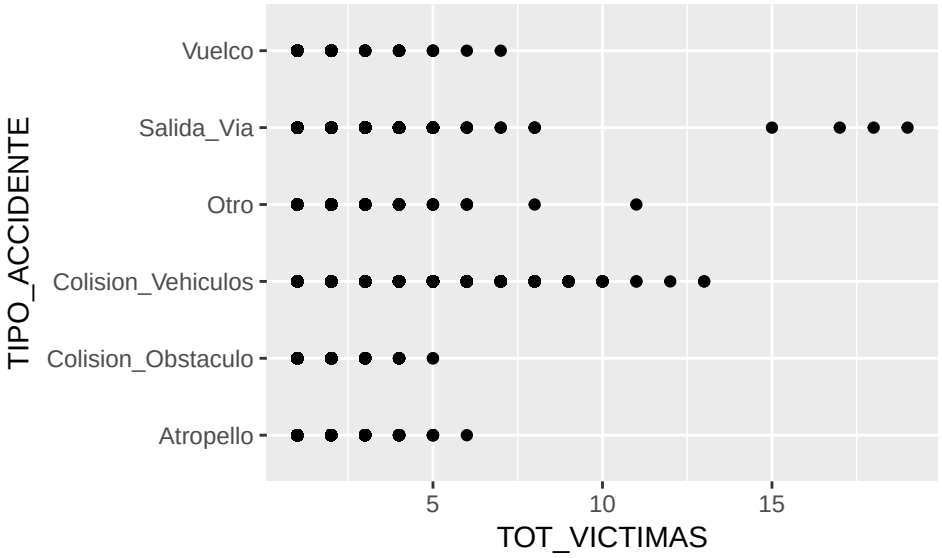
3 Vusualización del dataset

Como no se ha hecho antes, y debería ser uno de los primeros pasos a realizar, vamos a realizar una visualización del dataset.

3.1 Análisis de las variables actuales

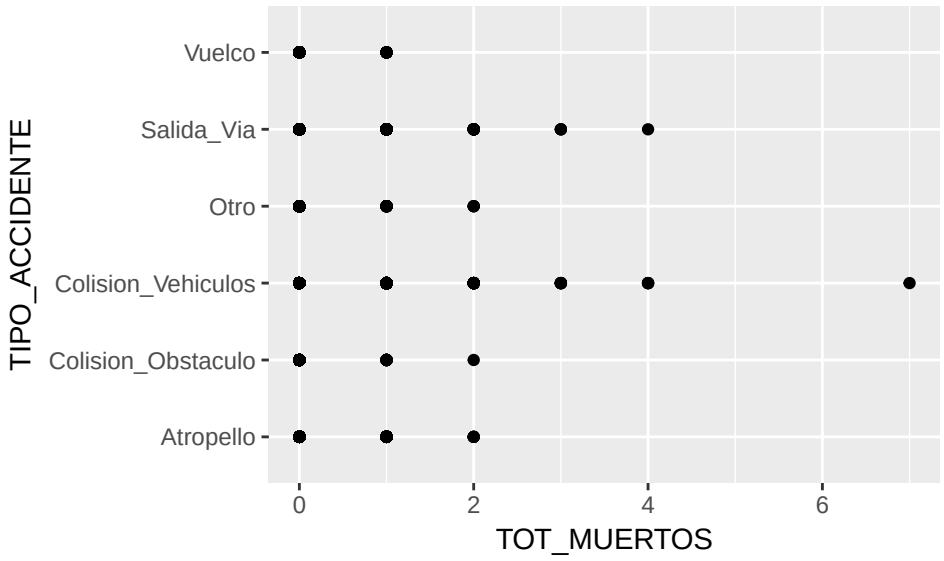
Vamos a ver el comportamiento de nuestras variables con respecto al TIPO_ACCIDENTE, a ver que relación pueden tener.

```
ggplot(data = accidentes.train.sin.variables.2) + geom_point(mapping = aes(x = TOT_VICTIMAS , y = TIPO_A
```

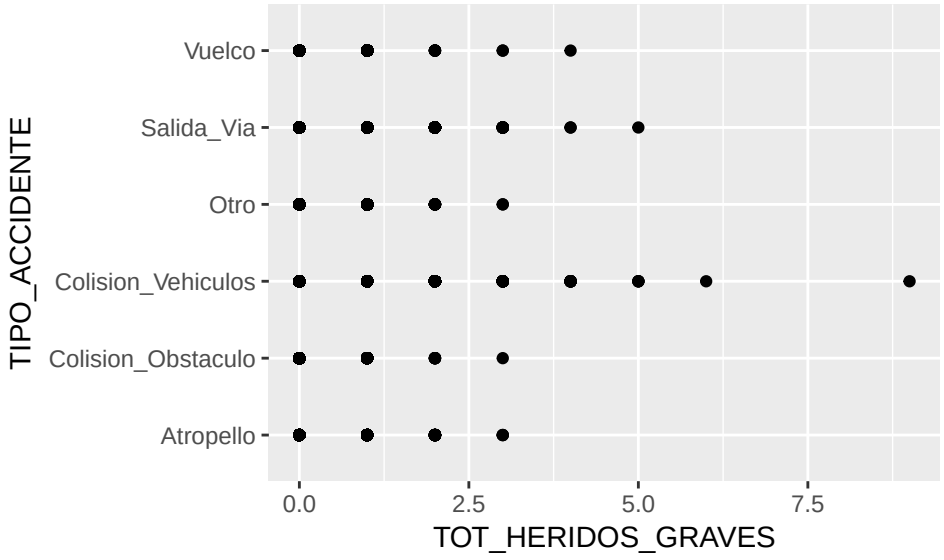


Podemos ver como para a partir de 10 victimas, el accidente suele ser o una colisión de vehículos, salida de vía, o muy pocas veces otro tipo de accidente. Por lo que puede ser una relación interesante.

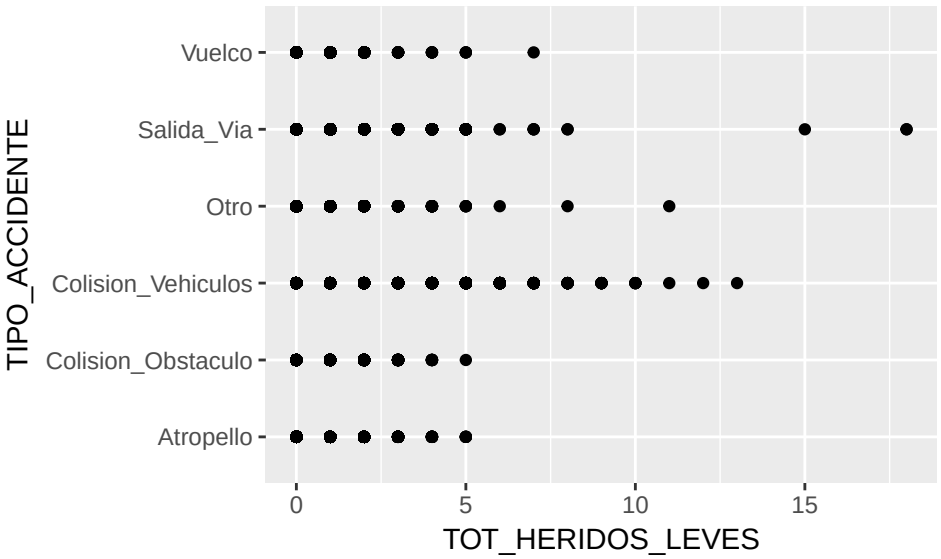
```
ggplot(data = accidentes.train.sin.variables.2) + geom_point(mapping = aes(x = TOT_MUERTOS , y = TIPO_A
```



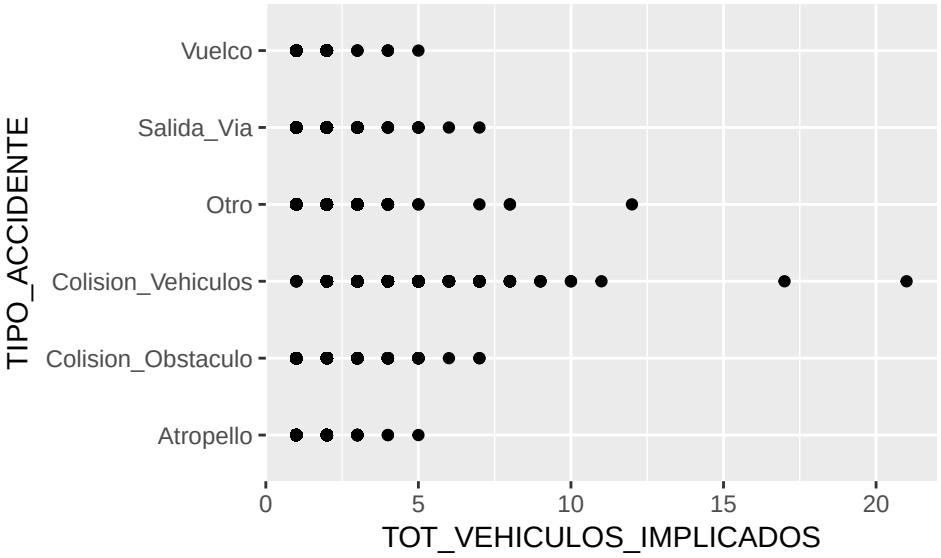

```
ggplot(data = accidentes.train.sin.variables.2) + geom_point(mapping = aes(x = TOT_HERIDOS_GRAVES , y =
```




```
ggplot(data = accidentes.train.sin.variables.2) + geom_point(mapping = aes(x = TOT_HERIDOS_LEVES , y = 
```

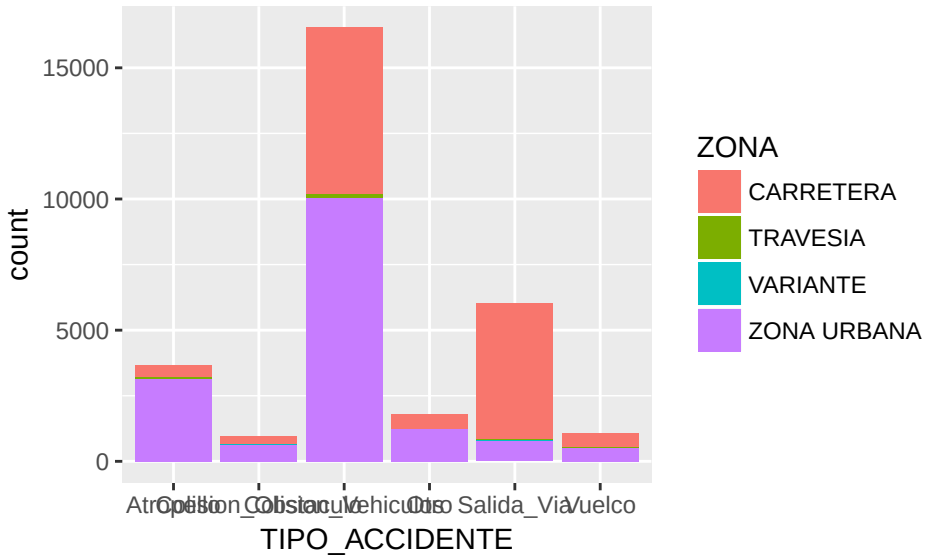



```
ggplot(data = accidentes.train.sin.variables.2) + geom_point(mapping = aes(x = TOT_VEHICULOS_IMPLICADOS
```



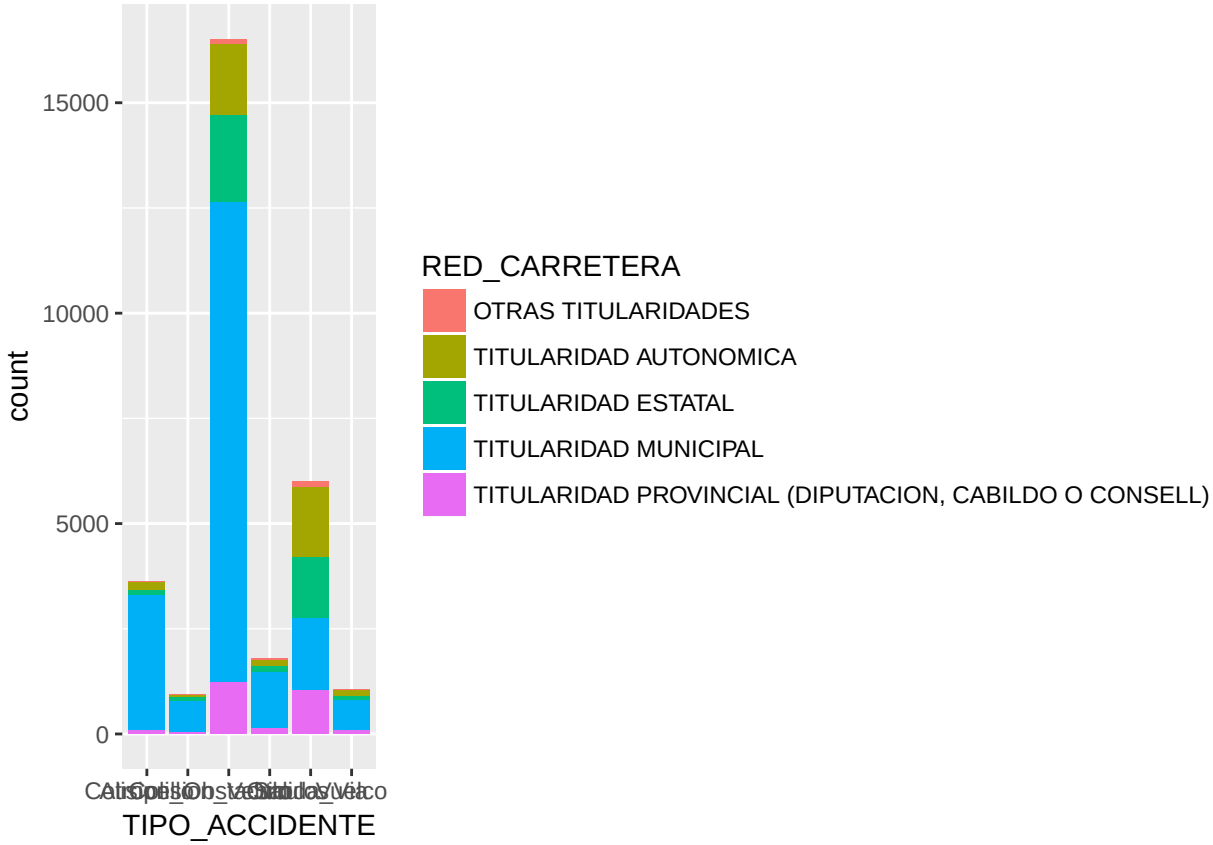
Normalmente a partir de 3 muertos, el accidente es una colisión de vehículos o una salida de vía. Si hay más de 3 heridos graves, suele ser colisión de vehículos, salida de vía o vuelco. A partir de 6 heridos leves el accidente es una colisión, una salida de vía, un vuelco o otro accidente. A partir de 6 vehículos implicados, los accidentes suelen ser colisiones, salida de vía u otro tipo. Por lo que ya tenemos varias relaciones que podrían ser representadas en un árbol.

```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=ZONA))
```



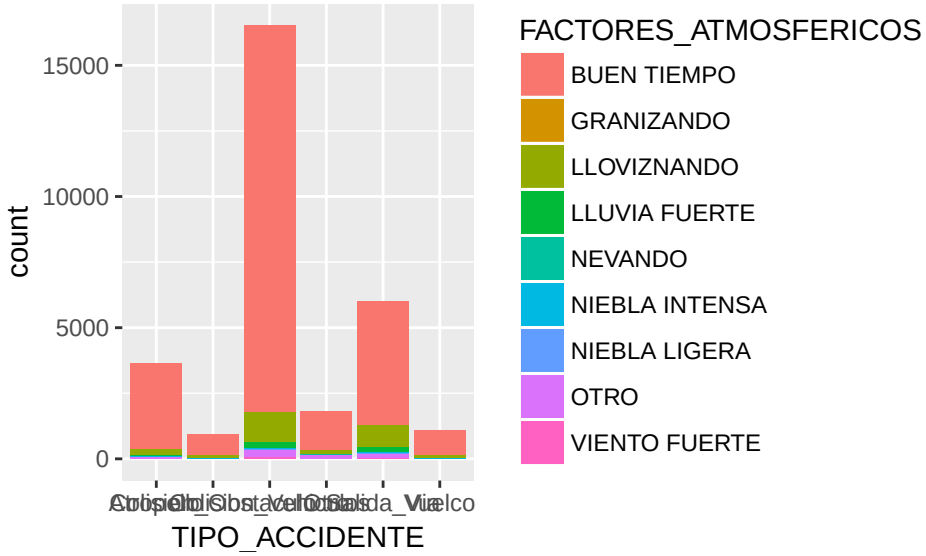
Podemos ver como las zonas predominantes son carretera y zona urbana, pero no parece que esta variable pueda ser influyente a la hora de decir que tipo de accidente se produce por lo que eliminaré esta variable para futuras pruebas.

```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=RED_CARR
```



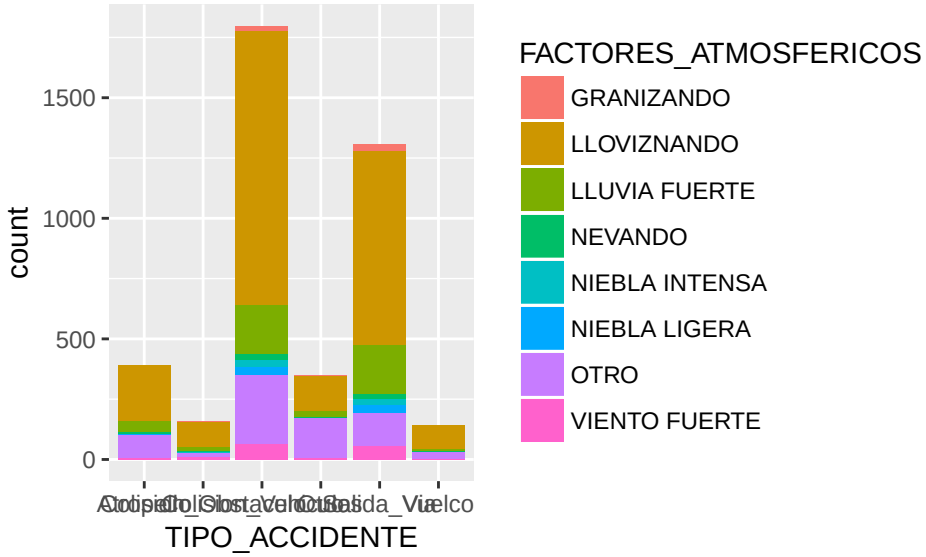
Puede parecer que esta variable no tiene demasiado que ver con la variable que queremos predecir por lo que puede ser que la descartemos.

```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=FACTORES
```



Por el conocimiento que tenemos, seguramente esta variable no sea demasiado importante para el tipo de accidente. Veamos que le ocurre si eliminamos los elementos que tienen buen tiempo.

```
vector.buen.tiempo <- accidentes.train.sin.variables.1$FACTORES_ATMOSFERICOS == "BUEN TIEMPO"
valores.sin.buen.tiempo <- accidentes.train.sin.variables.1[!vector.buen.tiempo,]
ggplot(data = valores.sin.buen.tiempo) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=FACTORES_ATMOSFE
```



Pero seguimos viendo que no se puede sacar ninguna conclusión de esta visualización.

3.2 Análisis de variables eliminadas sin valores perdidos

Recordemos las variables que eliminamos sin tener valores perdidos.

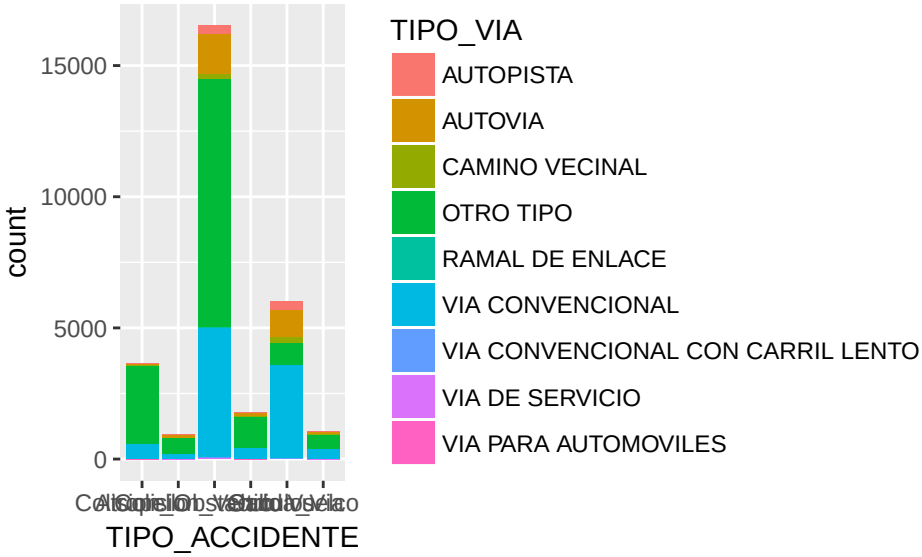
```
segundas.variables.eliminadas
```

```
##  [1] "ANIO"                "MES"                 "HORA"
##  [4] "DIASEMANA"           "PROVINCIA"           "COMUNIDAD_AUTONOMA"
```

```
##  [7] "ISLA"                "ZONA_AGRUPADA"       "TIPO_VIA"
## [10] "TRAZADO_NO_INTERSEC" "TIPO_INTERSEC"       "SUPERFICIE_CALZADA"
## [13] "LUMINOSIDAD"
```

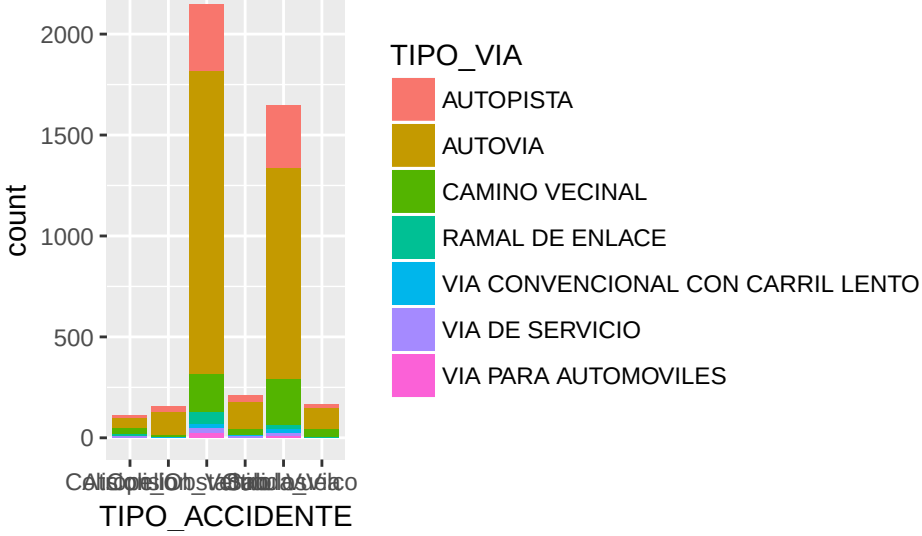
Una de la variables que podrían ser interesantes es TIPO_VIA, TRAZADO_NO_INTERSEC, TIPO_INTERSEC, SUPERFICIE_CALZADA y LUMINOSIDAD. Veamos visualizaciones de estas variables.

```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=TIPO_VI
```



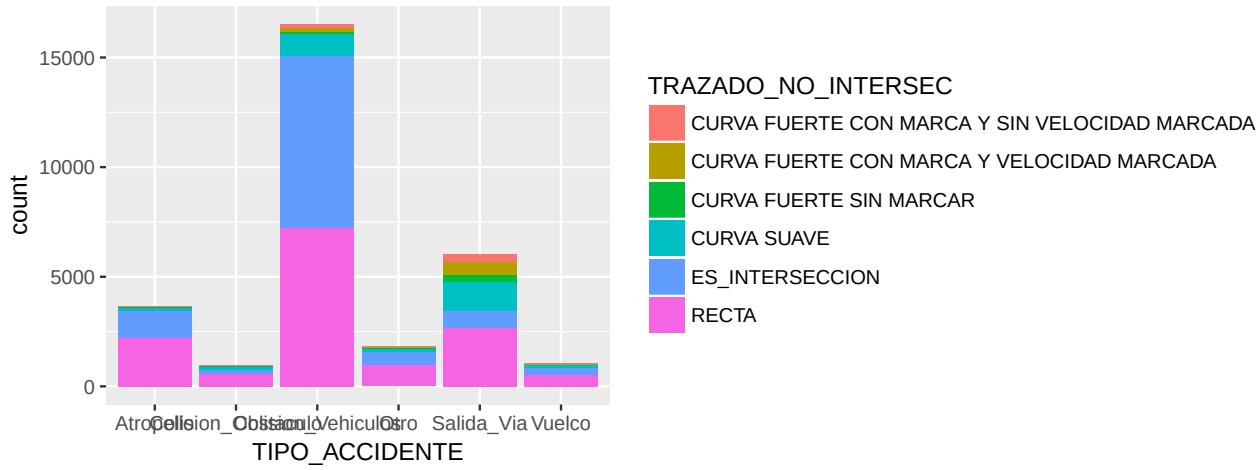
Eliminemos las instancias con OTRO TIPO o VIA CONVENCIONAL

```
vector.sin.otrotipo.y.viaconvencional <- ((accidentes.train.sin.variables.1$TIPO_VIA == "OTRO TIPO") |
valores.sin.otrotipo.y.viaconvencional <- accidentes.train.sin.variables.1[!vector.sin.otrotipo.y.viacon
ggplot(data = valores.sin.otrotipo.y.viaconvencional) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=TI
```

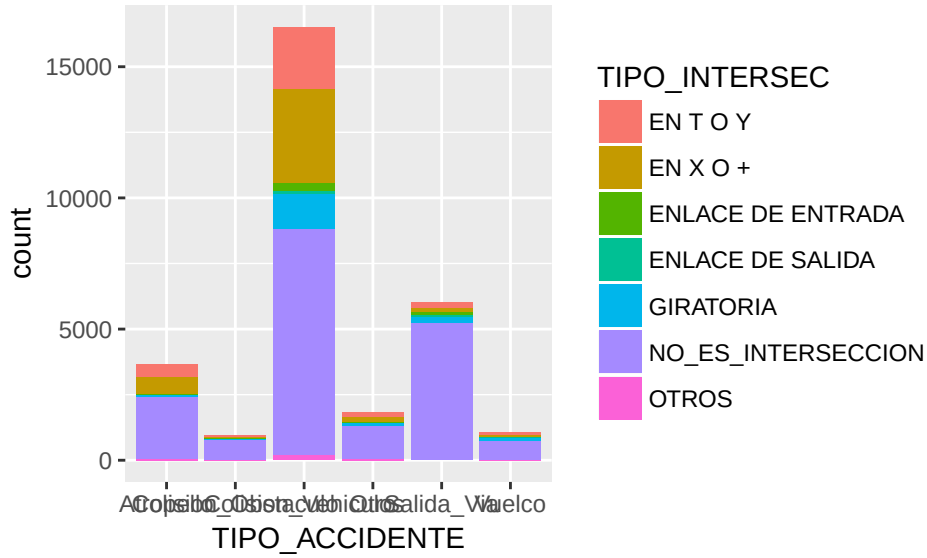


No se observa que sea una variable demasiada importante.

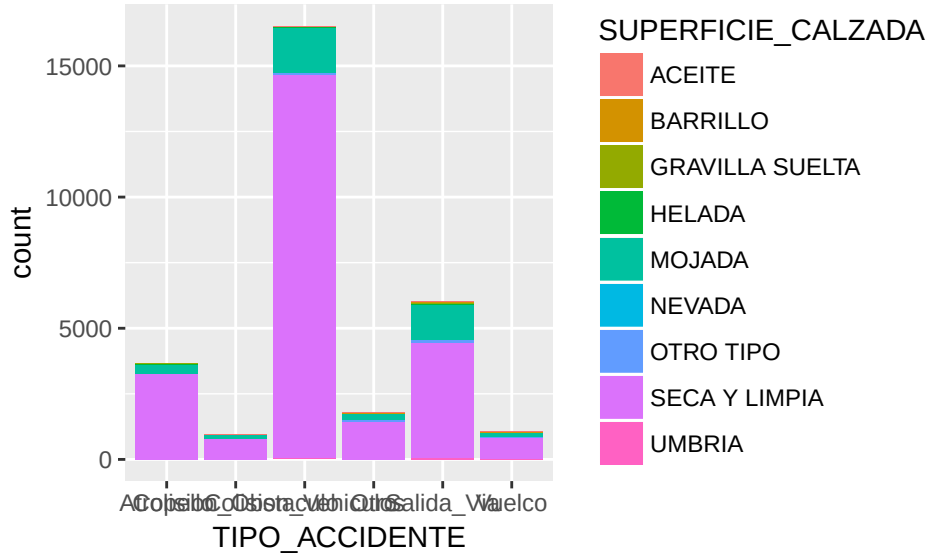
```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=TRAZADO_
```



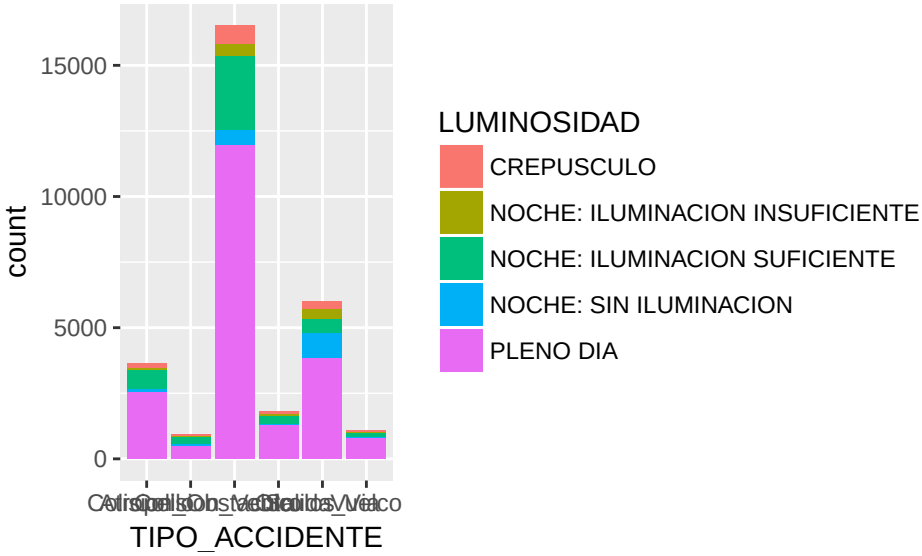

```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=TIPO_IN
```




```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=SUPERFI
```



```
ggplot(data = accidentes.train.sin.variables.1) + geom_bar(mapping = aes(x=TIPO_ACCIDENTE, fill=LUMINOSI
```



Por lo que no podemos sacar demasiada información así que no añadiremos ninguna a las que ya estamos usando de momento.

4 Visión preeliminar de los datos

Como anteriormente ya hicimos el summary, no será necesario volver a hacerlo. Lo que si vamos a hacer es un str, para obtener la información de las variables.

```
str(accidentes.train.sin.variables.1)
```

```
## 'data.frame':    30002 obs. of  22 variables:
##  $ ANIO                     : int  2009 2011 2008 2013 2009 2008 2010 2010 2013 2009 ...
##  $ MES                      : Factor w/ 12 levels "Abril","Agosto",..: 8 5 8 10 1 6 6 7 11 10 ...
##  $ HORA                     : Factor w/ 448 levels "0","0,016666667",..: 266 266 136 328 49 411 31 13
##  $ DIASEMANA                : Factor w/ 7 levels "DOMINGO","JUEVES",..: 7 3 6 7 7 6 4 1 7 6 ...
##  $ PROVINCIA                : Factor w/ 52 levels "Albacete","Alicante/Alacant",..: 13 39 49 11 2 23 
##  $ COMUNIDAD_AUTONOMA       : Factor w/ 18 levels "Andalucia","Aragon",..: 1 13 11 7 11 1 9 11 14 9 .
##  $ ISLA                     : Factor w/ 10 levels "FORMENTERA","FUERTEVENTURA",..: 9 9 9 9 9 9 9 9 9 
##  $ TOT_VICTIMAS             : int  1 1 1 3 1 2 3 1 1 1 ...
##  $ TOT_MUERTOS              : int  0 0 1 0 0 1 0 0 0 0 ...
##  $ TOT_HERIDOS_GRAVES       : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ TOT_HERIDOS_LEVES        : int  1 1 0 3 1 0 3 1 1 1 ...
##  $ TOT_VEHICULOS_IMPLICADOS: int  2 2 1 3 1 1 3 2 1 4 ...
##  $ ZONA                     : Factor w/ 4 levels "CARRETERA","TRAVESIA",..: 4 1 1 4 1 1 4 4 4 4 ...
##  $ ZONA_AGRUPADA            : Factor w/ 2 levels "VIAS INTERURBANAS",..: 2 1 1 2 1 1 2 2 2 2 ...
##  $ RED_CARRETERA            : Factor w/ 5 levels "OTRAS TITULARIDADES",..: 4 2 5 4 3 5 4 4 4 4 ...
##  $ TIPO_VIA                 : Factor w/ 9 levels "AUTOPISTA","AUTOVIA",..: 4 6 6 4 1 6 4 4 4 4 ...
##  $ TRAZADO_NO_INTERSEC      : Factor w/ 6 levels "CURVA FUERTE CON MARCA Y SIN VELOCIDAD MARCADA",..:
##  $ TIPO_INTERSEC            : Factor w/ 7 levels "EN T O Y","EN X O +",..: 6 1 6 6 6 6 1 2 6 6 ...
##  $ SUPERFICIE_CALZADA       : Factor w/ 9 levels "ACEITE","BARRILLO",..: 8 8 8 5 8 8 8 8 8 8 ...
##  $ LUMINOSIDAD              : Factor w/ 5 levels "CREPUSCULO","NOCHE: ILUMINACION INSUFICIENTE",..: 5
##  $ FACTORES_ATMOSFERICOS    : Factor w/ 9 levels "BUEN TIEMPO",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ TIPO_ACCIDENTE           : Factor w/ 6 levels "Atropello","Colision_Obstaculo",..: 3 3 5 3 5 5 3 3
```

Si queremos información más detallada:

```
describe(accidentes.train.sin.variables.2[1])
```

```
## accidentes.train.sin.variables.2[1]
##
##  1  Variables      30002  Observations
## ---------------------------------------------------------------------------
## TOT_VICTIMAS
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    30002        0       17    0.609    1.429   0.6909        1        1
##      .25      .50      .75      .90      .95
##        1        1        2        2        3
##
## Value          1     2     3     4     5     6     7     8     9    10
## Frequency  21826  5503  1540   681   248   105    43    25    13     8
## Proportion 0.727 0.183 0.051 0.023 0.008 0.003 0.001 0.001 0.000 0.000
##
## Value         11    12    13    15    17    18    19
## Frequency      3     1     2     1     1     1     1
## Proportion 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## ---------------------------------------------------------------------------
```

Esto lo podemos hacer con las variables que veamos oportunas. Otra forma de ver más información es:

```
basicStats(accidentes.train.sin.variables.2[1])
```

```
##             TOT_VICTIMAS
## nobs        30002.000000
## NAs             0.000000
## Minimum         1.000000
## Maximum        19.000000
## 1. Quartile     1.000000
## 3. Quartile     2.000000
## Mean            1.429371
## Median          1.000000
## Sum         42884.000000
## SE Mean         0.005258
## LCL Mean        1.419066
## UCL Mean        1.439677
## Variance        0.829334
## Stdev           0.910678
## Skewness        3.817690
## Kurtosis       27.886723
```

5 Imputación de valores perdidos

Vamos a usar uso del paquete mice para imputar los datos.

5.1 Imputación de variables

Veamos que variables teníamos con valores perdidos.

```
summary(accidentes.train.variables.eliminadas)
```

```
##    CARRETERA
##  A-7    :  294
##  A-2    :  278
##  AP-7   :  229
##  N-340  :  229
##  A-4    :  184
##  (Other):12098
##  NA's   :16690
##                                                ACOND_CALZADA
##  CARRIL CENTRAL DE ESPERA                                :  193
##  NADA ESPECIAL                                           : 4645
##  OTRO TIPO                                               :  791
##  PASO PARA PEATONES O ISLETAS EN CENTRO DE VIA PRINCIPAL:  397
##  RAQUETA DE GIRO IZQUIERDA                               :  109
##  SOLO ISLETAS O PASO PARA PEATONES                       :  168
##  NA's                                                    :23699
##                  PRIORIDAD                   VISIBILIDAD_RESTRINGIDA
##  NINGUNA (SOLO NORMA) :13495   SIN RESTRICCION          :16982
##  SEMAFORO             : 1778   CONFIGURACION DEL TERRENO:  989
##  SEÆAL DE STOP        : 1750   OTRA_CAUSA               :  491
##  SOLO MARCAS VIALES   : 1659   FACTORES ATMOSFERICOS    :  374
##  SEÆAL DE CEDA EL PASO: 1629   EDIFICIOS                :  229
##  (Other)              : 1569   (Other)                  :  252
##  NA's                 : 8122   NA's                     :10685
##          OTRA_CIRCUNSTANCIA          ACERAS         DENSIDAD_CIRCULACION
##  NINGUNA          :24967    NO HAY ACERA:21416   CONGESTIONADA:  308
##  OTRA             :  942    SI HAY ACERA: 5437   DENSA        : 1479
##  OBRAS            :  263    NA's        : 3149   FLUIDA       :17505
##  FUERTE DESCENSO  :  227                         NA's         :10710
##  CAMBIO DE RASANTE:  100
##  (Other)          :  264
##  NA's             : 3239
##           MEDIDAS_ESPECIALES
##  CARRIL REVERSIBLE :   17
##  HABILITACION ARCEN:    8
##  NINGUNA MEDIDA    :21024
##  OTRA MEDIDA       :  278
##  NA's              : 8675
##
##
```

Vemos que dos de estas variables que podrían ser más interesantes son visibilidad restringida y prioridad, por lo que vamos a proceder a imputar sus valores perdidos.

```
accidentes.train.a.imputar <- cbind(accidentes.train.sin.variables.2, accidentes.train.variables.elimina
accidentes.test.a.imputar <- cbind(accidentes.test.sin.variables.2, accidentes.test.variables.eliminadas
set.seed(1234)
train.imputados.incompletos <- mice::mice(accidentes.train.a.imputar, m=1, method="pmm")
train.imputados <- mice::complete(train.imputados.incompletos)
test.imputados.incompletos <- mice::mice(accidentes.test.a.imputar, m=5, method="pmm")
test.imputados <- mice::complete(test.imputados.incompletos)
```

5.2 Prueba del modelo con imputación de valores perdidos

Hagamos por lo tanto una prueba de como afecta la imputación de valores perdidos.

```
set.seed(1234)
ct3 <- ctree(TIPO_ACCIDENTE ~., train.imputados)
testPred3 <- predict(ct3, newdata = test.imputados)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct3
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.tercer.modelo <- as.matrix(testPred3)
salida.tercer.modelo <- cbind(c(1:(dim(salida.tercer.modelo)[1])), salida.tercer.modelo)
colnames(salida.tercer.modelo) <- c("Id","Prediction")
write.table(salida.tercer.modelo,file="predicciones/TerceraPrediccion.txt",sep=",",quote = F,row.names =
```

3 El resultado de este modelo para la competición de Kaggel, subido el 19/02/2017 a las 17:42, con un total de 14 personas entregadas, se ha quedado en la posición 9 con una puntuación del 0.81753. Bajando muy poco con respecto a la anterior puntuación.

6 Detección de anomalias

Veamos como detectar valores anómalos en nuestros datos.

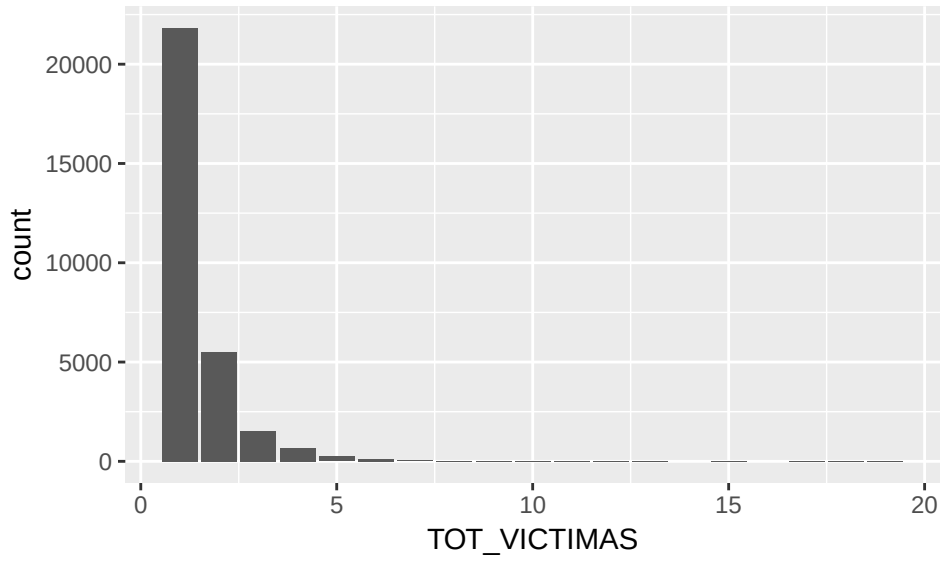
6.1 Uso del paquete outliers

Veamos si tenemos valores perdidos en nuestros datos, solo con valores que no son discretas.

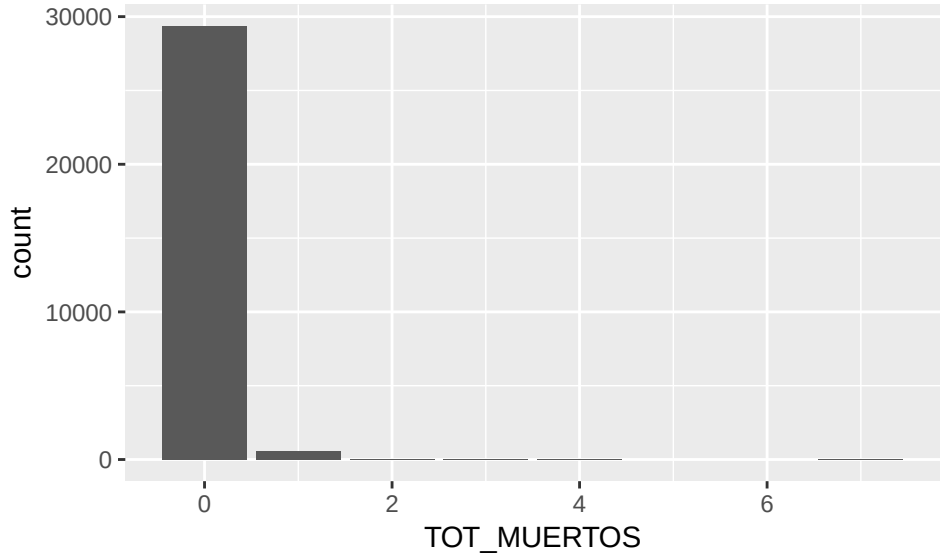
```
valores.anomalos <- outliers::outlier(train.imputados[,1:5])
print(valores.anomalos)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                       19                         7                         9
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                       18                        21
```

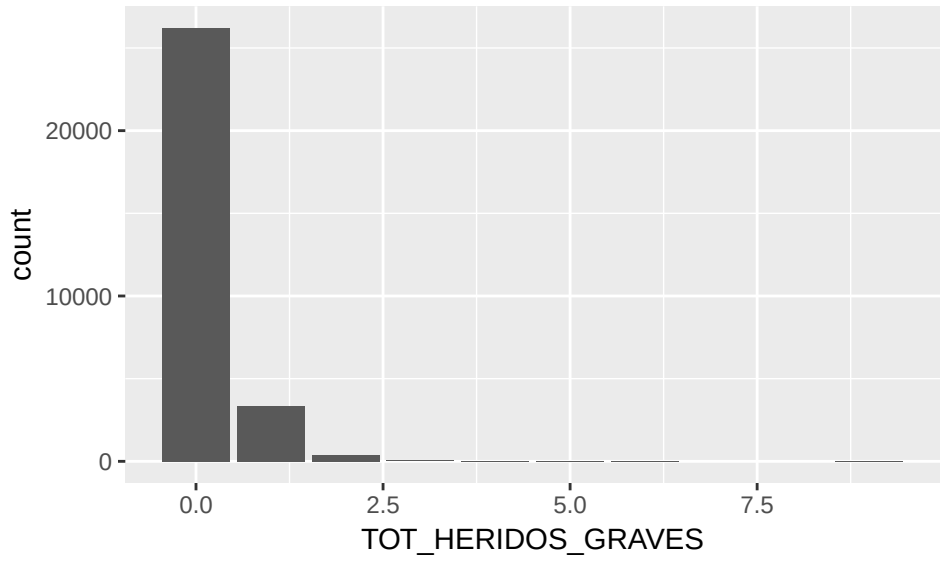
```
ggplot(data = train.imputados) + geom_bar(mapping = aes(x=TOT_VICTIMAS))
```



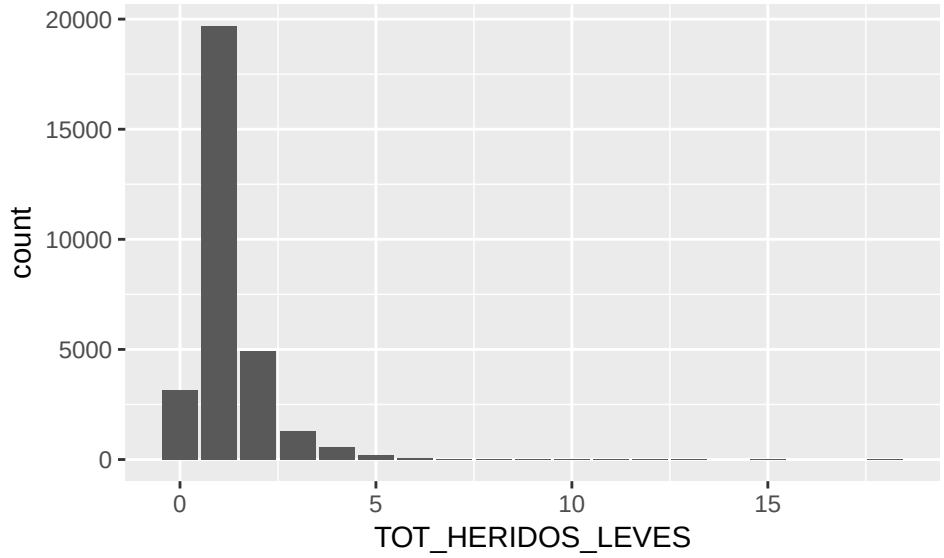

```
ggplot(data = train.imputados) + geom_bar(mapping = aes(x=TOT_MUERTOS))
```



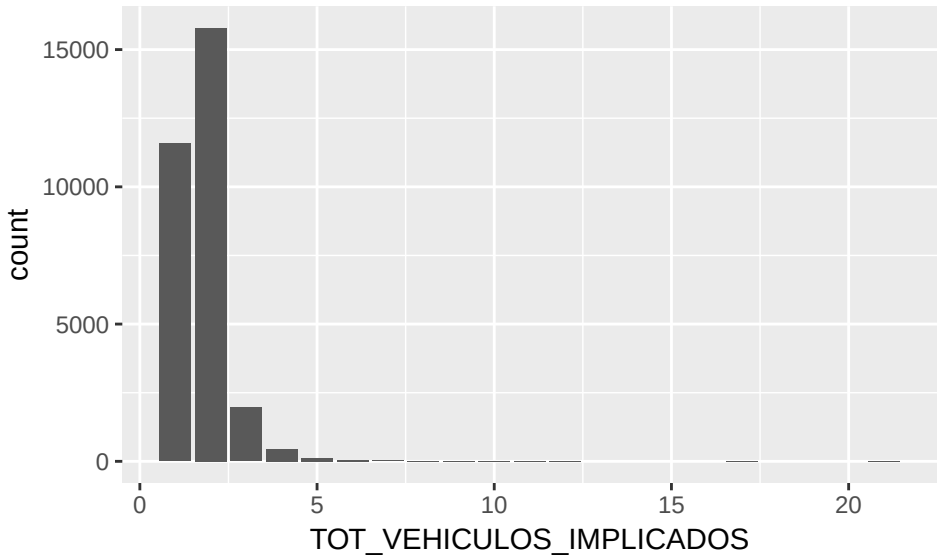

```
ggplot(data = train.imputados) + geom_bar(mapping = aes(x=TOT_HERIDOS_GRAVES))
```




```
ggplot(data = train.imputados) + geom_bar(mapping = aes(x=TOT_HERIDOS_LEVES))
```




```
ggplot(data = train.imputados) + geom_bar(mapping = aes(x=TOT_VEHICULOS_IMPLICADOS))
```



Viendo que en cada variable tenemos distintos valores anómalos como sería el valor 19 en TOT_VICTIMAS.

6.2 Paquete mvoutlier

Voy a intentar usar el paquete mvoutlier.

```
require(mvoutlier)
#resultado.busqueda.anomalias <- uni.plot(train.imputados[1:200,1:2])
```

Como se puede ver, se ha obtenido un error el cual no he podido solucionar.

6.3 Eliminación de valores anómalos

En función de lo obtenido con el paquete outlier, voy a intentar realizar algo con este paquete para ver que tal se comporta nuestro dataset.

```
valores.anomalos.train <- outliers::outlier(train.imputados[,1:5])
valores.anomalos.test <- outliers::outlier(test.imputados[,1:5])
print(valores.anomalos.train)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                       19                        7                        9
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                       18                       21
```

```
print(valores.anomalos.test)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                       10                        5                        5
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                       10                       11
```

Veamos, por ejemplo, para la variable TOT_VICTIMAS, cuantas instancias cumplen tener mas de 19 victimas o de 10.

```
vector.con.victimas.19 <- train.imputados$TOT_VICTIMAS >= 19
sum(vector.con.victimas.19)
```

```
## [1] 1
```

```
vector.con.victimas.18 <- train.imputados$TOT_VICTIMAS >= 18
sum(vector.con.victimas.18)
```

```
## [1] 2
```

```
vector.con.victimas.17 <- train.imputados$TOT_VICTIMAS >=17
sum(vector.con.victimas.17)
```

```
## [1] 3
```

```
vector.con.victimas.10 <- train.imputados$TOT_VICTIMAS >= 10
sum(vector.con.victimas.10)
```

```
## [1] 18
```

```
valores.con.victimas.10 <- train.imputados[vector.con.victimas.10,]
valores.con.victimas.10$TIPO_ACCIDENTE
```

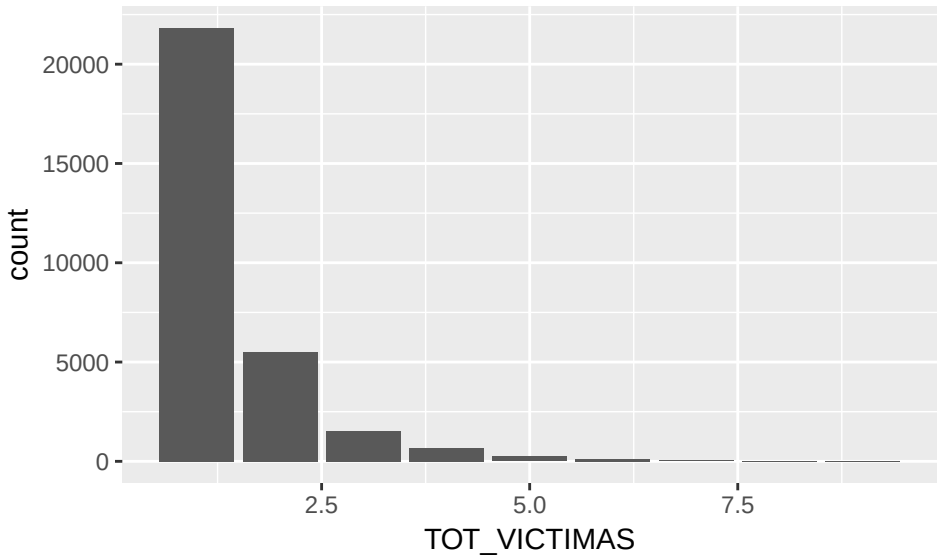
```
##  [1] Salida_Via         Colision_Vehiculos Colision_Vehiculos
##  [4] Colision_Vehiculos Colision_Vehiculos Colision_Vehiculos
##  [7] Salida_Via         Colision_Vehiculos Salida_Via
## [10] Colision_Vehiculos Colision_Vehiculos Colision_Vehiculos
## [13] Salida_Via         Otro               Colision_Vehiculos
## [16] Colision_Vehiculos Colision_Vehiculos Colision_Vehiculos
## attr(,"contrasts")
##                    2 3 4 5 6
## Atropello          0 0 0 0 0
## Colision_Obstaculo 1 0 0 0 0
## Colision_Vehiculos 0 1 0 0 0
## Otro               0 0 1 0 0
## Salida_Via         0 0 0 1 0
## Vuelco             0 0 0 0 1
## 6 Levels: Atropello Colision_Obstaculo Colision_Vehiculos ... Vuelco
```

Vemos que no son demasiados datos, ya que en total son 18 instancias, por lo que vamos a probar a eliminarlas a ver el comportamiento del paquete outlier de nuevo.

```
train.sin.outliers <- train.imputados[!vector.con.victimas.10,]
valores.anomalos.sin.victimas.10 <- outliers::outlier(train.sin.outliers[,1:5])
print(valores.anomalos.sin.victimas.10)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                        9                        7                        6
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                        9                       17
```

```
ggplot(data = train.sin.outliers) + geom_bar(mapping = aes(x=TOT_VICTIMAS))
```



Vamos a probar a eliminar algunas instancias, con los criterios de otras variables.

```
vector.con.muertos.7 <- train.sin.outliers$TOT_MUERTOS >=7
sum(vector.con.muertos.7)
```

```
## [1] 1
```

```
vector.con.muertos.6 <- train.sin.outliers$TOT_MUERTOS >= 6
sum(vector.con.muertos.6)
```

```
## [1] 1
```

```
vector.con.muertos.5 <- train.sin.outliers$TOT_MUERTOS >= 5
sum(vector.con.muertos.5)
```

```
## [1] 1
```

```
vector.con.muertos.4 <- train.sin.outliers$TOT_MUERTOS >= 4
sum(vector.con.muertos.4)
```

```
## [1] 6
```

```
train.sin.outliers <- train.sin.outliers[!vector.con.muertos.4,]
valores.anomalos.sin.muertos.4 <- outliers::outlier(train.sin.outliers[,1:5])
print(valores.anomalos.sin.muertos.4)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                        9                        3                        6
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                        9                       17
```

Vamos a realizarlo más rápidamente

```
vector.con.anomalias <- ((train.sin.outliers$TOT_HERIDOS_GRAVES >= 6) | (train.sin.outliers$TOT_HERIDOS
sum(vector.con.anomalias)
```

```
## [1] 10
```

```
vector.con.anomalias <- ((train.sin.outliers$TOT_HERIDOS_GRAVES >= 5) | (train.sin.outliers$TOT_HERIDOS
sum(vector.con.anomalias)
```

```
## [1] 31
```

```
train.sin.outliers <- train.sin.outliers[!vector.con.anomalias,]
```

Pero, que pasaría si eliminamos en función de las anomalías que nos marca el test:

```
print(valores.anomalos.test)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                       10                        5                        5
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                       10                       11
```

```
vector.con.anomalias <- ((train.imputados$TOT_HERIDOS_GRAVES > 5) | (train.imputados$TOT_HERIDOS_LEVES >
sum(vector.con.anomalias)
```

```
## [1] 14
```

En total eliminaríamos 14 instancias. Vamos a comprobarlo:

```
train.sin.outliers <- train.imputados[!vector.con.anomalias,]
valores.anomalos.train <- outliers::outlier(train.sin.outliers[,1:5])
print(valores.anomalos.train)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                       10                        4                        5
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                       10                       11
```

```
print(valores.anomalos.test)
```

```
##             TOT_VICTIMAS              TOT_MUERTOS       TOT_HERIDOS_GRAVES
##                       10                        5                        5
##        TOT_HERIDOS_LEVES TOT_VEHICULOS_IMPLICADOS
##                       10                       11
```

6.4 Prueba del modelo con imputación de valores perdidos

Hagamos por lo tanto una prueba de como afecta la imputación de valores perdidos.

```
set.seed(1234)
ct4 <- ctree(TIPO_ACCIDENTE ~., train.sin.outliers)
testPred4 <- predict(ct4, newdata = test.imputados)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct4
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.4 <- as.matrix(testPred4)
salida.modelo.4 <- cbind(c(1:(dim(salida.modelo.4)[1])), salida.modelo.4)
colnames(salida.modelo.4) <- c("Id","Prediction")
write.table(salida.modelo.4,file="predicciones/Prediccion4.txt",sep=",",quote = F,row.names = F)
```

4 El resultado de este modelo para la competición de Kaggel, subido el 19/02/2017 a las 20:12, con un total de 16 personas entregadas, se ha quedado en la posición 10 con una puntuación del 0.81753. Bajando muy poco con respecto a la anterior puntuación.

7 Transformación de los datos

Tal y como se vio en el guión de prácticas en el punto 7, vamos a aplicar la transformación para ver que tal nos funciona.

7.1 Transformando los datos

Vamos a aplicar centrado y escalado sobre el conjunto de datos con los valores ya imputados, para las variables que se consideran continuas.

```
valores.preprocesados <- caret::preProcess(train.sin.outliers[,1:5],method=c("center","scale"))
valores.transofrmados <- predict(valores.preprocesados,train.sin.outliers[,1:5])
train.transformado <- cbind(valores.transofrmados,train.sin.outliers[,6:11])
valores.preprocesados.test <- caret::preProcess(test.imputados[,1:5],method=c("center","scale"))
valores.transofrmados.test <- predict(valores.preprocesados.test,test.imputados[,1:5])
test.transformado <- cbind(valores.transofrmados.test,test.imputados[,6:10])
```

7.2 Prueba del modelo con transformación de los datos

Hagamos por lo tanto una prueba de como afecta la transformación de los datos.

```
set.seed(1234)
ct5 <- ctree(TIPO_ACCIDENTE ~., train.transformado)
testPred5 <- predict(ct5, newdata = test.transformado)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct5
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.5 <- as.matrix(testPred5)
salida.modelo.5 <- cbind(c(1:(dim(salida.modelo.5)[1])), salida.modelo.5)
colnames(salida.modelo.5) <- c("Id","Prediction")
write.table(salida.modelo.5,file="predicciones/Prediccion5.txt",sep=",",quote = F,row.names = F)
```

5 El resultado de este modelo para la competición de Kaggel, subido el 20/02/2017 a las 13:15, con un total de 18 personas entregadas, se ha quedado en la posición 12 con una puntuación del 0.55147. Bajando mucho con respecto a la anterior puntuación, por lo que esta transformación no la tendremos en cuenta.

8 Discretización

Para este conjunto de datos no se realiza discretización ya que no tenemos variables continuas como para poder discretizarlas.

9 Selección de características

Para este apartado comenzaremos con los dataset originales.

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
```

9.1 Paquete FSelector

9.1.1 Aproximación filter: chi.squared

Determina los pesos de los atributos discretos usando el test de independencia chi-cuadrado (con respecto a la variable clase). Calculamos los pesos de los atributos: la medida devuelta indica el nivel de dependencia de cada atributo frente a la variable clase

```
set.seed(1234)
pesos <- FSelector::chi.squared(TIPO_ACCIDENTE~.,train.original)
pesos
```

Vamos a seleccionar los 7 mejores

```
subset <- FSelector::cutoff.k(pesos, 7)
las.7.mas.importantes.chi.squared <- as.simple.formula(subset, "TIPO_ACCIDENTE")
las.7.mas.importantes.chi.squared
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + CARRETERA + ZONA_AGRUPADA +
##     ZONA + ACERAS + PRIORIDAD + RED_CARRETERA
## <environment: 0x7fd6a1043708>
```

Por lo que vamos a montar un modelo con estas variables

```
train.filter.chi.squared <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","CARRETERA","ZONA_AGRUPADA","ZO
test.filter.chi.squared <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","CARRETERA","ZONA_AGRUPADA","ZONA
```

Vemos que la variable CARRETERA tiene un alto número de valores perdidos por lo que la vamos a descartar, a pesar de que la selección de características nos ha dicho que es importante.

```
train.filter.chi.squared["CARRETERA"] <- NULL
test.filter.chi.squared["CARRETERA"] <- NULL
```

9.1.2 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct6 <- ctree(TIPO_ACCIDENTE ~., train.filter.chi.squared)
testPred6 <- predict(ct6, newdata = test.filter.chi.squared)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct6
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.6 <- as.matrix(testPred6)
salida.modelo.6 <- cbind(c(1:(dim(salida.modelo.6)[1])), salida.modelo.6)
colnames(salida.modelo.6) <- c("Id","Prediction")
write.table(salida.modelo.6,file="predicciones/Prediccion6.txt",sep=",",quote = F,row.names = F)
```

6 El resultado de este modelo para la competición de Kaggel, subido el 22/02/2017 a las 13:20, con un total de 21 personas entregadas, se ha quedado en la posición 13 con una puntuación del 0.82089. Mejorando a la que ya se tenia anteriormente, por lo que vemos que esta selección de características ha funcionado correctamente.

9.1.3 Aproximación filter: correlation

Busca los pesos de atributos continuos en base a medidas de correlación. Por lo tanto esta aproximación no podremos realizarla al tener la variable clase no numérica.

9.1.4 Aproximación filter: entropy.based

Encontraremos los pesos de los atributos discretos en base a su correlación con el atributo clase.

```
set.seed(1234)
pesos <- FSelector::information.gain(TIPO_ACCIDENTE~., train.original)
subset <- cutoff.k(pesos,7)
los.7.mas.importantes.information.gain <- as.simple.formula(subset, "TIPO_ACCIDENTE")
los.7.mas.importantes.information.gain
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + CARRETERA + ZONA +
##     ZONA_AGRUPADA + TIPO_VIA + TRAZADO_NO_INTERSEC + PRIORIDAD
## <environment: 0x7fd6886515f0>
```

```
set.seed(1234)
pesos <- FSelector::gain.ratio(TIPO_ACCIDENTE~., train.original)
subset <- cutoff.k(pesos,7)
los.7.mas.importantes.gain.ratio <- as.simple.formula(subset, "TIPO_ACCIDENTE")
los.7.mas.importantes.gain.ratio
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + ZONA_AGRUPADA + ZONA +
##     TIPO_VIA + CARRETERA + RED_CARRETERA + TRAZADO_NO_INTERSEC
## <environment: 0x7fd6887265f0>
```

```
set.seed(1234)
pesos <- FSelector::symmetrical.uncertainty(TIPO_ACCIDENTE~., train.original)
subset <- cutoff.k(pesos,7)
los.7.mas.importantes.symmetrical.uncertainty <- as.simple.formula(subset, "TIPO_ACCIDENTE")
los.7.mas.importantes.symmetrical.uncertainty
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + CARRETERA + ZONA_AGRUPADA +
##     ZONA + TIPO_VIA + TRAZADO_NO_INTERSEC + RED_CARRETERA
## <environment: 0x7fd69de52550>
```

Por lo que en función de estas tres salidas tenemos que las variables más importantes serían: TOT_VEHICULOS_IMPLICADOS, CARRETERA, ZONA, ZONA_AGRUPADA, TIPO_VIA, TRAZADO_NO_INTERSEC, PRIORIDAD y RED_CARRETERA. El único que difiere entre algoritmos es PRIORIDAD y RED_CARRETERA. Recordemos los que teníamos con chi.cuadrado:

```
las.7.mas.importantes.chi.squared
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + CARRETERA + ZONA_AGRUPADA +
##     ZONA + ACERAS + PRIORIDAD + RED_CARRETERA
## <environment: 0x7fd6a1043708>
```

Es decir, tenemos diferencias en ACERAS. Vamos a probar un modelo con todas las variables dadas por este nuevo método.

```
train.filter.entropy.bases <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","ZONA","TIPO_
test.filter.entropy.bases <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","ZONA","TIPO_V
```

Como anteriormente, hemos quitado la variable CARRETERA

9.1.5 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct7 <- ctree(TIPO_ACCIDENTE ~., train.filter.entropy.bases)
testPred7 <- predict(ct7, newdata = test.filter.entropy.bases)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct7
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.7 <- as.matrix(testPred7)
salida.modelo.7 <- cbind(c(1:(dim(salida.modelo.7)[1])), salida.modelo.7)
colnames(salida.modelo.7) <- c("Id","Prediction")
write.table(salida.modelo.7,file="predicciones/Prediccion7.txt",sep=",",quote = F,row.names = F)
```

7 El resultado de este modelo para la competición de Kaggel, subido el 22/02/2017 a las 14:00, con un total de 22 personas entregadas, se ha quedado en la posición 14 con una puntuación del 0.82227. Mejorando a la que ya se tenia anteriormente, por lo que vemos que esta selección de características ha funcionado mejor.

9.1.6 Aproximación filter: oneR

Método simple de cálculo de pesos para atributos discretos mediante el uso de reglas de asociación con un sólo término en el antecedente.

```
pesos <- FSelector::oneR(TIPO_ACCIDENTE~., train.original)
subset <- cutoff.k(pesos,7)
los.7.mas.importantes.oneR <- as.simple.formula(subset,"TIPO_ACCIDENTE")
los.7.mas.importantes.oneR
```

```
## TIPO_ACCIDENTE ~ ANIO + TOT_VEHICULOS_IMPLICADOS + ACERAS + DENSIDAD_CIRCULACION +
##     ZONA_AGRUPADA + TOT_MUERTOS + TOT_VICTIMAS
## <environment: 0x7fd683f39910>
```

Podemos ver que tenemos distintos atributos más importantes según este método, por lo que vamos aprobar que tal se comportan estos atributos.

```
train.filter.oneR <- train.original[,c("ANIO","TOT_VEHICULOS_IMPLICADOS","ACERAS","DENSIDAD_CIRCULACION
test.filter.oneR <- test.original[,c("ANIO","TOT_VEHICULOS_IMPLICADOS","ACERAS","DENSIDAD_CIRCULACION",
```

9.1.7 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct8 <- ctree(TIPO_ACCIDENTE ~., train.filter.oneR)
testPred8 <- predict(ct8, newdata = test.filter.oneR)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct8
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.8 <- as.matrix(testPred8)
salida.modelo.8 <- cbind(c(1:(dim(salida.modelo.8)[1])), salida.modelo.8)
colnames(salida.modelo.8) <- c("Id","Prediction")
write.table(salida.modelo.8,file="predicciones/Prediccion8.txt",sep=",",quote = F,row.names = F)
```

8 El resultado de este modelo para la competición de Kaggel, subido el 23/02/2017 a las 12:51, con un total de 22 personas entregadas, se ha quedado en la posición 14 con una puntuación del 0.81891. La cual no mejora a la mejor que ya teníamos.

9.1.8 Aproximación filter: relief

Algoritmo de búsqueda de pesos de atributos continuos y discretos en base a la distancia entre instancias.

```
pesos <- relief(TIPO_ACCIDENTE~., train.original, neighbours.count = 5, sample.size = 20)
pesos
```

```
subset <- cutoff.k(pesos,7)
los.7.mas.importantes.relief <- as.simple.formula(subset,"TIPO_ACCIDENTE")
los.7.mas.importantes.relief
```

```
## TIPO_ACCIDENTE ~ COMUNIDAD_AUTONOMA + PROVINCIA + MES + HORA +
##     PRIORIDAD + ANIO + LUMINOSIDAD
## <environment: 0x7fd683b548b0>
```

Vemos que los más importantes, según este método son: COMUNIDAD_AUTONOMA, PROVINCIA, MES, HORA, PRIORIDAD, ANIO y LUMINOSIDAD.

Vamos a probar que tal se comportan estos atributos. (HORA la elimino al tener un gran número de factores y ralentizar los cálculos)

```
train.filter.relief <- train.original[,c("COMUNIDAD_AUTONOMA","PROVINCIA","MES","PRIORIDAD","ANIO","LUM
test.filter.relief <- test.original[,c("COMUNIDAD_AUTONOMA","PROVINCIA","MES","PRIORIDAD","ANIO","LUMIN
```

9.1.9 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct9 <- ctree(TIPO_ACCIDENTE ~., train.filter.relief)
testPred9 <- predict(ct9, newdata = test.filter.relief)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct9
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.9 <- as.matrix(testPred9)
salida.modelo.9 <- cbind(c(1:(dim(salida.modelo.9)[1])), salida.modelo.9)
colnames(salida.modelo.9) <- c("Id","Prediction")
write.table(salida.modelo.9,file="predicciones/Prediccion9.txt",sep=",",quote = F,row.names = F)
```

9 El resultado de este modelo para la competición de Kaggel, subido el 28/02/2017 a las 11:51, con un total de 26 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.59119. La cual empeora mucho a lo que ya se tenia, por lo que no es una buena selección de características.

9.1.10 Aproximación wrapper: cfs

Decir que todos los métodos propuestos de wrapper como: best.first.search, exhaustive.search, greedy.search y hill.climbing.search, no me han funcionado debido al tipo de datos que tenemos. Por contra, el método cfs si ha funcionado correctamente.

```
set.seed(1234)
subset <- FSelector::cfs(TIPO_ACCIDENTE~.,train.original)
el.mejor.segun.cfs <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.cfs
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS
## <environment: 0x7fd683fe7228>
```

Siendo TOT_VEHICULOS_IMPLICADOS, la mejor característica, repitamos esto para obtener las 5 mejores, de forma que eliminamos la que mejor se obtiene.

```
set.seed(1234)
train.wrapper.cfs = train.original[,-12]
subset <- FSelector::cfs(TIPO_ACCIDENTE~.,train.wrapper.cfs)
el.mejor.segun.cfs <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.cfs
```

```
## TIPO_ACCIDENTE ~ ZONA_AGRUPADA + CARRETERA + TRAZADO_NO_INTERSEC
## <environment: 0x7fd683a6edb8>
```

Ahora obtenemos ZONA_AGRUPADA, CARRETERA y TRAZADO_NO_INTERSEC.

```
set.seed(1234)
train.wrapper.cfs = train.original[,-c(12,14,15,18)]
subset <- FSelector::cfs(TIPO_ACCIDENTE~.,train.wrapper.cfs)
el.mejor.segun.cfs <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.cfs
```

```
## TIPO_ACCIDENTE ~ TOT_HERIDOS_LEVES + ZONA + RED_CARRETERA + TIPO_VIA +
##     TIPO_INTERSEC + PRIORIDAD + SUPERFICIE_CALZADA + ACERAS
## <environment: 0x7fd6886c75f0>
```

Obteniendo: TOT_HERIDOS_LEVES, ZONA, RED_CARRETERA, TIPO_VIA, TIPO_INTERSEC, PRIORIDAD, SUPERFICIE_CALZADA y ACERAS.

Por lo tanto, ya que tenemos muchas características, vamos a probar el modelo con todas, salvo CARRETERA.

```
train.wrapper.cfs <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_INTERSEC"
test.wrapper.cfs <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_INTERSEC","
```

9.1.11 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct10 <- ctree(TIPO_ACCIDENTE ~., train.wrapper.cfs)
testPred10 <- predict(ct10, newdata = test.wrapper.cfs)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct10
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.10 <- as.matrix(testPred10)
salida.modelo.10 <- cbind(c(1:(dim(salida.modelo.10)[1])), salida.modelo.10)
colnames(salida.modelo.10) <- c("Id","Prediction")
write.table(salida.modelo.10,file="predicciones/Prediccion10.txt",sep=",",quote = F,row.names = F)
```

10 El resultado de este modelo para la competición de Kaggel, subido el 28/02/2017 a las 12:17, con un total de 26 personas entregadas, se ha quedado en la posición 15 con una puntuación del 0.82395. Por lo que se ha mejorado a la mejor obtenida por mi hasta el momento.

9.1.12 Aproximación wrapper: consistency

Probamos este método:

```
set.seed(1234)
subset <- consistency(TIPO_ACCIDENTE~.,train.original)
el.mejor.segun.consistency <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.consistency
```

```
## TIPO_ACCIDENTE ~ MES + HORA + DIASEMANA + PROVINCIA + TOT_VICTIMAS +
##     TOT_HERIDOS_LEVES + TOT_VEHICULOS_IMPLICADOS + CARRETERA +
##     TIPO_VIA + TRAZADO_NO_INTERSEC + TIPO_INTERSEC + ACOND_CALZADA +
##     PRIORIDAD + SUPERFICIE_CALZADA + LUMINOSIDAD + FACTORES_ATMOSFERICOS +
##     OTRA_CIRCUNSTANCIA + ACERAS + DENSIDAD_CIRCULACION + MEDIDAS_ESPECIALES
## <environment: 0x7fd6a064a920>
```

Obteniendo 20 características como importantes, no apareciendo: ANIO, COMUNIDAD_AUTONOMA, ISLA, TOT_MUERTOS, TOT_HERIDOS_GRAVES, ZONA, ZONA_AGRUPADA, RED_CARRETERA, VISIBILIDAD_RESTRINGIDA. Por lo tanto, ya que tenemos muchas características, vamos a ejecutar nuestro modelo. Quitando además CARRETERA al ralentizar los cálculos.

```
train.wrapper.consistency <- train.original[,c(2,3,4,5,8,11,12,17,18,19,20,21,22,23,24,26,27,28,29,30)]
test.wrapper.consistency <-  test.original[,c(2,3,4,5,8,11,12,17,18,19,20,21,22,23,24,26,27,28,29)]
```

9.1.13 Prueba del modelo

Hagamos por lo tanto una prueba. Quitando además CARRETERA para realizar los cálculos más rápidamente. Además, para que nuestro árbol funcione, he eliminado la variable HORA, junto con MEDIDAS_ESPECIAL, al contener esta un factor que en test no aparece.

```
train.wrapper.consistency <- train.wrapper.consistency[,-2]
test.wrapper.consistency <- test.wrapper.consistency[,-2]
train.wrapper.consistency <- train.wrapper.consistency[,-18]
test.wrapper.consistency <- test.wrapper.consistency[,-18]
set.seed(1234)
ct11 <- ctree(TIPO_ACCIDENTE ~., train.wrapper.consistency)
testPred11 <- predict(ct11, newdata = test.wrapper.consistency)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct11
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.11 <- as.matrix(testPred11)
salida.modelo.11 <- cbind(c(1:(dim(salida.modelo.11)[1])), salida.modelo.11)
```

```
colnames(salida.modelo.11) <- c("Id","Prediction")
write.table(salida.modelo.11,file="predicciones/Prediccion11.txt",sep=",",quote = F,row.names = F)
```

11 El resultado de este modelo para la competición de Kaggel, subido el 01/03/2017 a las 16:56, con un total de 26 personas entregadas, se ha quedado en la posición 15 con una puntuación del 0.82237. Empeorando un poco a la mejor actual.

9.1.14 Aproximación embedded: random.forest.importance

Método de cálculo de pesos de importancia de atributos calculados sobre un modelo construido usando el algoritmo RandomForest. Al no funcionar este método con variables que tengan más de 53 categorías hemos tenido que eliminar algunas variables como: CARRETERA y HORA. El tercer argumento con valor 1 significa la reducción en la media de fiabilidad predictiva.

```
train.embedded.random.forest.importance <- train.original[,c(1,2,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19
set.seed(1234)
pesos1 <- FSelector::random.forest.importance(TIPO_ACCIDENTE~.,train.embedded.random.forest.importance,
subset1 <- cutoff.k(pesos1,7)
los.7.mas.importantes.random.forest.importance1 <- as.simple.formula(subset1, "TIPO_ACCIDENTE")
los.7.mas.importantes.random.forest.importance1
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + PRIORIDAD + SUPERFICIE_CALZADA +
##     TIPO_INTERSEC + ZONA + ZONA_AGRUPADA + TIPO_VIA
## <environment: 0x7fd6850bc468>
```

El tercer argumento con valor 2 significa la reducción en la media de impureza de nodos.

```
set.seed(1234)
pesos2 <- FSelector::random.forest.importance(TIPO_ACCIDENTE~.,train.embedded.random.forest.importance,
subset2 <- cutoff.k(pesos2,7)
los.7.mas.importantes.random.forest.importance2 <- as.simple.formula(subset2, "TIPO_ACCIDENTE")
los.7.mas.importantes.random.forest.importance2
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS + PRIORIDAD + MES +
##     PROVINCIA + DIASEMANA + TIPO_INTERSEC + COMUNIDAD_AUTONOMA
## <environment: 0x7fd6a1ed9720>
```

Saliendo con ambos métodos las mismas variables como las más importantes.

```
train.embedded.random.forest.importance <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","PRIORIDAD","SU
test.embedded.random.forest.importance <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","PRIORIDAD","SUPE
```

9.1.15 Prueba del modelo

Hagamos por lo tanto una prueba. Quitando además CARRETERA para realizar los cálculos más rápidamente.

```
set.seed(1234)
ct12 <- ctree(TIPO_ACCIDENTE ~., train.embedded.random.forest.importance)
testPred12 <- predict(ct12, newdata = test.embedded.random.forest.importance)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct12
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.12 <- as.matrix(testPred12)
salida.modelo.12 <- cbind(c(1:(dim(salida.modelo.12)[1])), salida.modelo.12)
colnames(salida.modelo.12) <- c("Id","Prediction")
write.table(salida.modelo.12,file="predicciones/Prediccion12.txt",sep=",",quote = F,row.names = F)
```

12 El resultado de este modelo para la competición de Kaggel, subido el 01/03/2017 a las 16:32, con un total de 26 personas entregadas, se ha quedado en la posición 15 con una puntuación del 0.82326. Por lo que ha empeorado muy poco a la mejor obtenida por mi.

9.2 Paquete caret

Para este apartado comenzaremos con los dataset originales.

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
```

9.2.1 Esquema de valoración con aprendizaje de random forest

Probemos este paquete

```
train.caret <- train.sin.na[,-3]
set.seed(1234)
control <- caret::rfeControl(functions = rfFuncs, method = "cv", number = 10)
results <- caret::rfe(train.caret[,1:20], train.caret[,21],sizes=c(1:20),rfeControl=control)
```

```
##
## Attaching package: 'plyr'
## The following objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize
## The following object is masked from 'package:modeltools':
##
##     empty
```

```
results
```

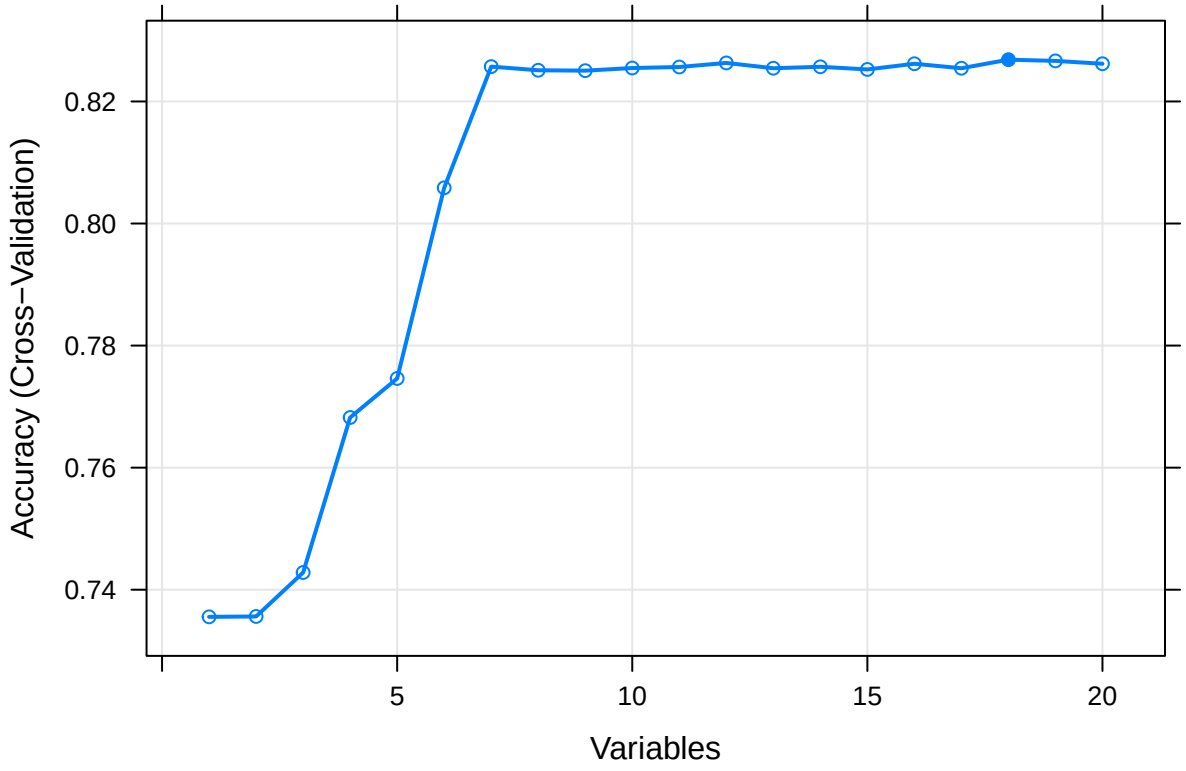
```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold)
##
## Resampling performance over subset size:
##
##  Variables Accuracy  Kappa AccuracySD  KappaSD Selected
##          1   0.7356 0.5476   0.001958 0.003753
##          2   0.7356 0.5488   0.002060 0.005799
##          3   0.7428 0.5633   0.008822 0.018186
##          4   0.7682 0.6103   0.003999 0.006967
##          5   0.7746 0.6198   0.003718 0.006957
##          6   0.8058 0.6746   0.024927 0.043455
```

```
##          7   0.8257 0.7092   0.004365 0.007473
##          8   0.8251 0.7080   0.004406 0.007606
##          9   0.8250 0.7084   0.004879 0.008312
##         10   0.8255 0.7091   0.004767 0.008148
##         11   0.8256 0.7094   0.005054 0.008533
##         12   0.8263 0.7104   0.004471 0.007590
##         13   0.8254 0.7090   0.004504 0.007600
##         14   0.8257 0.7093   0.004064 0.006927
##         15   0.8252 0.7086   0.004978 0.008451
##         16   0.8262 0.7106   0.004493 0.007700
##         17   0.8254 0.7093   0.004945 0.008465
##         18   0.8268 0.7114   0.004381 0.007590        *
##         19   0.8266 0.7110   0.004279 0.007404
##         20   0.8262 0.7098   0.004392 0.007524
##
## The top 5 variables (out of 18):
##    TOT_VEHICULOS_IMPLICADOS, SUPERFICIE_CALZADA, COMUNIDAD_AUTONOMA, TRAZADO_NO_INTERSEC, TIPO_INTERS
```

```
predictors(results)
```

```
##  [1] "TOT_VEHICULOS_IMPLICADOS" "SUPERFICIE_CALZADA"
##  [3] "COMUNIDAD_AUTONOMA"       "TRAZADO_NO_INTERSEC"
##  [5] "TIPO_INTERSEC"            "ZONA"
##  [7] "FACTORES_ATMOSFERICOS"    "ZONA_AGRUPADA"
##  [9] "TIPO_VIA"                 "RED_CARRETERA"
## [11] "TOT_HERIDOS_LEVES"        "LUMINOSIDAD"
## [13] "TOT_VICTIMAS"             "DIASEMANA"
## [15] "TOT_HERIDOS_GRAVES"       "ANIO"
## [17] "TOT_MUERTOS"              "MES"
```

```
plot(results, type=c("g","o"),lw=2)
```



Por lo tanto nos quedamos con las variables: TOT_VEHICULOS_IMPLICADOS, SUPERFICI_CALZADA, COMUNIDAD_AUTONOMA, TRAZADO_NO_INTERSEC, TIPO_INTERSEC, FACTORES_ATMOSFERICOS y ZONA.

```
train.caret <- train.caret[,c(11,18,5,16,17,20,12,21)]
test.caret <- test.sin.na[,c(12,19,6,17,18,21,13)]
```

9.2.2 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct13 <- ctree(TIPO_ACCIDENTE ~., train.caret)
testPred13 <- predict(ct13, newdata = test.caret)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct13
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.13 <- as.matrix(testPred13)
salida.modelo.13 <- cbind(c(1:(dim(salida.modelo.13)[1])), salida.modelo.13)
colnames(salida.modelo.13) <- c("Id","Prediction")
write.table(salida.modelo.13,file="predicciones/Prediccion13.txt",sep=",",quote = F,row.names = F)
```

13 El resultado de este modelo para la competición de Kaggel, subido el 03/03/2017 a las 12:44, con un total de 28 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82286. Por lo que ha empeorado muy poco a la mejor obtenida por mi.

9.3 Paquete Boruta

Para este apartado comenzaremos con los dataset originales.

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
```

9.3.1 Obtención de estadísticas sobre los atributos

```
set.seed(1234)
Bor.son <- Boruta(TIPO_ACCIDENTE~.,data=train.sin.na,doTrace = 2)
```

```
##  1. run of importance source...
##  2. run of importance source...
##  3. run of importance source...
##  4. run of importance source...
##  5. run of importance source...
##  6. run of importance source...
##  7. run of importance source...
##  8. run of importance source...
##  9. run of importance source...
##  10. run of importance source...
##  11. run of importance source...
##  12. run of importance source...
## After 12 iterations, +5.1 mins:
##  confirmed 19 attributes: ANIO, COMUNIDAD_AUTONOMA, DIASEMANA, FACTORES_ATMOSFERICOS, HORA and 14 mo
##  rejected 1 attribute: ISLA;
##  still have 1 attribute left.
##  13. run of importance source...
##  14. run of importance source...
##  15. run of importance source...
##  16. run of importance source...
##  17. run of importance source...
##  18. run of importance source...
##  19. run of importance source...
##  20. run of importance source...
##  21. run of importance source...
##  22. run of importance source...
```

```
##  23. run of importance source...
##  24. run of importance source...
##  25. run of importance source...
##  26. run of importance source...
##  27. run of importance source...
##  28. run of importance source...
##  29. run of importance source...
##  30. run of importance source...
##  31. run of importance source...
##  32. run of importance source...
##  33. run of importance source...
##  34. run of importance source...
##  35. run of importance source...
##  36. run of importance source...
##  37. run of importance source...
##  38. run of importance source...
##  39. run of importance source...
##  40. run of importance source...
##  41. run of importance source...
##  42. run of importance source...
## After 42 iterations, +17 mins:
##  confirmed 1 attribute: MES;
##  no more attributes left.
```

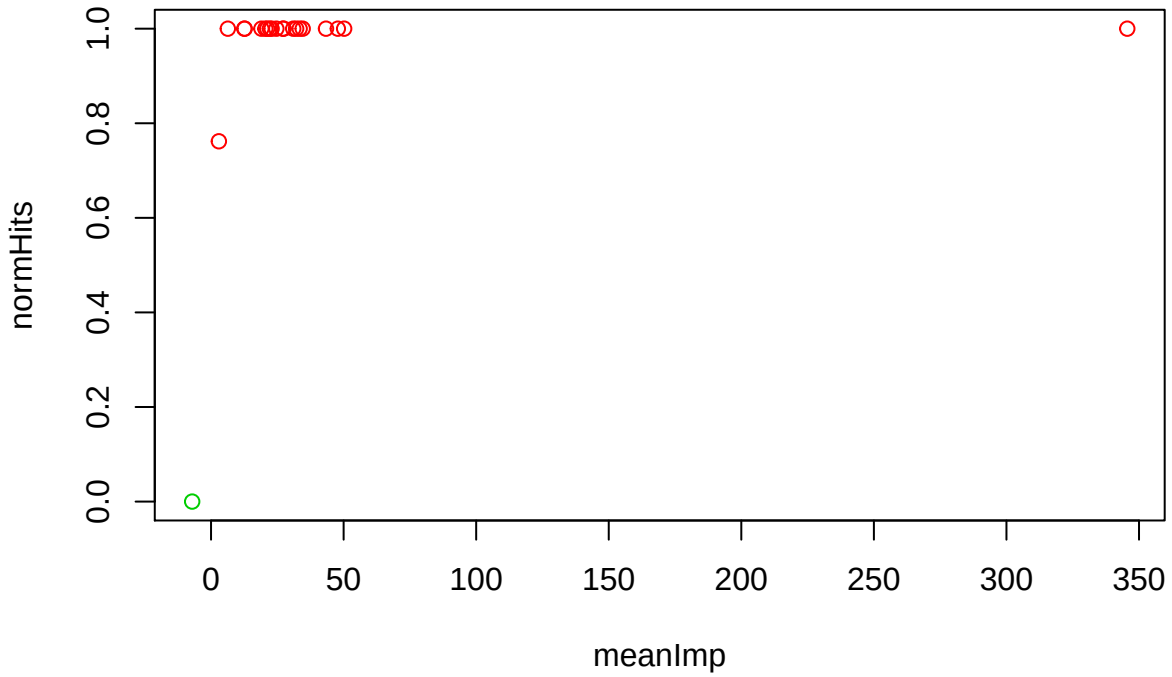
```
Bor.son
```

```
## Boruta performed 42 iterations in 17.36021 mins.
##  20 attributes confirmed important: ANIO, COMUNIDAD_AUTONOMA,
## DIASEMANA, FACTORES_ATMOSFERICOS, HORA and 15 more;
##  1 attributes confirmed unimportant: ISLA;
```

```
stats <- attStats(Bor.son)
```

```
stats
```

```
plot(normHits~meanImp, col=stats$decision, data=stats)
```




```
stats[stats$minImp > 20,]
```

```
##                             meanImp medianImp    minImp    maxImp normHits
## PROVINCIA                  24.62447  24.65554  21.62142  27.74662        1
## COMUNIDAD_AUTONOMA         32.09589  32.03774  29.12813  34.36039        1
## TOT_HERIDOS_LEVES          27.05657  27.00104  23.93419  31.27374        1
## TOT_VEHICULOS_IMPLICADOS 345.57459 347.17415 315.22617 368.69549        1
## ZONA                       34.56331  34.64699  31.92398  36.50529        1
## ZONA_AGRUPADA              31.00591  30.99580  29.11479  32.66536        1
## RED_CARRETERA              22.17482  22.17517  20.62672  24.82324        1
## TIPO_VIA                   27.39018  27.50940  24.88098  29.61594        1
## TRAZADO_NO_INTERSEC        47.81782  48.30661  44.46816  51.16734        1
## TIPO_INTERSEC              50.20120  50.11808  43.85470  55.09565        1
## SUPERFICIE_CALZADA         43.35270  43.80609  38.82273  47.76990        1
## FACTORES_ATMOSFERICOS      33.41960  33.48123  29.81075  36.64106        1
##                            decision
## PROVINCIA                 Confirmed
## COMUNIDAD_AUTONOMA        Confirmed
## TOT_HERIDOS_LEVES         Confirmed
## TOT_VEHICULOS_IMPLICADOS  Confirmed
## ZONA                      Confirmed
## ZONA_AGRUPADA             Confirmed
## RED_CARRETERA             Confirmed
## TIPO_VIA                  Confirmed
## TRAZADO_NO_INTERSEC       Confirmed
## TIPO_INTERSEC             Confirmed
## SUPERFICIE_CALZADA        Confirmed
## FACTORES_ATMOSFERICOS     Confirmed
```

Hemos visto como Boruta nos indica que el atributo que podemos quitar es ISLA, pero además he podado por minImp $<$ 20, quedándome con los atributos: PROVINCIA, COMUNIDAD_AUTONOMA, TOT_HERIDOS_LEVES, TOT_VEHICULOS_IMPLICADOS, ZONA, ZONA_AGRUPADA, RED_CARRETERA, TIPO_VIA, TRAZADO_NO_INTERSEC, TIPO_INTERSEC, SUPERFI-

CIE_CALZADA y FACTORES_ATMOSFERICOS. Vamos a probar un modelo con estas variables.

```
train.boruta <- train.sin.na[,c(5,6,11,12,13,14,15,16,17,18,19,21,22)]
test.boruta <- test.sin.na[,c(5,6,11,12,13,14,15,16,17,18,19,21)]
```

9.3.2 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct14 <- ctree(TIPO_ACCIDENTE ~., train.boruta)
testPred14 <- predict(ct14, newdata = test.boruta)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct14
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.14 <- as.matrix(testPred14)
salida.modelo.14 <- cbind(c(1:(dim(salida.modelo.14)[1])), salida.modelo.14)
colnames(salida.modelo.14) <- c("Id","Prediction")
write.table(salida.modelo.14,file="predicciones/Prediccion14.txt",sep=",",quote = F,row.names = F)
```

14 El resultado de este modelo para la competición de Kaggel, subido el 03/03/2017 a las 12:47, con un total de 28 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82326. Por lo que ha empeorado muy poco a la mejor obtenida por mi.

9.3.3 Combinación con random forest

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
```

```
train.boruta.random.forest <- train.sin.na[,-3]
set.seed(1234)
Bor.ran <- Boruta(TIPO_ACCIDENTE~.,data=train.boruta.random.forest,doTrace=2)
```

```
##  1. run of importance source...
##  2. run of importance source...
##  3. run of importance source...
##  4. run of importance source...
##  5. run of importance source...
##  6. run of importance source...
##  7. run of importance source...
##  8. run of importance source...
##  9. run of importance source...
##  10. run of importance source...
```

```
##  11. run of importance source...
## After 11 iterations, +4.5 mins:
##  confirmed 18 attributes: ANIO, COMUNIDAD_AUTONOMA, DIASEMANA, FACTORES_ATMOSFERICOS, LUMINOSIDAD an
##  rejected 1 attribute: ISLA;
##  still have 1 attribute left.
##  12. run of importance source...
##  13. run of importance source...
##  14. run of importance source...
##  15. run of importance source...
##  16. run of importance source...
##  17. run of importance source...
##  18. run of importance source...
##  19. run of importance source...
##  20. run of importance source...
##  21. run of importance source...
##  22. run of importance source...
##  23. run of importance source...
##  24. run of importance source...
##  25. run of importance source...
##  26. run of importance source...
##  27. run of importance source...
##  28. run of importance source...
##  29. run of importance source...
##  30. run of importance source...
##  31. run of importance source...
##  32. run of importance source...
##  33. run of importance source...
##  34. run of importance source...
##  35. run of importance source...
##  36. run of importance source...
##  37. run of importance source...
##  38. run of importance source...
##  39. run of importance source...
##  40. run of importance source...
##  41. run of importance source...
##  42. run of importance source...
```

```
##  43. run of importance source...
##  44. run of importance source...
##  45. run of importance source...
##  46. run of importance source...
##  47. run of importance source...
##  48. run of importance source...
##  49. run of importance source...
##  50. run of importance source...
##  51. run of importance source...
##  52. run of importance source...
##  53. run of importance source...
##  54. run of importance source...
##  55. run of importance source...
##  56. run of importance source...
##  57. run of importance source...
##  58. run of importance source...
##  59. run of importance source...
##  60. run of importance source...
##  61. run of importance source...
##  62. run of importance source...
##  63. run of importance source...
##  64. run of importance source...
##  65. run of importance source...
##  66. run of importance source...
##  67. run of importance source...
##  68. run of importance source...
##  69. run of importance source...
##  70. run of importance source...
##  71. run of importance source...
##  72. run of importance source...
##  73. run of importance source...
##  74. run of importance source...
##  75. run of importance source...
##  76. run of importance source...
##  77. run of importance source...
##  78. run of importance source...
```

```
##  79. run of importance source...
##  80. run of importance source...
##  81. run of importance source...
##  82. run of importance source...
##  83. run of importance source...
## After 83 iterations, +34 mins:
##  confirmed 1 attribute: MES;
##  no more attributes left.
```

```
model1 <- randomForest(TIPO_ACCIDENTE~.,data=train.boruta.random.forest)
model1
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.boruta.random.forest)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.36%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3060                 8                134   15
## Colision_Obstaculo        335                36                366   18
## Colision_Vehiculos          1                 0              16514    2
## Otro                      634                14                745   88
## Salida_Via                468                 3                461   18
## Vuelco                    354                12                190   13
##                     Salida_Via Vuelco  class.error
## Atropello                  420      5 0.1598023064
## Colision_Obstaculo         195      2 0.9621848739
## Colision_Vehiculos           3      0 0.0003631961
## Otro                       312     14 0.9513004981
## Salida_Via                5045     18 0.1609845335
## Vuelco                     449     50 0.9531835206
```

```
model2 <- randomForest(train.boruta.random.forest[,getSelectedAttributes(Bor.ran)],train.boruta.random.
model2
```

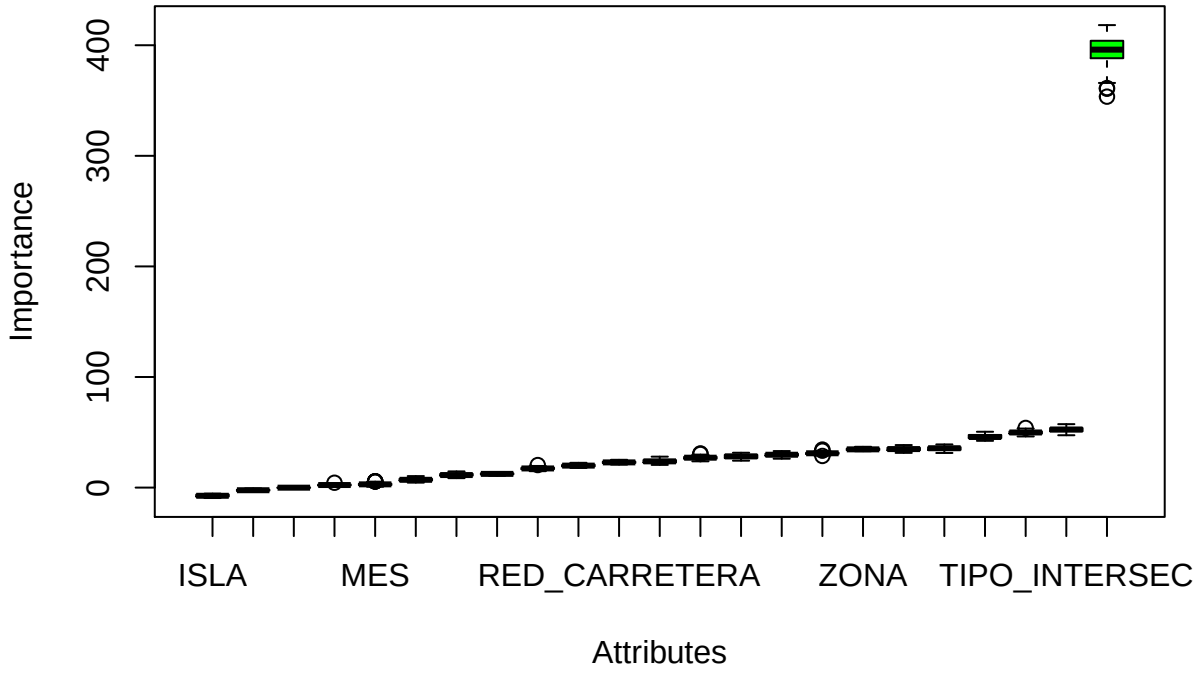
```
##
## Call:
##  randomForest(x = train.boruta.random.forest[, getSelectedAttributes(Bor.ran)],      y = train.boruta
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.38%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3055                 4                134   15
## Colision_Obstaculo        330                38                366   22
## Colision_Vehiculos          0                 0              16514    3
```

```
## Otro                    635                14                746   87
## Salida_Via              470                 1                460   15
## Vuelco                  349                12                190   15
##                    Salida_Via Vuelco  class.error
## Atropello                 427      7 0.1611751785
## Colision_Obstaculo        193      3 0.9600840336
## Colision_Vehiculos          3      0 0.0003631961
## Otro                      316      9 0.9518539015
## Salida_Via               5049     18 0.1603193082
## Vuelco                    456     46 0.9569288390
```

```
Bor.ran
```

```
## Boruta performed 83 iterations in 34.29272 mins.
##  19 attributes confirmed important: ANIO, COMUNIDAD_AUTONOMA,
## DIASEMANA, FACTORES_ATMOSFERICOS, LUMINOSIDAD and 14 more;
##  1 attributes confirmed unimportant: ISLA;
```

```
plot(Bor.ran)
```



Vemos como el atributo que nos indica que se puede eliminar es ISLA. Probemos el modelo sin esta variable.

```
train.boruta.random.forest <- train.boruta.random.forest[,-6]
test.boruta.random.forest <- test.sin.na[,-3]
test.boruta.random.forest <- test.boruta.random.forest[,-6]
```

9.3.4 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct15 <- ctree(TIPO_ACCIDENTE ~., train.boruta.random.forest)
testPred15 <- predict(ct15, newdata = test.boruta.random.forest)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct15
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.15 <- as.matrix(testPred15)
salida.modelo.15 <- cbind(c(1:(dim(salida.modelo.15)[1])), salida.modelo.15)
colnames(salida.modelo.15) <- c("Id","Prediction")
write.table(salida.modelo.15,file="predicciones/Prediccion15.txt",sep=",",quote = F,row.names = F)
```

15 El resultado de este modelo para la competición de Kaggel, subido el 03/03/2017 a las 12:51, con un total de 28 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82286. Por lo que ha empeorado muy poco a la mejor obtenida por mi.

10 Detección de ruido

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
```

Vamos a probar este método.

```
out <- IPF(TIPO_ACCIDENTE~., data=train.original,s=2)
```

```
## Iteration 1: 305 noisy instances removed
```

```
## Iteration 2: 10 noisy instances removed
```

```
## Iteration 3: 0 noisy instances removed
```

```
#summary(out, explicit=TRUE)
identical(out$cleanData, train.original[setdiff(1:nrow(train.original),out$remIdx),])
```

```
## [1] TRUE
```

```
train.sin.ruido1 <- train.original[setdiff(1:nrow(train.original),out$remIdx),]
```

```
set.seed(1234)
out <- IPF(TIPO_ACCIDENTE~., data=train.sin.ruido1,s=2)
```

```
## Iteration 1: 313 noisy instances removed
```

```
## Iteration 2: 0 noisy instances removed
```

```
## Iteration 3: 0 noisy instances removed
```

```
#summary(out, explicit=TRUE)
train.sin.ruido2 <- train.sin.ruido1[setdiff(1:nrow(train.sin.ruido1),out$remIdx),]
```

```
set.seed(1234)
out <- IPF(TIPO_ACCIDENTE~., data=train.sin.ruido2,s=2)
```

```
## Iteration 1: 300 noisy instances removed
```

```
## Iteration 2: 8 noisy instances removed
```

```
## Iteration 3: 0 noisy instances removed
```

```
#summary(out, explicit=TRUE)
train.sin.ruido3 <- train.sin.ruido2[setdiff(1:nrow(train.sin.ruido2),out$remIdx),]
```

```
set.seed(1234)
out <- IPF(TIPO_ACCIDENTE~., data=train.sin.ruido3,s=2)
```

```
## Iteration 1: 297 noisy instances removed
```

```
## Iteration 2: 4 noisy instances removed
```

```
## Iteration 3: 2 noisy instances removed
```

```
#summary(out, explicit=TRUE)
train.sin.ruido4 <- train.sin.ruido3[setdiff(1:nrow(train.sin.ruido3),out$remIdx),]
```

```
set.seed(1234)
out <- IPF(TIPO_ACCIDENTE~., data=train.sin.ruido4,s=2)
```

```
## Iteration 1: 293 noisy instances removed
```

```
## Iteration 2: 5 noisy instances removed
```

```
## Iteration 3: 0 noisy instances removed
```

```
#summary(out, explicit=TRUE)
train.sin.ruido5 <- train.sin.ruido4[setdiff(1:nrow(train.sin.ruido4),out$remIdx),]
```

```
numero.de.eliminadas <- nrow(train.original)-nrow(train.sin.ruido5)
numero.de.eliminadas
```

```
## [1] 1537
```

Por lo que se han eliminado en total 1537 instancias al ser ruido, pero podrían ser más o menos, depende del número de pasadas que se decidan hacer.

Hagamos ahora un selección de características, con el mejor método hasta ahora, cfs, para crear un modelo y realizar una prueba.

```
set.seed(1234)
subset <- FSelector::cfs(TIPO_ACCIDENTE~.,train.sin.ruido5)
el.mejor.segun.cfs <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.cfs
```

```
## TIPO_ACCIDENTE ~ TOT_VEHICULOS_IMPLICADOS
## <environment: 0x7ff8d53410a8>
```

Siendo TOT_VEHICULOS_IMPLICADOS, la mejor característica, repitamos esto para obtener las 5 mejores, de forma que eliminamos la que mejor se obtiene.

```
set.seed(1234)
train.wrapper.cfs = train.sin.ruido5[,-12]
subset <- FSelector::cfs(TIPO_ACCIDENTE~.,train.wrapper.cfs)
```

```
el.mejor.segun.cfs <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.cfs
```

```
## TIPO_ACCIDENTE ~ ZONA_AGRUPADA + CARRETERA + TRAZADO_NO_INTERSEC
## <environment: 0x7ff8d0b4e4d0>
```

Ahora obtenemos ZONA_AGRUPADA, CARRETERA y TRAZADO_NO_INTERSEC.

```
set.seed(1234)
train.wrapper.cfs = train.sin.ruido5[,-c(12,14,15,18)]
subset <- FSelector::cfs(TIPO_ACCIDENTE~.,train.wrapper.cfs)
```

```
el.mejor.segun.cfs <- as.simple.formula(subset, "TIPO_ACCIDENTE")
el.mejor.segun.cfs
```

```
## TIPO_ACCIDENTE ~ TOT_HERIDOS_LEVES + ZONA + RED_CARRETERA + TIPO_VIA +
##     TIPO_INTERSEC + PRIORIDAD + SUPERFICIE_CALZADA + ACERAS
## <environment: 0x7ff8d0951c40>
```

Obteniendo: TOT_HERIDOS_LEVES, ZONA, RED_CARRETERA, TIPO_VIA, TIPO_INTERSEC, PRIORIDAD, SUPERFICIE_CALZADA y ACERAS.

Por lo tanto, ya que tenemos muchas características, vamos a probar el modelo con todas, salvo CARRETERA.

```
train.deteccion.de.ruido <- train.sin.ruido5[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_
test.deteccion.de.ruido <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_INTE
```

Vamos a probar este modelo.

10.0.1 Prueba del modelo

Hagamos por lo tanto una prueba.

```
set.seed(1234)
ct16 <- ctree(TIPO_ACCIDENTE ~., train.deteccion.de.ruido)
testPred16 <- predict(ct16, newdata = test.deteccion.de.ruido)
```

Por lo que ya tenemos el conjunto de test predicho. Además el árbol creado tendría la siguiente estructura:

```
#ct16
```

Vamos a escribir la salida del modelo para ver su puntuación en Kaggel.

```
salida.modelo.16 <- as.matrix(testPred16)
salida.modelo.16 <- cbind(c(1:(dim(salida.modelo.16)[1])), salida.modelo.16)
colnames(salida.modelo.16) <- c("Id","Prediction")
write.table(salida.modelo.16,file="predicciones/Prediccion16.txt",sep=",",quote = F,row.names = F)
```

16 El resultado de este modelo para la competición de Kaggel, subido el 03/03/2017 a las 13:00, con un total de 28 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82365. Por lo que ha empeorado muy poco a la mejor obtenida por mi, cosa que no era de esperar ya que es el mismo modelo que el mejor obtenido, pero eliminando instancias de ruido.

11 Modelo SVM Radial

Limpiamos el espacio de trabajo

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
train.mejor.resultado.sin.na <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO
test.mejor.resultado.sin.na <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_
```

Vamos a aplicar el modelo SVM Radial a nuestros datos.

```
set.seed(1234)
modelo.svm.radial <- train(TIPO_ACCIDENTE ~., data = train.mejor.resultado.sin.na, methods = "svmRadial
predicciciones.svm.radial <- predict(modelo.svm.radial, test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.17 <- as.matrix(predicciciones.svm.radial)
salida.modelo.17 <- cbind(c(1:(dim(salida.modelo.17)[1])), salida.modelo.17)
colnames(salida.modelo.17) <- c("Id","Prediction")
write.table(salida.modelo.17,file="predicciones/Prediccion17.txt",sep=",",quote = F,row.names = F)
```

17 El resultado de este modelo para la competición de Kaggel, subido el 04/03/2017 a las 16:28, con un total de 29 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.81762.

Si lo que queremos es preprocesar los datos con un centrado y escalado

```
set.seed(1234)
modelo.svm.radial2 <- train(TIPO_ACCIDENTE ~., data = train.mejor.resultado.sin.na, methods = "svmRadial
predicciciones.svm.radial2 <- predict(modelo.svm.radial2, test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.18 <- as.matrix(predicciciones.svm.radial2)
salida.modelo.18 <- cbind(c(1:(dim(salida.modelo.18)[1])), salida.modelo.18)
colnames(salida.modelo.18) <- c("Id","Prediction")
write.table(salida.modelo.18,file="predicciones/Prediccion18.txt",sep=",",quote = F,row.names = F)
```

18 El resultado de este modelo para la competición de Kaggel, subido el 04/03/2017 a las 17:43, con un total de 29 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.81822.

Este método tiene un parámetro de coste que regula el coste asociado a los errores de predicción: las diferencias entre el valor predicho y el real. Es posible evaluar diferentes valores de coste directamente:

```
set.seed(1234)
modelo.svm.radial3 <- train(TIPO_ACCIDENTE ~., data = train.mejor.resultado.sin.na, methods = "svmRadial
predicciciones.svm.radial3 <- predict(modelo.svm.radial3, test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.19 <- as.matrix(predicciciones.svm.radial3)
salida.modelo.19 <- cbind(c(1:(dim(salida.modelo.19)[1])), salida.modelo.19)
colnames(salida.modelo.19) <- c("Id","Prediction")
write.table(salida.modelo.19,file="predicciones/Prediccion19.txt",sep=",",quote = F,row.names = F)
```

19 El resultado de este modelo para la competición de Kaggel, subido el 05/03/2017 a las 12:15, con un total de 31 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82256.

También se puede modificar esta llamada para que se utilicen diferentes particionados. Realicemos 2 repeticiones de validación cruzada con k = 10.

```
set.seed(1234)
modelo.svm.radial4 <- train(TIPO_ACCIDENTE ~., data = train.mejor.resultado.sin.na, methods = "svmRadial
predicciciones.svm.radial4 <- predict(modelo.svm.radial4, test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.20 <- as.matrix(predicciciones.svm.radial4)
salida.modelo.20 <- cbind(c(1:(dim(salida.modelo.20)[1])), salida.modelo.20)
```

```
colnames(salida.modelo.20) <- c("Id","Prediction")
write.table(salida.modelo.20,file="predicciones/Prediccion20.txt",sep=",",quote = F,row.names = F)
```

20 El resultado de este modelo para la competición de Kaggel, subido el 05/03/2017 a las 12:17, con un total de 31 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82237.

12 SVM (Support Vector Machine)

Limpiamos el espacio de trabajo

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
train.mejor.resultado.sin.na <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO
test.mejor.resultado.sin.na <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_
```

Vamos a probar esta técnica:

```
set.seed(1234)
modelo.svm <- e1071::svm(TIPO_ACCIDENTE~., data=train.mejor.resultado.sin.na, method="C-classification"
prediccion.svm <- predict(modelo.svm, test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.21 <- as.matrix(prediccion.svm)
salida.modelo.21 <- cbind(c(1:(dim(salida.modelo.21)[1])), salida.modelo.21)
colnames(salida.modelo.21) <- c("Id","Prediction")
write.table(salida.modelo.21,file="predicciones/Prediccion21.txt",sep=",",quote = F,row.names = F)
```

21 El resultado de este modelo para la competición de Kaggel, subido el 05/03/2017 a las 12:26, con un total de 31 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.82296.

13 Métodos ensamble de construcción de conjuntos de modelos

13.1 Bagging

Limpiamos el espacio de trabajo

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
train.mejor.resultado.sin.na <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO
test.mejor.resultado.sin.na <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_
```

```
set.seed(1234)
modelo.bagging.1 <- adabag::bagging(TIPO_ACCIDENTE~.,data=train.mejor.resultado.sin.na, control=rpart::
predicciones.bagging.1 <- adabag::predict.bagging(modelo.bagging.1,newdata=test.mejor.resultado.sin.na)

set.seed(1234)
```

```
modelo.bagging.2 <- adabag::bagging(TIPO_ACCIDENTE~.,data=train.mejor.resultado.sin.na,mfinal=20 ,contr
predicciones.bagging.2 <- adabag::predict.bagging(modelo.bagging.2,newdata=test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.22 <- as.matrix(predicciones.bagging.1$class)
salida.modelo.22 <- cbind(c(1:(dim(salida.modelo.22)[1])), salida.modelo.22)
colnames(salida.modelo.22) <- c("Id","Prediction")
write.table(salida.modelo.22,file="predicciones/Prediccion22.txt",sep=",",quote = F,row.names = F)
```

22 El resultado de este modelo para la competición de Kaggel, subido el 05/03/2017 a las 12:31, con un total de 31 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.81891.

Veamos la puntuación en kaggel

```
salida.modelo.23 <- as.matrix(predicciones.bagging.2$class)
salida.modelo.23 <- cbind(c(1:(dim(salida.modelo.23)[1])), salida.modelo.23)
colnames(salida.modelo.23) <- c("Id","Prediction")
write.table(salida.modelo.23,file="predicciones/Prediccion23.txt",sep=",",quote = F,row.names = F)
```

23 El resultado de este modelo para la competición de Kaggel, subido el 05/03/2017 a las 12:37, con un total de 31 personas entregadas, se ha quedado en la posición 16 con una puntuación del 0.81891.

13.2 Random forest

Limpiamos el espacio de trabajo

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
train.mejor.resultado.sin.na <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_N
test.mejor.resultado.sin.na <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_
```

```
set.seed(1234)
modelo.random.forest.10 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.mejor.resultado.sin.na
print(modelo.random.forest.10)
```

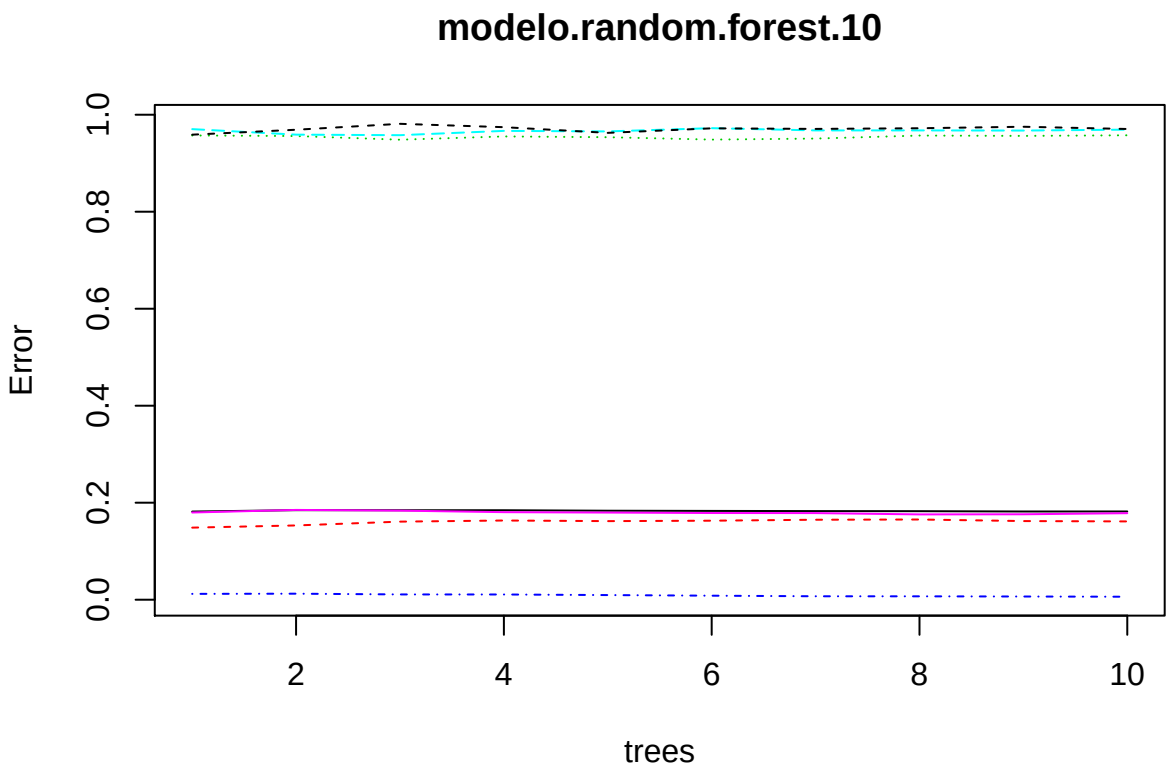
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.mejor.resultado.sin.na,      ntree = 10)
##                Type of random forest: classification
##                      Number of trees: 10
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 18.19%
## Confusion matrix:
##                    Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello               3028                 6                134   18
## Colision_Obstaculo       330                40                353   14
## Colision_Vehiculos         8                18              16267   26
## Otro                     646                24                731   55
## Salida_Via               504                21                443   55
## Vuelco                   351                 7                179   36
##                    Salida_Via Vuelco class.error
```

```
## Atropello                 418      7 0.161451122
## Colision_Obstaculo        200      7 0.957627119
## Colision_Vehiculos         40      9 0.006170577
## Otro                      305     27 0.969239374
## Salida_Via               4886     38 0.178409282
## Vuelco                    450     31 0.970588235
```

```
randomForest::importance(modelo.random.forest.10)
```

```
##                           MeanDecreaseGini
## TOT_VEHICULOS_IMPLICADOS        7949.0655
## ZONA_AGRUPADA                    921.2303
## TRAZADO_NO_INTERSEC              325.9847
## TOT_HERIDOS_LEVES                229.3772
## ZONA                             744.0811
## RED_CARRETERA                    127.7767
## TIPO_VIA                         359.3147
## TIPO_INTERSEC                    270.0264
## SUPERFICIE_CALZADA               245.3562
```

```
plot(modelo.random.forest.10)
```




```
predicciones.rf.10 <- predict(modelo.random.forest.10,newdata=test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.24 <- as.matrix(predicciones.rf.10)
salida.modelo.24 <- cbind(c(1:(dim(salida.modelo.24)[1])), salida.modelo.24)
colnames(salida.modelo.24) <- c("Id","Prediction")
write.table(salida.modelo.24,file="predicciones/Prediccion24.txt",sep=",",quote = F,row.names = F)
```

24 El resultado de este modelo para la competición de Kaggel, subido el 06/03/2017 a las 10:54, con un total

de 32 personas entregadas, se ha quedado en la posición 18 con una puntuación del 0.82089.

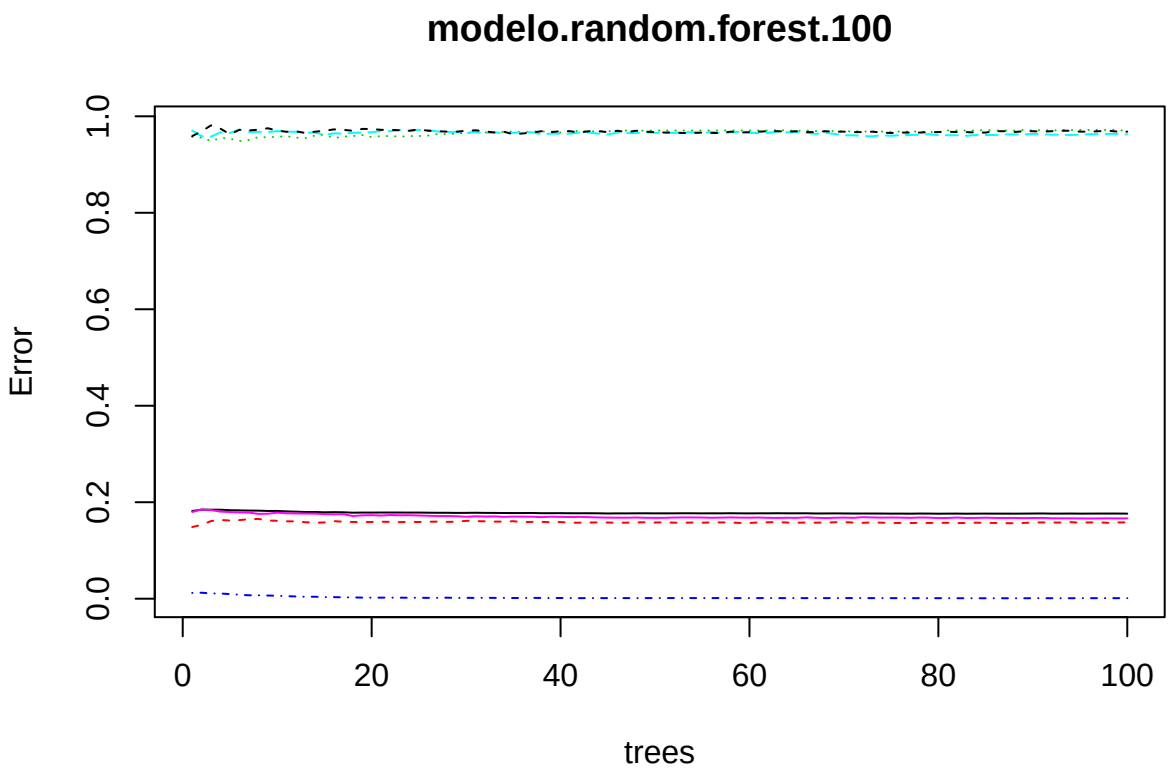
```
set.seed(1234)
modelo.random.forest.100 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.mejor.resultado.sin.
print(modelo.random.forest.100)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.mejor.resultado.sin.na,      ntree = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.64%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3066                  3                 132   11
## Colision_Obstaculo        324                 29                 361   27
## Colision_Vehiculos          1                  1               16502    4
## Otro                      638                 16                 751   68
## Salida_Via                478                  6                 459   28
## Vuelco                    346                  8                 188   40
##                     Salida_Via Vuelco class.error
## Atropello                  427      3 0.158154860
## Colision_Obstaculo         209      2 0.969537815
## Colision_Vehiculos          12      0 0.001089588
## Otro                       321     13 0.962368567
## Salida_Via                5012     30 0.166472643
## Vuelco                     452     34 0.968164794
```

```
randomForest::importance(modelo.random.forest.100)
```

```
##                           MeanDecreaseGini
## TOT_VEHICULOS_IMPLICADOS        8066.3529
## ZONA_AGRUPADA                    526.6196
## TRAZADO_NO_INTERSEC              401.7404
## TOT_HERIDOS_LEVES                228.8489
## ZONA                             634.5463
## RED_CARRETERA                    202.4311
## TIPO_VIA                         494.7834
## TIPO_INTERSEC                    285.8930
## SUPERFICIE_CALZADA               267.8467
```

```
plot(modelo.random.forest.100)
```




```
predicciones.rf.100 <- predict(modelo.random.forest.100,newdata=test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.25 <- as.matrix(predicciones.rf.100)
salida.modelo.25 <- cbind(c(1:(dim(salida.modelo.25)[1])), salida.modelo.25)
colnames(salida.modelo.25) <- c("Id","Prediction")
write.table(salida.modelo.25,file="predicciones/Prediccion25.txt",sep=",",quote = F,row.names = F)
```

25 El resultado de este modelo para la competición de Kaggel, subido el 06/03/2017 a las 10:56, con un total de 32 personas entregadas, se ha quedado en la posición 18 con una puntuación del 0.82306.

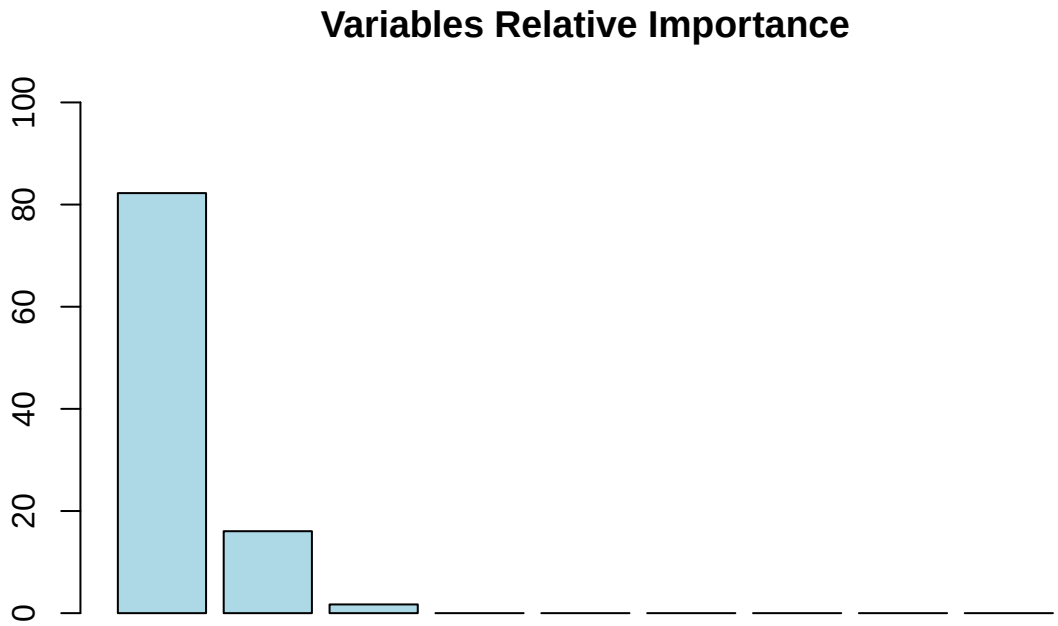
13.3 Boosting

Limpiamos el espacio de trabajo

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
train.sin.na <- train.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24,30)]
test.sin.na <- test.original[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,22,23,24)]
train.mejor.resultado.sin.na <- train.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_N
test.mejor.resultado.sin.na <- test.original[,c("TOT_VEHICULOS_IMPLICADOS","ZONA_AGRUPADA","TRAZADO_NO_
```

Vamos a probar este método:

```
set.seed(1234)
modelo.boosting <- adabag::boosting(TIPO_ACCIDENTE~.,data=train.mejor.resultado.sin.na, mfinal = 10, co
barplot(modelo.boosting$imp[order(modelo.boosting$imp, decreasing = TRUE)],
          ylim = c(0, 100), main = "Variables Relative Importance",
          col = "lightblue")
```




```
prediccion.boosting <- predict.boosting(modelo.boosting, newdata = test.mejor.resultado.sin.na)
modelo.boosting$importance[modelo.boosting$importance>0]
```

```
## TOT_VEHICULOS_IMPLICADOS                     ZONA            ZONA_AGRUPADA
##                82.246757                 1.709175                16.044068
```

Veamos la puntuación en kaggel

```
salida.modelo.26 <- as.matrix(prediccion.boosting$class)
salida.modelo.26 <- cbind(c(1:(dim(salida.modelo.26)[1])), salida.modelo.26)
colnames(salida.modelo.26) <- c("Id","Prediction")
write.table(salida.modelo.26,file="predicciones/Prediccion26.txt",sep=",",quote = F,row.names = F)
```

26 El resultado de este modelo para la competición de Kaggel, subido el 06/03/2017 a las 10:58, con un total de 32 personas entregadas, se ha quedado en la posición 18 con una puntuación del 0.81891.

14 Arboles de clasificación

Aunque todos los modelos que he realizado en este guión han sido con este método, voy a realizarlo de nuevo para ver su puntuación y compararlo con los métodos anteriores, ya que he usado otro conjunto de variables.

```
set.seed(1234)
ct27 <- ctree(TIPO_ACCIDENTE ~., train.mejor.resultado.sin.na)
testPred27 <- predict(ct27, newdata = test.mejor.resultado.sin.na)
```

Veamos la puntuación en kaggel

```
salida.modelo.27 <- as.matrix(testPred27)
salida.modelo.27 <- cbind(c(1:(dim(salida.modelo.27)[1])), salida.modelo.27)
colnames(salida.modelo.27) <- c("Id","Prediction")
write.table(salida.modelo.27,file="predicciones/Prediccion27.txt",sep=",",quote = F,row.names = F)
```

27 El resultado de este modelo para la competición de Kaggel, subido el 06/03/2017 a las 10:59, con un total

de 32 personas entregadas, se ha quedado en la posición 18 con una puntuación del 0.82000.

15 Primeras Conclusiones

Lo realizado hasta ahora ha sido lo visto durante el guión de preprocesamiento, reproduciendo lo visto en clase. A partir de ahora, voy a realizar nuevos modelos, usando lo visto hasta ahora, viendo como podemos conseguir el mejor modelo posible. Además el modelo que mejor resultado ha obtenido ha sido RandomForest.

16 Prueba de modelos

16.1 Prueba del primer modelo random forest 1000

Como ya vimos, las variables con valores perdidos son: CARRETERA, ACOND_CALZADA, PRIORIDAD, VISIBILIDAD_RESTRINGIDA, OTRA_CIRCUNSTANCIA, ACERAS, DENSIDAD_CIRCULACION y MEDIDAS_ESPECIALES. Además, a modo de resumen, voy a ver las variables que, en selección de características, tenían mayor importancia:

Variable	Número de veces que ha sido seleccionada
ANIO	2
MES	3
HORA	2
DIASEMANA	2
PROVINCIA	4
COMUNIDAD_AUTONOMA	4
TOT_VICTIMAS	2
TOT_MUERTOS	1
TOT_HERIDOS_LEVES	3
TOT_VEHICULOS_IMPLICADOS	11
ZONA	8
ZONA_AGRUPADA	8
CARRETERA	6
RED_CARRETERA	5
TIPO_VIA	7
TRAZADO_NO_INTERSEC	7
TIPO_INTERSEC	6
ACOND_CALZADA	1
PRIORIDAD	7
SUPERFICE_CALZADA	5
LUMINOSIDAD	2
FACTORES_ATMOSFERICOS	3
OTRA_CIRCUNSTANCIA	1
ACERAS	4
DENSIDAD_CIRCULACION	2
MEDIDAS_ESPECIALES	1

Siendo, según esta clasificación las más importantes: 11 (TOT_VEHICULOS_IMPLICADOS), 8 (ZONA, ZONA_AGRUPADA), 7 (TIPO_VIA, TRAZADO_NO_INTERSEC,PRIORIDAD) Además, viendo como las variables CARRETERA y HORA, tienen más de 53 factores, vamos discretizarlas en menos valores. Por ejemplo, vemos la variable HORA actualmente:

```
rm(list=ls())
train.original <- read.csv("accidentes-kaggle.csv")
test.original <- read.csv("accidentes-kaggle-test.csv")
valores.hora <- train.original[,3]
valores.hora.character <- as.character(valores.hora)
valores.hora.cortados <- unlist(lapply(valores.hora.character,function(x) strtrim(x,2)))
valores.hora.factor <- as.factor(valores.hora.cortados)
levels(valores.hora.factor) <- c("0", "0", "1","1","10","11","12","13","14","15","16","17","18","19","2
train.modelos <- train.original
train.modelos$HORA = valores.hora.factor
```

```
valores.hora <- test.original[,3]
valores.hora.character <- as.character(valores.hora)
valores.hora.cortados <- unlist(lapply(valores.hora.character,function(x) strtrim(x,2)))
valores.hora.factor <- as.factor(valores.hora.cortados)
levels(valores.hora.factor) <- c("0", "0", "1","1","10","11","12","13","14","15","16","17","18","19","2
test.modelos <- test.original
test.modelos$HORA = valores.hora.factor
```

Además, viendo que Carretera tiene 3268 niveles, y que no es fácilmente discretizable, voy a eliminarla del conjunto de datos.

```
train.modelos$CARRETERA = NULL
test.modelos$CARRETERA = NULL
```

Como tenemos varias variables que tienen valores perdidos, vamos a imputar estos valores.

```
set.seed(1234)
train.modelos.imputados <- mice::mice(train.modelos,m=5,method="pmm")
train.imputados <- mice::complete(train.modelos.imputados)
set.seed(1234)
test.modelos.imputados <- mice::mice(test.modelos,m=5,method="pmm")
test.imputados <- mice::complete(test.modelos.imputados)
```

Escribimos los datos imputados en un csv para que su carga sea mucho más rápida:

```
write.csv(train.imputados,"datasetmodificados/train-imputados.csv")
write.csv(test.imputados,"datasetmodificados/test-imputados.csv")
```

Vamos a ver que variables son las más importantes según random forest y boosting: Primero con RandomForest.

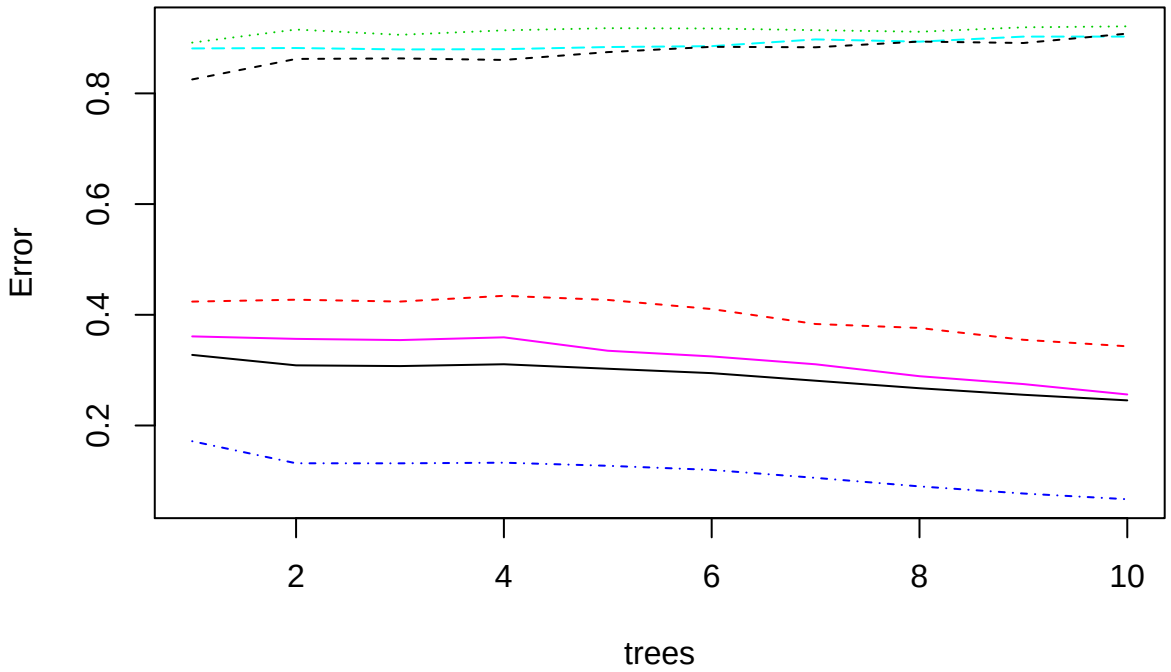
```
set.seed(1234)
modelo.random.forest.10 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados,ntree=10)
print(modelo.random.forest.10)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados,      ntree = 10)
##                Type of random forest: classification
##                      Number of trees: 10
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 24.54%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                2375                169                140  237
## Colision_Obstaculo        198                 74                324   64
## Colision_Vehiculos         68                230              15269  432
## Otro                      456                 75                663  174
## Salida_Via                449                136                449  230
## Vuelco                    243                 48                173   72
##                     Salida_Via Vuelco class.error
## Atropello                  506    188  0.34301521
## Colision_Obstaculo         224     54  0.92110874
## Colision_Vehiculos         252    110  0.06674409
## Otro                       343     76  0.90263011
## Salida_Via                4423    258  0.25601346
## Vuelco                     421     97  0.90796964
```

```
randomForest::importance(modelo.random.forest.10)>100
```

```
##                            MeanDecreaseGini
## ANIO                                   TRUE
## MES                                    TRUE
## HORA                                   TRUE
## DIASEMANA                              TRUE
## PROVINCIA                              TRUE
## COMUNIDAD_AUTONOMA                     TRUE
## ISLA                                  FALSE
## TOT_VICTIMAS                           TRUE
## TOT_MUERTOS                           FALSE
## TOT_HERIDOS_GRAVES                     TRUE
## TOT_HERIDOS_LEVES                      TRUE
## TOT_VEHICULOS_IMPLICADOS               TRUE
## ZONA                                   TRUE
## ZONA_AGRUPADA                          TRUE
## RED_CARRETERA                          TRUE
## TIPO_VIA                               TRUE
## TRAZADO_NO_INTERSEC                    TRUE
## TIPO_INTERSEC                          TRUE
## ACOND_CALZADA                          TRUE
## PRIORIDAD                              TRUE
## SUPERFICIE_CALZADA                     TRUE
## LUMINOSIDAD                            TRUE
## FACTORES_ATMOSFERICOS                  TRUE
## VISIBILIDAD_RESTRINGIDA                TRUE
## OTRA_CIRCUNSTANCIA                     TRUE
## ACERAS                                 TRUE
## DENSIDAD_CIRCULACION                   TRUE
## MEDIDAS_ESPECIALES                    FALSE
```

```
plot(modelo.random.forest.10)
```



Por lo que las variables menos importantes serían: ISLA, TOT_MUERTOS y MEDIDAS_ESPECIALES. Vamos a probar un nuevo árbol sin estas variables.

```
train.imputados2 <- train.imputados
test.imputados2 <- test.imputados
train.imputados2$ISLA=NULL
train.imputados2$TOT_MUERTOS = NULL
train.imputados2$MEDIDAS_ESPECIALES =NULL
test.imputados2$ISLA=NULL
test.imputados2$TOT_MUERTOS = NULL
test.imputados2$MEDIDAS_ESPECIALES =NULL
set.seed(1234)
modelo.random.forest.50 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados2,ntree=50)
print(modelo.random.forest.50)
```

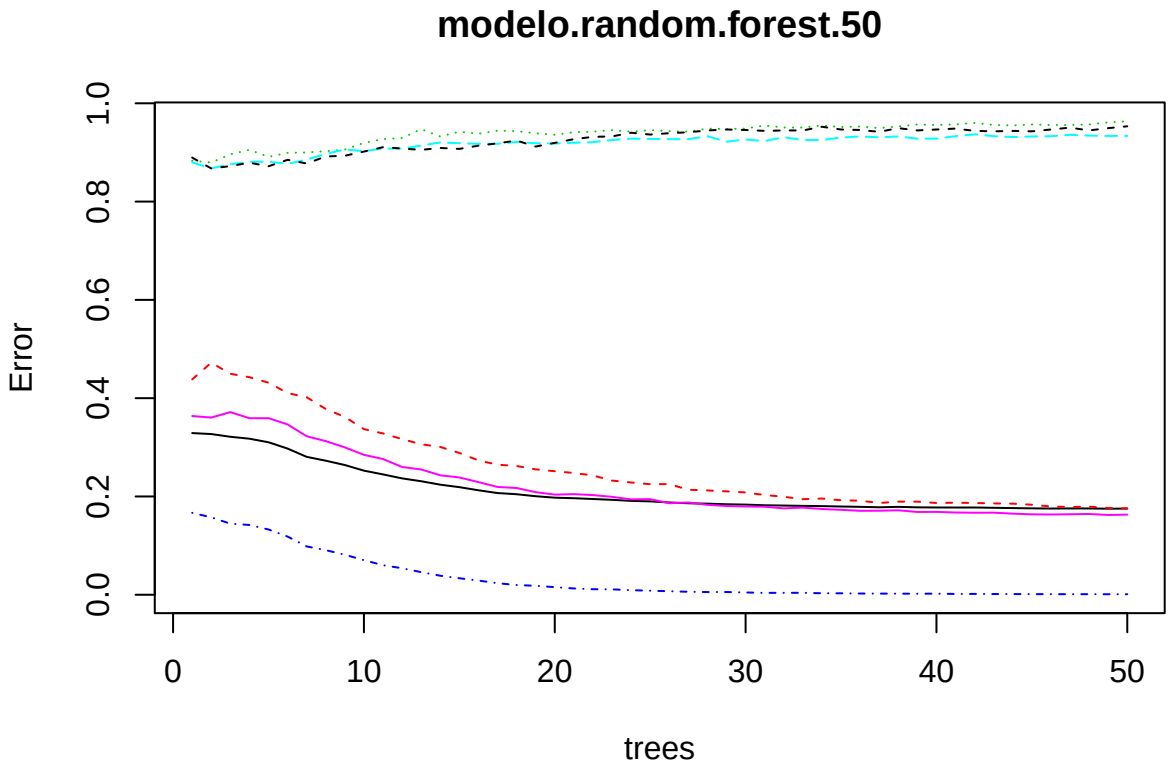
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados2,      ntree = 50)
##                Type of random forest: classification
##                      Number of trees: 50
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 17.53%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3002                12                133   68
## Colision_Obstaculo        286                35                364   39
## Colision_Vehiculos          2                 1              16503    7
## Otro                      571                16                739  120
## Salida_Via                411                12                457   51
## Vuelco                    308                 8                190   33
```

```
##                     Salida_Via Vuelco class.error
## Atropello                  410     17 0.175727622
## Colision_Obstaculo         216     12 0.963235294
## Colision_Vehiculos           7      0 0.001029056
## Otro                       335     26 0.933591588
## Salida_Via                5033     49 0.162980210
## Vuelco                     479     50 0.953183521
```

```
randomForest::importance(modelo.random.forest.50)
```

```
##                            MeanDecreaseGini
## ANIO                               596.2095
## MES                               1101.3421
## HORA                              1312.8801
## DIASEMANA                          777.0993
## PROVINCIA                          905.6392
## COMUNIDAD_AUTONOMA                 591.1622
## TOT_VICTIMAS                       186.7373
## TOT_HERIDOS_GRAVES                 142.5554
## TOT_HERIDOS_LEVES                  270.0019
## TOT_VEHICULOS_IMPLICADOS          7551.3066
## ZONA                               450.0445
## ZONA_AGRUPADA                      406.3041
## RED_CARRETERA                      394.8192
## TIPO_VIA                           406.7617
## TRAZADO_NO_INTERSEC                493.2030
## TIPO_INTERSEC                      407.9113
## ACOND_CALZADA                      378.2436
## PRIORIDAD                          589.5785
## SUPERFICIE_CALZADA                 273.2826
## LUMINOSIDAD                        328.1767
## FACTORES_ATMOSFERICOS              197.5519
## VISIBILIDAD_RESTRINGIDA            176.3192
## OTRA_CIRCUNSTANCIA                 151.8418
## ACERAS                             141.1512
## DENSIDAD_CIRCULACION               155.6430
```

```
plot(modelo.random.forest.50)
```



Tenemos varias variables que podrían ser eliminadas, pero de momento las vamos a dejar. Veamos que puntuación obtenemos en kaggel con un randomForest de 1000 arboles.

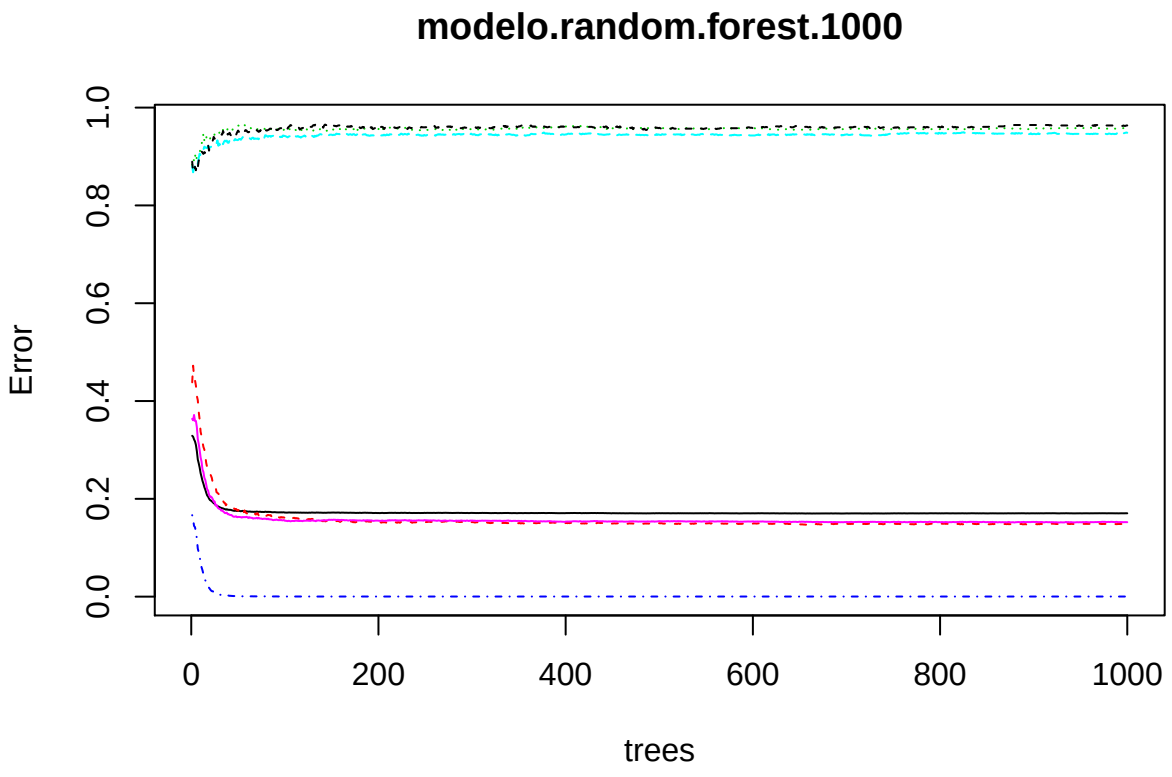
```
set.seed(1234)
modelo.random.forest.1000 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados2,ntree=10
print(modelo.random.forest.1000)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados2,      ntree = 1000)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 17.04%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3102                 4                134   15
## Colision_Obstaculo        309                41                366   26
## Colision_Vehiculos          1                 0              16516    0
## Otro                      620                 8                741   93
## Salida_Via                428                 1                461   12
## Vuelco                    344                 7                190   18
##                     Salida_Via Vuelco  class.error
## Atropello                  381      6 0.1482701812
## Colision_Obstaculo         207      3 0.9569327731
## Colision_Vehiculos           3      0 0.0002421308
## Otro                       329     16 0.9485334809
## Salida_Via                5098     13 0.1521702977
## Vuelco                     470     39 0.9634831461
```

```
randomForest::importance(modelo.random.forest.1000)
```

```
##                             MeanDecreaseGini
## ANIO                                589.6587
## MES                                1087.2367
## HORA                               1310.8334
## DIASEMANA                           774.7058
## PROVINCIA                           907.3905
## COMUNIDAD_AUTONOMA                  575.5232
## TOT_VICTIMAS                        186.6700
## TOT_HERIDOS_GRAVES                  137.6317
## TOT_HERIDOS_LEVES                   274.2688
## TOT_VEHICULOS_IMPLICADOS           7514.5480
## ZONA                                498.6843
## ZONA_AGRUPADA                       472.1825
## RED_CARRETERA                       309.1534
## TIPO_VIA                            448.5083
## TRAZADO_NO_INTERSEC                 501.7204
## TIPO_INTERSEC                       404.5535
## ACOND_CALZADA                       374.1605
## PRIORIDAD                           570.6430
## SUPERFICIE_CALZADA                  279.7407
## LUMINOSIDAD                         313.9580
## FACTORES_ATMOSFERICOS               197.2152
## VISIBILIDAD_RESTRINGIDA             177.0487
## OTRA_CIRCUNSTANCIA                  154.9273
## ACERAS                              149.1053
## DENSIDAD_CIRCULACION                162.9500
```

```
plot(modelo.random.forest.1000)
```



```
predicciones.rf.1000 <- predict(modelo.random.forest.1000,newdata=test.imputados2)
```

Veamos la puntuación en kaggel

```
salida.modelo.28 <- as.matrix(predicciones.rf.1000)
salida.modelo.28 <- cbind(c(1:(dim(salida.modelo.28)[1])), salida.modelo.28)
colnames(salida.modelo.28) <- c("Id","Prediction")
write.table(salida.modelo.28,file="predicciones/Prediccion28.txt",sep=",",quote = F,row.names = F)
```

28 El resultado de este modelo para la competición de Kaggel, subido el 06/03/2017 a las 15:04, con un total de 32 personas entregadas, se ha quedado en la posición 9 con una puntuación del 0.82859. Mi mejor puntuación obtenida hasta el momento.

16.2 Prueba del segundo modelo random forest 1000

```
rm(list=ls())
train.imputados <- read.csv("datasetmodificados/train-imputados.csv")
train.imputados$X = NULL
test.imputados <- read.csv("datasetmodificados/test-imputados.csv")
test.imputados$X = NULL
```

Vamos a probar a eliminar las variables: ISLA, TOT_MUERTOS, TOT_HERIDOS_GRAVES, ACERAS y MEDIDAS_ESPECIALES, es decir, dos más que anteriormente.

```
train.imputados3 <- train.imputados
test.imputados3 <- test.imputados
train.imputados3$ISLA = NULL
train.imputados3$TOT_MUERTOS=NULL
train.imputados3$TOT_HERIDOS_GRAVES=NULL
train.imputados3$ACERAS=NULL
train.imputados3$MEDIDAS_ESPECIALES=NULL
test.imputados3$ISLA = NULL
test.imputados3$TOT_MUERTOS=NULL
test.imputados3$TOT_HERIDOS_GRAVES=NULL
test.imputados3$ACERAS=NULL
test.imputados3$MEDIDAS_ESPECIALES=NULL
```

```
set.seed(1234)
modelo.random.forest.1000.3 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados3,ntree=
print(modelo.random.forest.1000.3)
```

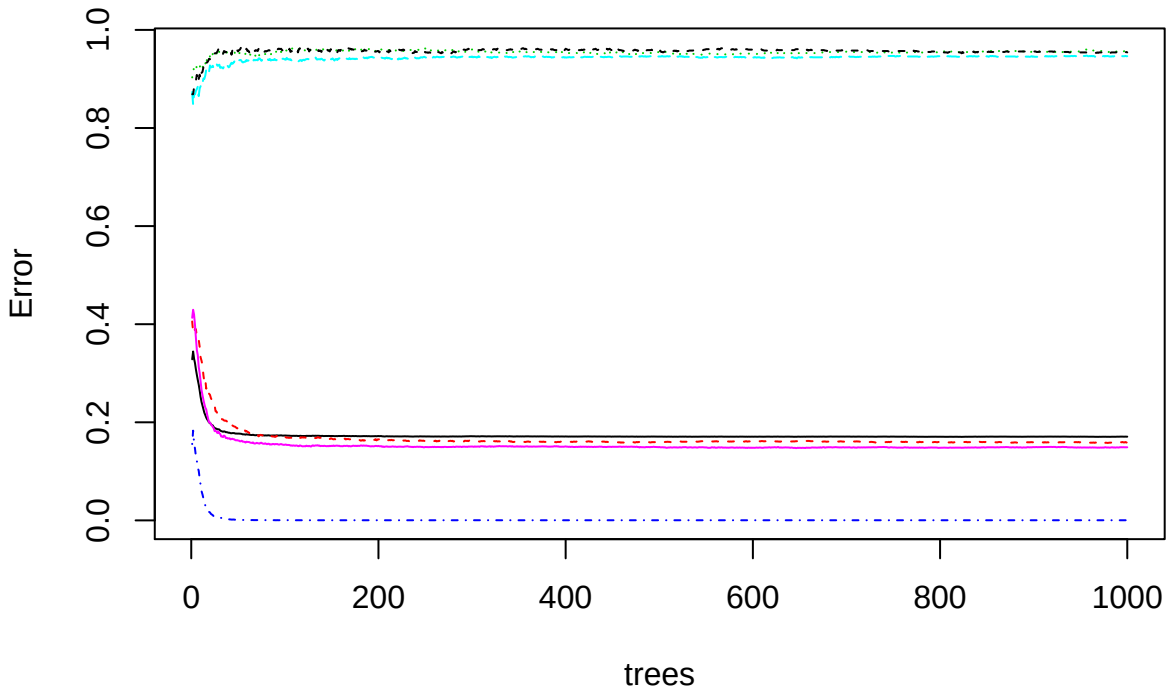
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados3,      ntree = 1000)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.07%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3063                 7                134   18
## Colision_Obstaculo        304                41                366   22
## Colision_Vehiculos          1                 0              16515    1
```

```
## Otro                     605                12               741   97
## Salida_Via               395                 2               461   20
## Vuelco                   325                 8               190   13
##                    Salida_Via Vuelco  class.error
## Atropello                 412      8 0.1589785832
## Colision_Obstaculo        218      1 0.9569327731
## Colision_Vehiculos          3      0 0.0003026634
## Otro                      339     13 0.9463198672
## Salida_Via               5117     18 0.1490104773
## Vuelco                    483     49 0.9541198502
```

```
randomForest::importance(modelo.random.forest.1000.3)
```

```
##                           MeanDecreaseGini
## ANIO                              580.3505
## MES                              1069.0716
## HORA                              923.6370
## DIASEMANA                         760.8058
## PROVINCIA                         908.2482
## COMUNIDAD_AUTONOMA                598.3356
## TOT_VICTIMAS                      202.8713
## TOT_HERIDOS_LEVES                 321.1894
## TOT_VEHICULOS_IMPLICADOS         7410.3409
## ZONA                              499.1266
## ZONA_AGRUPADA                     450.1925
## RED_CARRETERA                     339.0270
## TIPO_VIA                          463.6989
## TRAZADO_NO_INTERSEC               530.9900
## TIPO_INTERSEC                     396.0430
## ACOND_CALZADA                     383.3045
## PRIORIDAD                         588.7957
## SUPERFICIE_CALZADA                286.3667
## LUMINOSIDAD                       320.4969
## FACTORES_ATMOSFERICOS             205.8885
## VISIBILIDAD_RESTRINGIDA           189.3002
## OTRA_CIRCUNSTANCIA                164.6751
## DENSIDAD_CIRCULACION              176.9352
```

```
plot(modelo.random.forest.1000.3)
```




```
predicciones.rf.1000.3 <- predict(modelo.random.forest.1000.3,newdata=test.imputados3)
```

Veamos la puntuación en kaggel

```
salida.modelo.29 <- as.matrix(predicciones.rf.1000.3)
salida.modelo.29 <- cbind(c(1:(dim(salida.modelo.29)[1])), salida.modelo.29)
colnames(salida.modelo.29) <- c("Id","Prediction")
write.table(salida.modelo.29,file="predicciones/Prediccion29.txt",sep=",",quote = F,row.names = F)
```

29 El resultado de este modelo para la competición de Kaggel, subido el 07/03/2017 a las 13:17, con un total de 34 personas entregadas, se ha quedado en la posición 7 con una puntuación del 0.82909. Mi mejor puntuación hasta el momento.

16.3 Prueba del tercer modelo random forest 1000

```
train.imputados4 <- train.imputados3
test.imputados4 <- test.imputados3
train.imputados4$VISIBILIDAD_RESTRINGIDA=NULL
train.imputados4$OTRA_CIRCUNSTANCIA=NULL
train.imputados4$DENSIDAD_CIRCULACION=NULL
test.imputados4$VISIBILIDAD_RESTRINGIDA=NULL
test.imputados4$OTRA_CIRCUNSTANCIA=NULL
test.imputados4$DENSIDAD_CIRCULACION=NULL
```

```
write.csv(train.imputados4,"datasetmodificados/train-imputados4.csv",row.names = FALSE)
write.csv(test.imputados4,"datasetmodificados/test-imputados4.csv",row.names = FALSE)
```

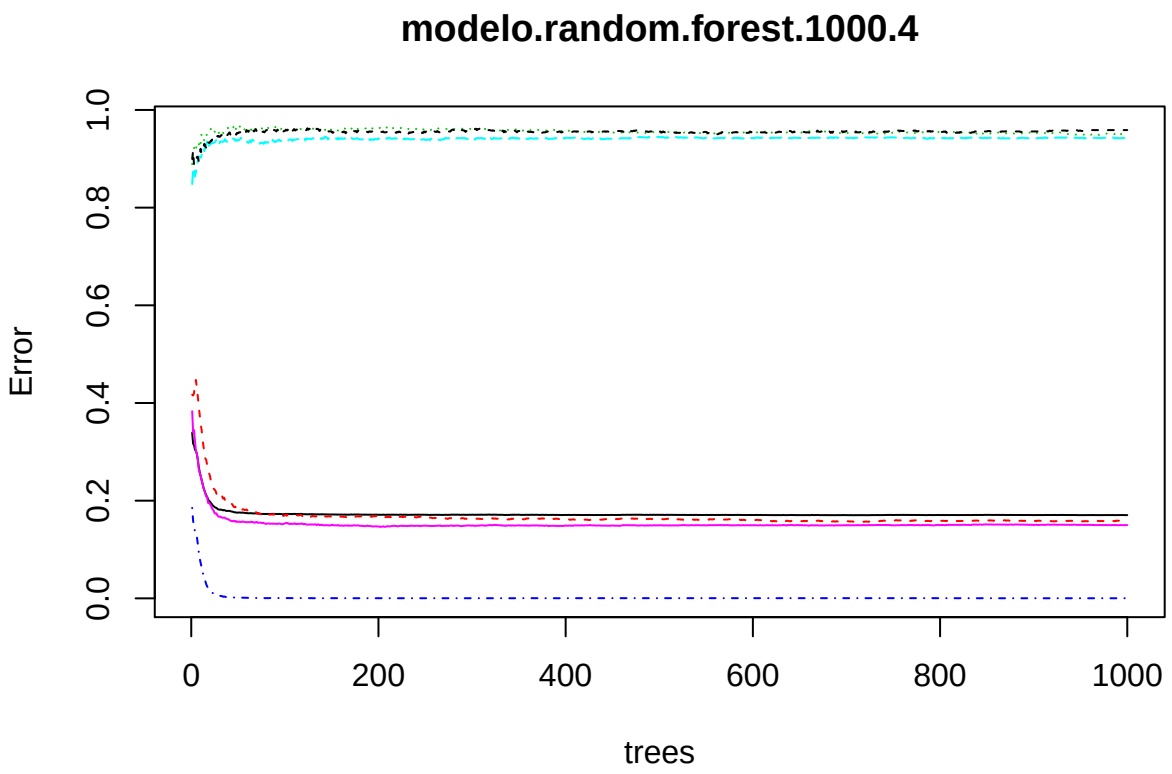
```
set.seed(1234)
modelo.random.forest.1000.4 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados4,ntree=
print(modelo.random.forest.1000.4)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados4,      ntree = 1000)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.05%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3065                  6                134   15
## Colision_Obstaculo        299                 46                366   21
## Colision_Vehiculos          1                  0              16516    0
## Otro                      594                 14                741  105
## Salida_Via                401                  2                461   19
## Vuelco                    332                  9                190   14
##                     Salida_Via Vuelco  class.error
## Atropello                  414      8 0.1584294344
## Colision_Obstaculo         219      1 0.9516806723
## Colision_Vehiculos           3      0 0.0002421308
## Otro                       344      9 0.9418926397
## Salida_Via                5111     19 0.1500083153
## Vuelco                     479     44 0.9588014981
```

```
randomForest::importance(modelo.random.forest.1000.4)
```

```
##                           MeanDecreaseGini
## ANIO                              637.0646
## MES                              1146.7712
## HORA                             1014.0819
## DIASEMANA                         818.1059
## PROVINCIA                         952.5064
## COMUNIDAD_AUTONOMA                613.9671
## TOT_VICTIMAS                      204.7555
## TOT_HERIDOS_LEVES                 333.1728
## TOT_VEHICULOS_IMPLICADOS         7600.2504
## ZONA                              511.2554
## ZONA_AGRUPADA                     465.0194
## RED_CARRETERA                     354.1712
## TIPO_VIA                          467.5787
## TRAZADO_NO_INTERSEC               532.0823
## TIPO_INTERSEC                     401.1101
## ACOND_CALZADA                     409.7885
## PRIORIDAD                         605.8878
## SUPERFICIE_CALZADA                307.1805
## LUMINOSIDAD                       332.7771
## FACTORES_ATMOSFERICOS             218.1122
```

```
plot(modelo.random.forest.1000.4)
```




```
predicciones.rf.1000.4 <- predict(modelo.random.forest.1000.4,newdata=test.imputados4)
```

Veamos la puntuación en kaggel

```
salida.modelo.30 <- as.matrix(predicciones.rf.1000.4)
salida.modelo.30 <- cbind(c(1:(dim(salida.modelo.30)[1])), salida.modelo.30)
colnames(salida.modelo.30) <- c("Id","Prediction")
write.table(salida.modelo.30,file="predicciones/Prediccion30.txt",sep=",",quote = F,row.names = F)
```

30 El resultado de este modelo para la competición de Kaggel, subido el 07/03/2017 a las 16:17, con un total de 34 personas entregadas, se ha quedado en la posición 5 con una puntuación del 0.82958. La mejor obtenida hasta el momento.

16.4 Prueba del cuarto modelo random forest 1000

```
train.imputados5 <- train.imputados4
test.imputados5 <- test.imputados4
train.imputados5$FACTORES_ATMOSFERICOS   =NULL
train.imputados5$TOT_VICTIMAS   =NULL
test.imputados5$FACTORES_ATMOSFERICOS  =NULL
test.imputados5$TOT_VICTIMAS  =NULL
```

```
set.seed(1234)
modelo.random.forest.1000.5 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados5,ntree=
print(modelo.random.forest.1000.5)
```

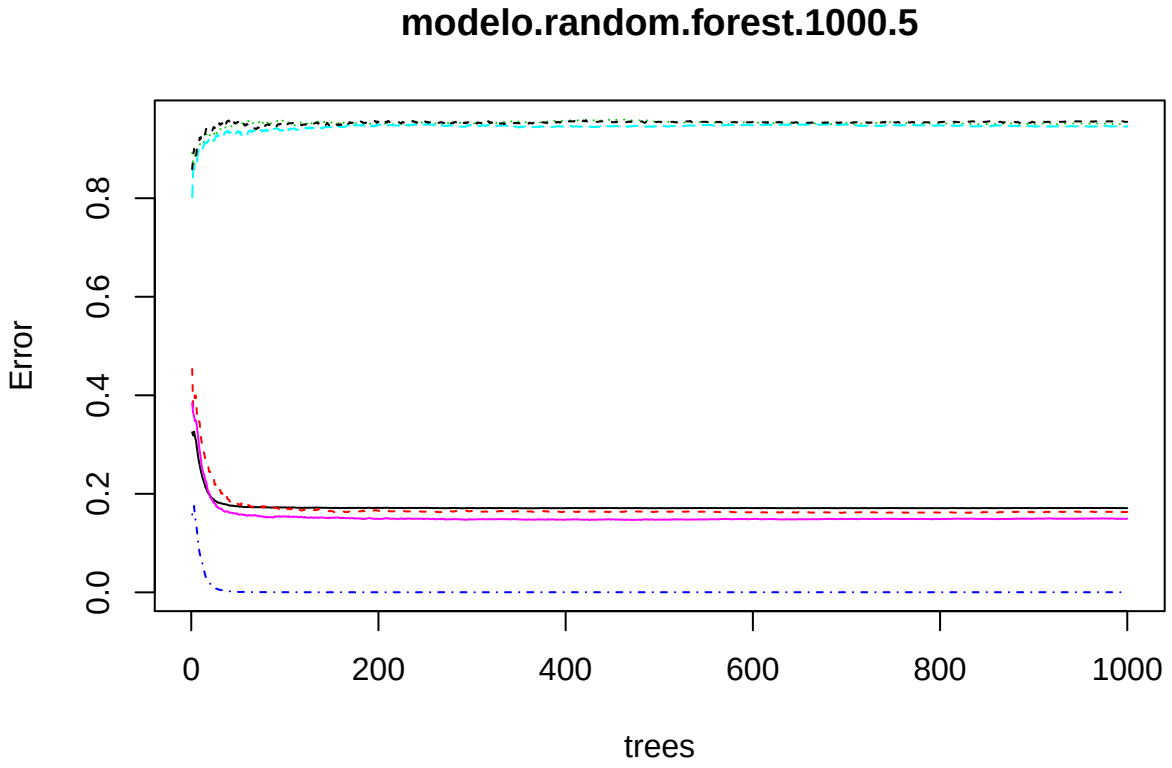
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados5,      ntree = 1000)
##                Type of random forest: classification
```

```
##                       Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.11%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3047                  6                134   23
## Colision_Obstaculo        279                 47                366   26
## Colision_Vehiculos          1                  0              16515    1
## Otro                      587                 14                744   98
## Salida_Via                393                  1                461   22
## Vuelco                    323                 12                190   12
##                     Salida_Via Vuelco  class.error
## Atropello                  423      9 0.1633717738
## Colision_Obstaculo         232      2 0.9506302521
## Colision_Vehiculos           3      0 0.0003026634
## Otro                       352     12 0.9457664638
## Salida_Via                5115     21 0.1493430900
## Vuelco                     483     48 0.9550561798
```

```
randomForest::importance(modelo.random.forest.1000.5)
```

```
##                           MeanDecreaseGini
## ANIO                              688.6595
## MES                              1203.4052
## HORA                             1096.9123
## DIASEMANA                         858.1233
## PROVINCIA                         977.0290
## COMUNIDAD_AUTONOMA                616.5996
## TOT_HERIDOS_LEVES                 423.4795
## TOT_VEHICULOS_IMPLICADOS         7687.2367
## ZONA                              492.3793
## ZONA_AGRUPADA                     501.1468
## RED_CARRETERA                     344.6490
## TIPO_VIA                          476.7495
## TRAZADO_NO_INTERSEC               555.8933
## TIPO_INTERSEC                     411.5885
## ACOND_CALZADA                     423.0923
## PRIORIDAD                         620.2188
## SUPERFICIE_CALZADA                350.1412
## LUMINOSIDAD                       341.1103
```

```
plot(modelo.random.forest.1000.5)
```




```
predicciones.rf.1000.5 <- predict(modelo.random.forest.1000.5,newdata=test.imputados5)
```

Veamos la puntuación en kaggel

```
salida.modelo.31 <- as.matrix(predicciones.rf.1000.5)
salida.modelo.31 <- cbind(c(1:(dim(salida.modelo.31)[1])), salida.modelo.31)
colnames(salida.modelo.31) <- c("Id","Prediction")
write.table(salida.modelo.31,file="predicciones/Prediccion31.txt",sep=",",quote = F,row.names = F)
```

31 El resultado de este modelo para la competición de Kaggel, subido el 07/03/2017 a las 16:46, con un total de 34 personas entregadas, se ha quedado en la posición 5 con una puntuación del 0.82928.

16.5 Prueba del quinto modelo random forest 1000

```
train.imputados6 <- train.imputados5
test.imputados6 <- test.imputados5
train.imputados6$RED_CARRETERA=NULL
train.imputados6$SUPERFICIE_CALZADA=NULL
train.imputados6$LUMINOSIDAD=NULL
test.imputados6$RED_CARRETERA=NULL
test.imputados6$SUPERFICIE_CALZADA=NULL
test.imputados6$LUMINOSIDAD=NULL
```

```
set.seed(1234)
modelo.random.forest.1000.6 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados6,ntree=
print(modelo.random.forest.1000.6)
```

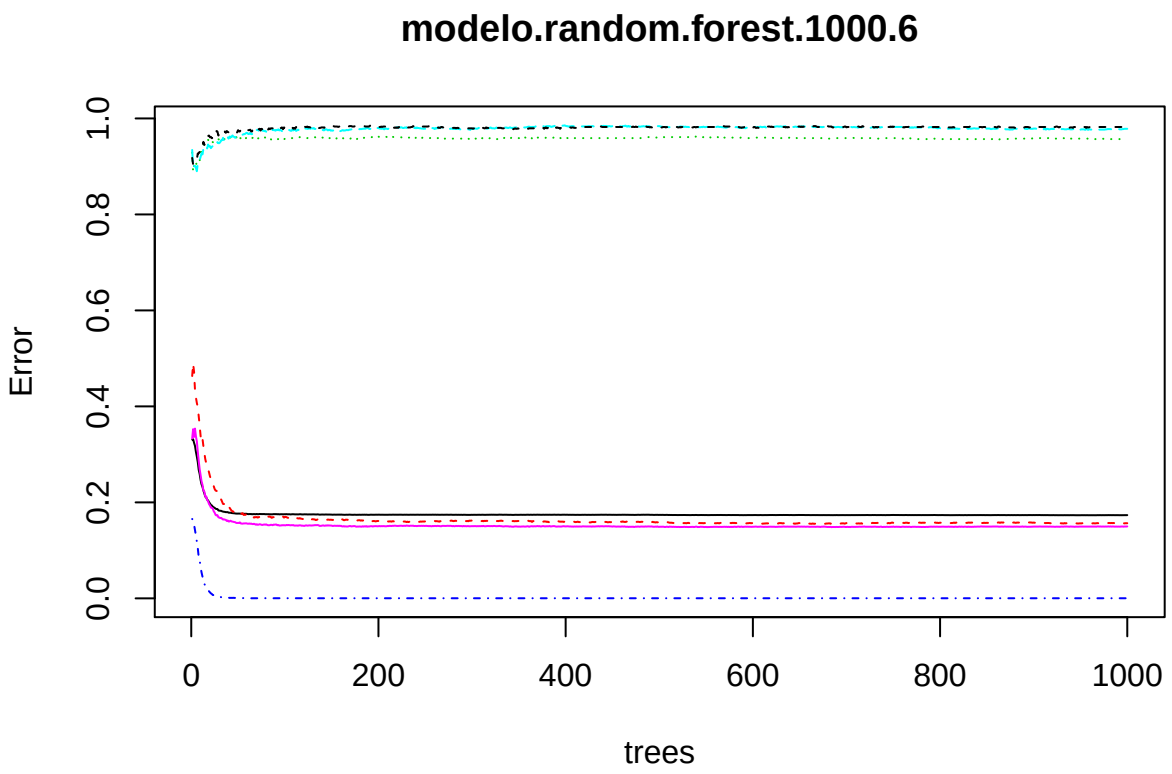
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados6,      ntree = 1000)
```

```
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.34%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3072                  2                134   16
## Colision_Obstaculo        302                 41                366   20
## Colision_Vehiculos          1                  0              16516    0
## Otro                      656                 15                752   39
## Salida_Via                423                  1                461   11
## Vuelco                    361                  9                190    4
##                     Salida_Via Vuelco  class.error
## Atropello                  411      7 0.1565074135
## Colision_Obstaculo         221      2 0.9569327731
## Colision_Vehiculos           3      0 0.0002421308
## Otro                       340      5 0.9784172662
## Salida_Via                5112      5 0.1498420090
## Vuelco                     485     19 0.9822097378
```

```
randomForest::importance(modelo.random.forest.1000.6)
```

```
##                           MeanDecreaseGini
## ANIO                              647.8791
## MES                              1147.0123
## HORA                             1097.9864
## DIASEMANA                         823.1808
## PROVINCIA                         971.8658
## COMUNIDAD_AUTONOMA                638.6504
## TOT_HERIDOS_LEVES                 428.5468
## TOT_VEHICULOS_IMPLICADOS         7585.4064
## ZONA                              571.1115
## ZONA_AGRUPADA                     546.0210
## TIPO_VIA                          559.9627
## TRAZADO_NO_INTERSEC               565.2896
## TIPO_INTERSEC                     433.2686
## ACOND_CALZADA                     423.3745
## PRIORIDAD                         638.3905
```

```
plot(modelo.random.forest.1000.6)
```




```
predicciones.rf.1000.6 <- predict(modelo.random.forest.1000.6,newdata=test.imputados6)
```

Veamos la puntuación en kaggel

```
salida.modelo.32 <- as.matrix(predicciones.rf.1000.6)
salida.modelo.32 <- cbind(c(1:(dim(salida.modelo.32)[1])), salida.modelo.32)
colnames(salida.modelo.32) <- c("Id","Prediction")
write.table(salida.modelo.32,file="predicciones/Prediccion32.txt",sep=",",quote = F,row.names = F)
```

32 El resultado de este modelo para la competición de Kaggel, subido el 07/03/2017 a las 17:08, con un total de 34 personas entregadas, se ha quedado en la posición 5 con una puntuación del 0.82523.

16.6 Prueba del sexto modelo random forest 1000

```
train.imputados7 <- train.imputados6
test.imputados7 <- test.imputados6
train.imputados7$TOT_HERIDOS_LEVES=NULL
train.imputados7$TIPO_INTERSEC=NULL
train.imputados7$ACOND_CALZADA=NULL
test.imputados7$TOT_HERIDOS_LEVES=NULL
test.imputados7$TIPO_INTERSEC=NULL
test.imputados7$ACOND_CALZADA=NULL
```

```
set.seed(1234)
modelo.random.forest.1000.7 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados7,ntree=
print(modelo.random.forest.1000.7)
```

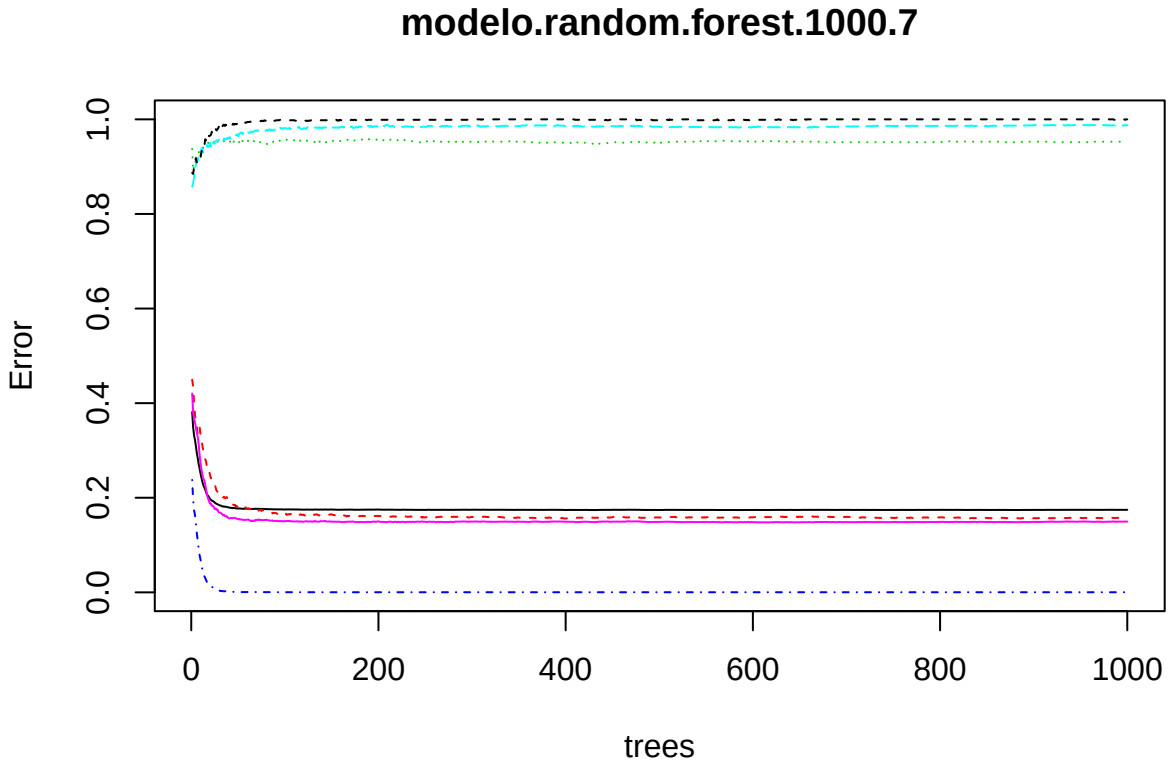
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados7,      ntree = 1000)
```

```
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 17.46%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3069                  4                134    9
## Colision_Obstaculo        304                 44                366   13
## Colision_Vehiculos          1                  0              16516    0
## Otro                      677                 15                752   22
## Salida_Via                433                  1                461    4
## Vuelco                    387                  8                190    2
##                     Salida_Via Vuelco  class.error
## Atropello                  426      0 0.1573311367
## Colision_Obstaculo         225      0 0.9537815126
## Colision_Vehiculos           3      0 0.0002421308
## Otro                       341      0 0.9878251245
## Salida_Via                5114      0 0.1495093963
## Vuelco                     481      0 1.0000000000
```

```
randomForest::importance(modelo.random.forest.1000.7)
```

```
##                           MeanDecreaseGini
## ANIO                             761.7012
## MES                             1263.0336
## HORA                            1304.9438
## DIASEMANA                        909.6144
## PROVINCIA                       1000.9767
## COMUNIDAD_AUTONOMA               625.2349
## TOT_VEHICULOS_IMPLICADOS        7900.4500
## ZONA                             585.8857
## ZONA_AGRUPADA                    583.7367
## TIPO_VIA                         568.2353
## TRAZADO_NO_INTERSEC              719.2759
## PRIORIDAD                        723.9653
```

```
plot(modelo.random.forest.1000.7)
```




```
predicciones.rf.1000.7 <- predict(modelo.random.forest.1000.7,newdata=test.imputados7)
```

Veamos la puntuación en kaggel

```
salida.modelo.33 <- as.matrix(predicciones.rf.1000.7)
salida.modelo.33 <- cbind(c(1:(dim(salida.modelo.33)[1])), salida.modelo.33)
colnames(salida.modelo.33) <- c("Id","Prediction")
write.table(salida.modelo.33,file="predicciones/Prediccion33.txt",sep=",",quote = F,row.names = F)
```

33 El resultado de este modelo para la competición de Kaggel, subido el 07/03/2017 a las 17:31, con un total de 34 personas entregadas, se ha quedado en la posición 5 con una puntuación del 0.82444.

16.7 Prueba del primer modelo random forest multihebra

Vamos a probar un random forest multihebra para ver si mejoramos los resultados:

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))

rf <- foreach(ntree = rep(250, 8), .combine = combine, .packages = "randomForest") %dopar% randomForest
rf
```

```
##
## Call:
##  randomForest(x = train.imputados4[, -21], y = train.imputados4[,      21], ntree = ntree)
##                Type of random forest: classification
##                      Number of trees: 2000
## No. of variables tried at each split: 4
```

```
randomForest::importance(rf)
```

```
##                            MeanDecreaseGini
## ANIO                                636.7936
## MES                                1147.3344
## HORA                               1017.6961
## DIASEMANA                           816.2991
## PROVINCIA                           955.8797
## COMUNIDAD_AUTONOMA                  616.4888
## TOT_VICTIMAS                        208.7926
## TOT_HERIDOS_LEVES                   335.4808
## TOT_VEHICULOS_IMPLICADOS           7596.4153
## ZONA                                532.8808
## ZONA_AGRUPADA                       454.7372
## RED_CARRETERA                       342.6276
## TIPO_VIA                            467.8246
## TRAZADO_NO_INTERSEC                 527.4450
## TIPO_INTERSEC                       407.5344
## ACOND_CALZADA                       409.0404
## PRIORIDAD                           604.7804
## SUPERFICIE_CALZADA                  306.4540
## LUMINOSIDAD                         333.1854
## FACTORES_ATMOSFERICOS               216.9858
```

```
predicciones.rf <- predict(rf,newdata=test.imputados4)
```

Veamos la puntuación en kaggel

```
salida.modelo.34 <- as.matrix(predicciones.rf)
salida.modelo.34 <- cbind(c(1:(dim(salida.modelo.34)[1])), salida.modelo.34)
colnames(salida.modelo.34) <- c("Id","Prediction")
write.table(salida.modelo.34,file="predicciones/Prediccion34.txt",sep=",",quote = F,row.names = F)
```

34 El resultado de este modelo para la competición de Kaggel, subido el 08/03/2017 a las 08:55, con un total de 34 personas entregadas, se ha quedado en la posición 4 con una puntuación del 0.82978. La mejor obtenida hasta el momento.

16.8 Prueba del primer modelo random forest 5000

Probemos el mejor modelo obtenido pero con 5000 árboles en lugar de 1000.

```
set.seed(1234)
modelo.random.forest.5000.4 <- randomForest::randomForest(TIPO_ACCIDENTE~.,data=train.imputados4,ntree=5
print(modelo.random.forest.5000.4)
```

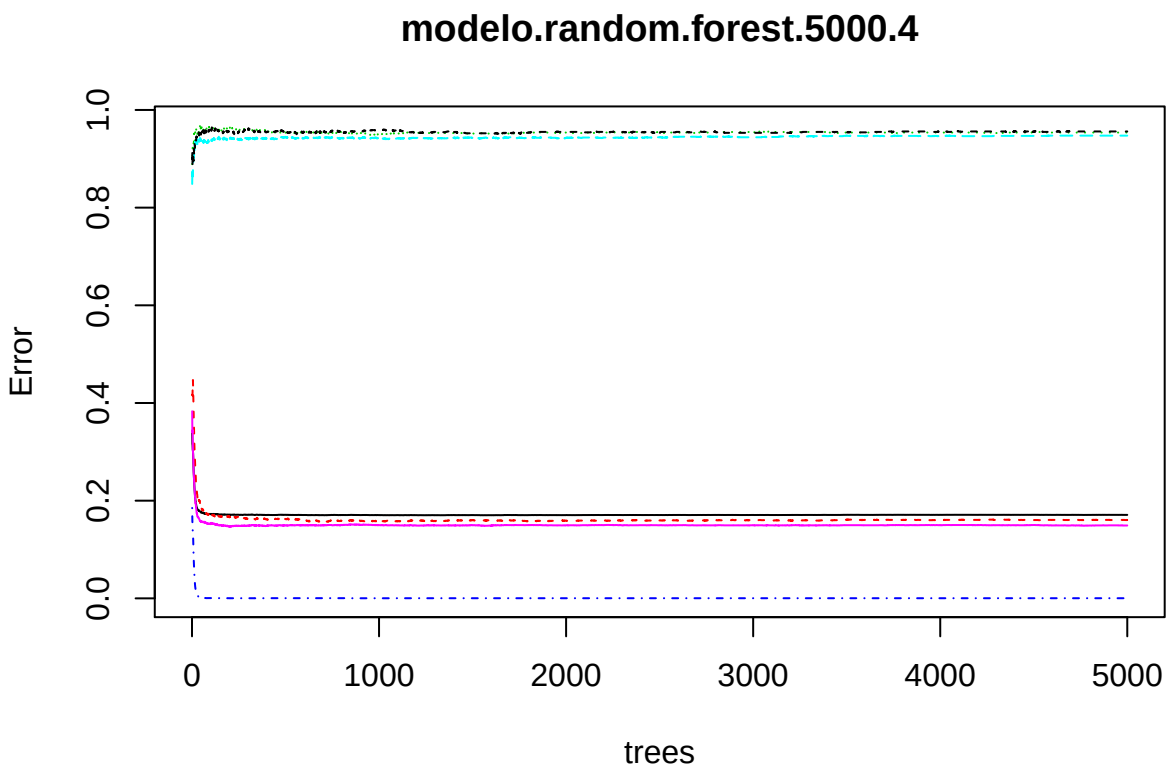
```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = train.imputados4,      ntree = 5000)
##                Type of random forest: classification
##                      Number of trees: 5000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.09%
## Confusion matrix:
##                    Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello               3058                  5                134   18
## Colision_Obstaculo       301                 44                366   24
```

```
## Colision_Vehiculos         1                 0              16515    1
## Otro                     603                14                742   95
## Salida_Via               394                 2                461   20
## Vuelco                   331                 9                190   12
##                    Salida_Via Vuelco  class.error
## Atropello                 418      9 0.1603514552
## Colision_Obstaculo        216      1 0.9537815126
## Colision_Vehiculos          3      0 0.0003026634
## Otro                      343     10 0.9474266740
## Salida_Via               5116     20 0.1491767836
## Vuelco                    479     47 0.9559925094
```

```
randomForest::importance(modelo.random.forest.5000.4)
```

```
##                           MeanDecreaseGini
## ANIO                              638.1873
## MES                              1147.2891
## HORA                             1018.7670
## DIASEMANA                         818.2054
## PROVINCIA                         955.8504
## COMUNIDAD_AUTONOMA                616.2137
## TOT_VICTIMAS                      208.1459
## TOT_HERIDOS_LEVES                 334.7055
## TOT_VEHICULOS_IMPLICADOS         7574.9246
## ZONA                              506.8121
## ZONA_AGRUPADA                     475.0043
## RED_CARRETERA                     342.4732
## TIPO_VIA                          469.0437
## TRAZADO_NO_INTERSEC               537.2965
## TIPO_INTERSEC                     407.2884
## ACOND_CALZADA                     408.9214
## PRIORIDAD                         608.9561
## SUPERFICIE_CALZADA                306.9504
## LUMINOSIDAD                       334.7172
## FACTORES_ATMOSFERICOS             217.7855
```

```
plot(modelo.random.forest.5000.4)
```




```
predicciones.rf.5000.4 <- predict(modelo.random.forest.5000.4,newdata=test.imputados4)
```

Veamos la puntuación en kaggel

```
salida.modelo.35 <- as.matrix(predicciones.rf.5000.4)
salida.modelo.35 <- cbind(c(1:(dim(salida.modelo.35)[1])), salida.modelo.35)
colnames(salida.modelo.35) <- c("Id","Prediction")
write.table(salida.modelo.35,file="predicciones/Prediccion35.txt",sep=",",quote = F,row.names = F)
```

35 El resultado de este modelo para la competición de Kaggel, subido el 08/03/2017 a las 08:55, con un total de 34 personas entregadas, se ha quedado en la posición 4 con una puntuación del 0.82918.

16.9 Prueba del segundo modelo random forest multihebra

Ya que tenemos un muy buen modelo, vamos a probar a eliminar algo de ruido para ver el comportamiento de nuestro modelo.

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Eliminamos ruido:

```
set.seed(1234)
out <- IPF(TIPO_ACCIDENTE~., data=train.imputados4,s=2)
```

```
## Iteration 1: 5150 noisy instances removed
## Iteration 2: 144 noisy instances removed
## Iteration 3: 35 noisy instances removed
```

```
identical(out$cleanData, train.imputados4[setdiff(1:nrow(train.imputados4),out$remIdx),])
```

```
## [1] TRUE
```

```
train.sin.ruido1 <- train.imputados4[setdiff(1:nrow(train.imputados4),out$remIdx),]
```

Veamos como funciona nuestro nuevo dataset sin ruido con el modelo.

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))

rf.sin.ruido <- foreach(ntree = rep(250, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf.sin.ruido
```

```
##
## Call:
##  randomForest(x = train.sin.ruido1[, -21], y = train.sin.ruido1[,      21], ntree = ntree)
##                Type of random forest: classification
##                      Number of trees: 2000
## No. of variables tried at each split: 4
```

```
randomForest::importance(rf.sin.ruido)
```

```
##                           MeanDecreaseGini
## ANIO                             32.68369
## MES                              78.57305
## HORA                             88.78645
## DIASEMANA                        66.35231
## PROVINCIA                       118.35612
## COMUNIDAD_AUTONOMA              100.81670
## TOT_VICTIMAS                     75.67428
## TOT_HERIDOS_LEVES               104.13490
## TOT_VEHICULOS_IMPLICADOS       8416.25119
## ZONA                            697.72363
## ZONA_AGRUPADA                   758.44991
## RED_CARRETERA                   313.57985
## TIPO_VIA                        410.13946
## TRAZADO_NO_INTERSEC             286.30057
## TIPO_INTERSEC                   195.63998
## ACOND_CALZADA                    27.38692
## PRIORIDAD                       233.95018
## SUPERFICIE_CALZADA              116.05747
## LUMINOSIDAD                      61.08356
## FACTORES_ATMOSFERICOS            24.65885
```

```
predicciones.rf.sin.ruido <- predict(rf.sin.ruido,newdata=test.imputados4)
```

Veamos la puntuación en kaggel

```
salida.modelo.36 <- as.matrix(predicciones.rf.sin.ruido)
salida.modelo.36 <- cbind(c(1:(dim(salida.modelo.36)[1])), salida.modelo.36)
colnames(salida.modelo.36) <- c("Id","Prediction")
write.table(salida.modelo.36,file="predicciones/Prediccion36.txt",sep=",",quote = F,row.names = F)
```

36 El resultado de este modelo para la competición de Kaggel, subido el 08/03/2017 a las 11:52, con un total de 36 personas entregadas, se ha quedado en la posición 4 con una puntuación del 0.82177.

16.10 Prueba del tercer modelo random forest multihebra

Como la detección de ruido no ha funcionado correctamente, vamos a probar a eliminar una nueva variable del mejor dataset que tenemos hasta ahora. Esta variable será TOT_VICTIMAS al ser la siguiente variable con menos importancia según el randomforest obtenido con 5000 árboles.

```
rm(list=ls())
train.multihebra <- read.csv("datasetmodificados/train-imputados4.csv")
test.multihebra <- read.csv("datasetmodificados/test-imputados4.csv")
train.multihebra$TOT_VICTIMAS=NULL
test.multihebra$TOT_VICTIMAS=NULL
```

Veamos como funciona en este modelo:

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf.1 <- foreach(ntree = rep(250, 8), .combine = combine, .packages = "randomForest") %dopar% randomFores
rf.1
```

```
##
## Call:
##  randomForest(x = train.multihebra[, -20], y = train.multihebra[,      20], ntree = ntree)
##                Type of random forest: classification
##                      Number of trees: 2000
## No. of variables tried at each split: 4
```

```
randomForest::importance(rf.1)
```

```
##                           MeanDecreaseGini
## ANIO                             662.1596
## MES                             1175.2766
## HORA                            1057.7225
## DIASEMANA                        836.9117
## PROVINCIA                        961.8621
## COMUNIDAD_AUTONOMA               613.2936
## TOT_HERIDOS_LEVES                420.3577
## TOT_VEHICULOS_IMPLICADOS        7656.2220
## ZONA                             508.8067
## ZONA_AGRUPADA                    464.6439
## RED_CARRETERA                    338.2711
## TIPO_VIA                         492.9251
## TRAZADO_NO_INTERSEC              534.4878
## TIPO_INTERSEC                    410.1515
## ACOND_CALZADA                    418.0177
## PRIORIDAD                        610.4477
## SUPERFICIE_CALZADA               308.6987
## LUMINOSIDAD                      336.2143
## FACTORES_ATMOSFERICOS            219.6317
```

```
predicciones.rf.1 <- predict(rf.1,newdata=test.multihebra)
```

Veamos la puntuación en kaggel

```
salida.modelo.37 <- as.matrix(predicciones.rf.1)
salida.modelo.37 <- cbind(c(1:(dim(salida.modelo.37)[1])), salida.modelo.37)
colnames(salida.modelo.37) <- c("Id","Prediction")
write.table(salida.modelo.37,file="predicciones/Prediccion37.txt",sep=",",quote = F,row.names = F)
```

37 El resultado de este modelo para la competición de Kaggel, subido el 08/03/2017 a las 12:47, con un total de 36 personas entregadas, se ha quedado en la posición 4 con una puntuación del 0.82928.

16.11 Prueba del cuarto modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados:

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))

rf2 <- foreach(ntree = rep(500, 8), .combine = combine, .packages = "randomForest") %dopar% randomFores
```

```
## Warning: cerrando la conenexion 28 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 27 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 26 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 25 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 24 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 23 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 22 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 21 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 20 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 19 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 18 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 17 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 16 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 15 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 14 (<-localhost:11539) que no esta siendo
## utilizada
```

```
## Warning: cerrando la conenexion 13 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 12 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 11 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 10 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 9 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 8 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 7 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 6 (<-localhost:11539) que no esta siendo
## utilizada

## Warning: cerrando la conenexion 5 (<-localhost:11539) que no esta siendo
## utilizada
```

```
rf2
```

```
##
## Call:
##  randomForest(x = train.imputados4[, -21], y = train.imputados4[,      21], ntree = ntree)
##                Type of random forest: classification
##                      Number of trees: 4000
## No. of variables tried at each split: 4
```

```
randomForest::importance(rf2)
```

```
##                           MeanDecreaseGini
## ANIO                              636.2332
## MES                              1146.5342
## HORA                             1020.6475
## DIASEMANA                         817.7046
## PROVINCIA                         954.8841
## COMUNIDAD_AUTONOMA                616.6516
## TOT_VICTIMAS                      207.9788
## TOT_HERIDOS_LEVES                 332.9851
## TOT_VEHICULOS_IMPLICADOS         7588.8921
## ZONA                              509.9174
## ZONA_AGRUPADA                     481.0992
## RED_CARRETERA                     334.7371
## TIPO_VIA                          474.2286
## TRAZADO_NO_INTERSEC               530.7893
## TIPO_INTERSEC                     403.8982
## ACOND_CALZADA                     409.0746
## PRIORIDAD                         608.5688
## SUPERFICIE_CALZADA                306.0741
## LUMINOSIDAD                       332.0145
## FACTORES_ATMOSFERICOS             217.9605
```

```
predicciones.rf2 <- predict(rf2,newdata=test.imputados4)
```

Veamos la puntuación en kaggel

```
salida.modelo.38 <- as.matrix(predicciones.rf2)
salida.modelo.38 <- cbind(c(1:(dim(salida.modelo.38)[1])), salida.modelo.38)
colnames(salida.modelo.38) <- c("Id","Prediction")
write.table(salida.modelo.38,file="predicciones/Prediccion38.txt",sep=",",quote = F,row.names = F)
```

38 El resultado de este modelo para la competición de Kaggel, subido el 08/03/2017 a las 12:53, con un total de 36 personas entregadas, se ha quedado en la posición 4 con una puntuación del 0.82968. La mejor obtenida hasta el momento.

16.12 Prueba del quinto modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf3 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf3
randomForest::importance(rf3)
predicciones.rf3 <- predict(rf3,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

Veamos la puntuación en kaggel

```
salida.modelo.39 <- as.matrix(predicciones.rf3)
salida.modelo.39 <- cbind(c(1:(dim(salida.modelo.39)[1])), salida.modelo.39)
colnames(salida.modelo.39) <- c("Id","Prediction")
write.table(salida.modelo.39,file="predicciones/Prediccion39.txt",sep=",",quote = F,row.names = F)
```

39 El resultado de este modelo para la competición de Kaggel, subido el 08/03/2017 a las 08:55, con un total de 34 personas entregadas, se ha quedado en la posición 4 con una puntuación del 0.82978. La mejor obtenida hasta el momento.

16.13 Prueba del sexto modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Quito la variable TOT_VICTIMAS ya que en la última ejecución era la que peores resultados obtenia en la importancia.

```
train.imputados4$TOT_VICTIMAS=NULL
test.imputados4$TOT_VICTIMAS=NULL
```

```
Call:
 randomForest(x = train.imputados4[, -21], y = train.imputados4[,      21], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 4

                         MeanDecreaseGini
ANIO                              637.4089
MES                              1148.4941
HORA                             1019.1715
DIASEMANA                         816.8215
PROVINCIA                         959.6640
COMUNIDAD_AUTONOMA                616.5881
TOT_VICTIMAS                      208.7811
TOT_HERIDOS_LEVES                 333.5539
TOT_VEHICULOS_IMPLICADOS         7571.3513
ZONA                              501.9540
ZONA_AGRUPADA                     471.2785
RED_CARRETERA                     337.2357
TIPO_VIA                          492.2486
TRAZADO_NO_INTERSEC               538.6846
TIPO_INTERSEC                     406.8601
ACOND_CALZADA                     409.4426
PRIORIDAD                         604.6783
SUPERFICIE_CALZADA                306.7240
LUMINOSIDAD                       333.2735
FACTORES_ATMOSFERICOS             217.9749
```

Figure 1: 39 modelo

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf4 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf4
randomForest::importance(rf4)
predicciones.rf4 <- predict(rf4,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 4

                          MeanDecreaseGini
ANIO                              661.7107
MES                              1174.2634
HORA                             1056.6390
DIASEMANA                         835.8159
PROVINCIA                         961.4743
COMUNIDAD_AUTONOMA                610.5590
TOT_HERIDOS_LEVES                 420.5197
TOT_VEHICULOS_IMPLICADOS         7668.1868
ZONA                              508.4001
ZONA_AGRUPADA                     480.5978
RED_CARRETERA                     333.9434
TIPO_VIA                          481.8083
TRAZADO_NO_INTERSEC               535.7853
TIPO_INTERSEC                     402.3122
ACOND_CALZADA                     418.7694
PRIORIDAD                         612.9580
SUPERFICIE_CALZADA                308.5999
LUMINOSIDAD                       337.4307
FACTORES_ATMOSFERICOS             218.2433
```

Figure 2: 40 modelo

Veamos la puntuación en kaggel

```
salida.modelo.40 <- as.matrix(predicciones.rf4)
salida.modelo.40 <- cbind(c(1:(dim(salida.modelo.40)[1])), salida.modelo.40)
colnames(salida.modelo.40) <- c("Id","Prediction")
write.table(salida.modelo.40,file="predicciones/Prediccion40.txt",sep=",",quote = F,row.names = F)
```

40 El resultado de este modelo para la competición de Kaggle, subido el 17/03/2017 a las 12:17, con un total de 43 personas entregadas, se ha quedado en la posición 8 con una puntuación del 0.82948.

16.14 Prueba del setimo modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Quito la variable FACTORES_ATMOSFERICOS ya que en la última ejecución era la que peores resultados

obtenia en la importancia.

```
train.imputados4$TOT_VICTIMAS=NULL
test.imputados4$TOT_VICTIMAS=NULL
train.imputados4$FACTORES_ATMOSFERICOS=NULL
test.imputados4$FACTORES_ATMOSFERICOS=NULL
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf5 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf5
randomForest::importance(rf5)
predicciones.rf5 <- predict(rf5,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 4

                         MeanDecreaseGini
ANIO                             687.6513
MES                             1202.9372
HORA                            1095.3743
DIASEMANA                        858.0655
PROVINCIA                        971.6621
COMUNIDAD_AUTONOMA               614.1638
TOT_HERIDOS_LEVES                423.2340
TOT_VEHICULOS_IMPLICADOS        7719.2923
ZONA                             523.4227
ZONA_AGRUPADA                    488.7407
RED_CARRETERA                    334.4452
TIPO_VIA                         480.0854
TRAZADO_NO_INTERSEC              541.2594
TIPO_INTERSEC                    403.6292
ACOND_CALZADA                    426.8352
PRIORIDAD                        617.3724
SUPERFICIE_CALZADA               350.3040
LUMINOSIDAD                      339.2668
```

Figure 3: 41 modelo

Veamos la puntuación en kaggel

```
salida.modelo.41 <- as.matrix(predicciones.rf5)
salida.modelo.41 <- cbind(c(1:(dim(salida.modelo.41)[1])), salida.modelo.41)
colnames(salida.modelo.41) <- c("Id","Prediction")
write.table(salida.modelo.41,file="predicciones/Prediccion41.txt",sep=",",quote = F,row.names = F)
```

41 El resultado de este modelo para la competición de Kaggle, subido el 17/03/2017 a las 12:45, con un total de 43 personas entregadas, se ha quedado en la posición 8 con una puntuación del 0.82869.

16.15 Prueba del octavo modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Quito la variable RED_CARRETERA ya que en la última ejecución era la que peores resultados obtenia en la importancia.

```
train.imputados4$TOT_VICTIMAS=NULL
test.imputados4$TOT_VICTIMAS=NULL
train.imputados4$FACTORES_ATMOSFERICOS=NULL
test.imputados4$FACTORES_ATMOSFERICOS=NULL
train.imputados4$RED_CARRETERA=NULL
test.imputados4$RED_CARRETERA=NULL
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf6 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf6
randomForest::importance(rf6)
predicciones.rf6 <- predict(rf6,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 4

                          MeanDecreaseGini
ANIO                              718.2862
MES                              1240.8111
HORA                             1143.5789
DIASEMANA                         883.3247
PROVINCIA                        1005.7515
COMUNIDAD_AUTONOMA                636.1076
TOT_HERIDOS_LEVES                 431.4646
TOT_VEHICULOS_IMPLICADOS         7755.8626
ZONA                              591.7884
ZONA_AGRUPADA                     543.4461
TIPO_VIA                          529.2569
TRAZADO_NO_INTERSEC               539.8228
TIPO_INTERSEC                     414.2548
ACOND_CALZADA                     440.9419
PRIORIDAD                         625.5630
SUPERFICIE_CALZADA                355.5542
LUMINOSIDAD                       347.1181
```

Figure 4: 42 modelo

Veamos la puntuación en kaggel

```
salida.modelo.42 <- as.matrix(predicciones.rf6)
salida.modelo.42 <- cbind(c(1:(dim(salida.modelo.42)[1])), salida.modelo.42)
```

```
colnames(salida.modelo.42) <- c("Id","Prediction")
write.table(salida.modelo.42,file="predicciones/Prediccion42.txt",sep=",",quote = F,row.names = F)
```

42 El resultado de este modelo para la competición de Kaggle, subido el 17/03/2017 a las 13:22, con un total de 43 personas entregadas, se ha quedado en la posición 8 con una puntuación del 0.82869.

16.16 Prueba del noveno modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Quito la variable LUMINOSIDAD ya que en la última ejecución era la que peores resultados obtenia en la importancia.

```
train.imputados4$TOT_VICTIMAS=NULL
test.imputados4$TOT_VICTIMAS=NULL
train.imputados4$FACTORES_ATMOSFERICOS=NULL
test.imputados4$FACTORES_ATMOSFERICOS=NULL
train.imputados4$RED_CARRETERA=NULL
test.imputados4$RED_CARRETERA=NULL
train.imputados4$LUMINOSIDAD=NULL
test.imputados4$LUMINOSIDAD=NULL
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf7 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf7
randomForest::importance(rf7)
predicciones.rf7 <- predict(rf7,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

Veamos la puntuación en kaggel

```
salida.modelo.43 <- as.matrix(predicciones.rf7)
salida.modelo.43 <- cbind(c(1:(dim(salida.modelo.43)[1])), salida.modelo.43)
colnames(salida.modelo.43) <- c("Id","Prediction")
write.table(salida.modelo.43,file="predicciones/Prediccion43.txt",sep=",",quote = F,row.names = F)
```

43 El resultado de este modelo para la competición de Kaggle, subido el 17/03/2017 a las 14:09, con un total de 43 personas entregadas, se ha quedado en la posición 8 con una puntuación del 0.82869.

16.17 Prueba del decimo modelo random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 4

                         MeanDecreaseGini
ANIO                             743.2508
MES                             1273.1541
HORA                            1237.4155
DIASEMANA                        905.5510
PROVINCIA                       1008.5695
COMUNIDAD_AUTONOMA               630.5009
TOT_HERIDOS_LEVES                435.8874
TOT_VEHICULOS_IMPLICADOS        7809.0897
ZONA                             603.7767
ZONA_AGRUPADA                    570.5565
TIPO_VIA                         536.8755
TRAZADO_NO_INTERSEC              535.5324
TIPO_INTERSEC                    414.4780
ACOND_CALZADA                    452.2411
PRIORIDAD                        629.9022
SUPERFICIE_CALZADA               360.3176
```

Figure 5: 43 modelo

Quito la variable SUPERFICIE_CALZADA ya que en la última ejecución era la que peores resultados obtenia en la importancia.

```
train.imputados4$TOT_VICTIMAS=NULL
test.imputados4$TOT_VICTIMAS=NULL
train.imputados4$FACTORES_ATMOSFERICOS=NULL
test.imputados4$FACTORES_ATMOSFERICOS=NULL
train.imputados4$RED_CARRETERA=NULL
test.imputados4$RED_CARRETERA=NULL
train.imputados4$LUMINOSIDAD=NULL
test.imputados4$LUMINOSIDAD=NULL
train.imputados4$SUPERFICIE_CALZADA=NULL
test.imputados4$SUPERFICIE_CALZADA=NULL
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf8 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf8
randomForest::importance(rf8)
predicciones.rf8 <- predict(rf8,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

Veamos la puntuación en kaggel

```
salida.modelo.44 <- as.matrix(predicciones.rf8)
salida.modelo.44 <- cbind(c(1:(dim(salida.modelo.44)[1])), salida.modelo.44)
colnames(salida.modelo.44) <- c("Id","Prediction")
write.table(salida.modelo.44,file="predicciones/Prediccion44.txt",sep=",",quote = F,row.names = F)
```

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 3

                         MeanDecreaseGini
ANIO                             651.6581
MES                             1157.2335
HORA                            1103.9065
DIASEMANA                        824.4057
PROVINCIA                        980.8982
COMUNIDAD_AUTONOMA               645.7240
TOT_HERIDOS_LEVES                436.3358
TOT_VEHICULOS_IMPLICADOS        7572.9743
ZONA                             569.8215
ZONA_AGRUPADA                    518.3150
TIPO_VIA                         552.2393
TRAZADO_NO_INTERSEC              565.3872
TIPO_INTERSEC                    434.8094
ACOND_CALZADA                    422.9605
PRIORIDAD                        641.1648
```

Figure 6: 44 modelo

44 El resultado de este modelo para la competición de Kaggle, subido el 18/03/2017 a las 10:59, con un total de 44 personas entregadas, se ha quedado en la posición 9 con una puntuación del 0.82513.

16.18 Prueba del modelo 11 de random forest multihebra

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
```

Quito la variable ACOND_CALZADA ya que en la última ejecución era la que peores resultados obtenia en la importancia.

```
train.imputados4$TOT_VICTIMAS=NULL
test.imputados4$TOT_VICTIMAS=NULL
train.imputados4$FACTORES_ATMOSFERICOS=NULL
test.imputados4$FACTORES_ATMOSFERICOS=NULL
train.imputados4$RED_CARRETERA=NULL
test.imputados4$RED_CARRETERA=NULL
train.imputados4$LUMINOSIDAD=NULL
test.imputados4$LUMINOSIDAD=NULL
train.imputados4$SUPERFICIE_CALZADA=NULL
test.imputados4$SUPERFICIE_CALZADA=NULL
train.imputados4$ACOND_CALZADA=NULL
test.imputados4$ACOND_CALZADA=NULL
```

Vamos a probar un random forest multihebra con más árboles para ver si mejoramos los resultados: (No lo he ejecutado para la creación del pdf ya que me da fallo de memoria, pero si lo ejecuto como un script normal de R no tengo ningún problema)

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf9 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
```

```
rf9
randomForest::importance(rf9)
predicciones.rf9 <- predict(rf9,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 3

                         MeanDecreaseGini
ANIO                              665.6072
MES                              1166.2957
HORA                             1131.7282
DIASEMANA                         831.4029
PROVINCIA                         977.1230
COMUNIDAD_AUTONOMA                636.4936
TOT_HERIDOS_LEVES                 441.2919
TOT_VEHICULOS_IMPLICADOS         7646.2754
ZONA                              573.5357
ZONA_AGRUPADA                     531.3980
TIPO_VIA                          551.0469
TRAZADO_NO_INTERSEC               570.0591
TIPO_INTERSEC                     436.8980
PRIORIDAD                         649.7545
```

Figure 7: 45 modelo

Veamos la puntuación en kaggel

```
salida.modelo.45 <- as.matrix(predicciones.rf9)
salida.modelo.45 <- cbind(c(1:(dim(salida.modelo.45)[1])), salida.modelo.45)
colnames(salida.modelo.45) <- c("Id","Prediction")
write.table(salida.modelo.45,file="predicciones/Prediccion45.txt",sep=",",quote = F,row.names = F)
```

45 El resultado de este modelo para la competición de Kaggle, subido el 18/03/2017 a las 11:00, con un total de 44 personas entregadas, se ha quedado en la posición 9 con una puntuación del 0.82543.

17 Balanceo de los datos

Veamos cuantos elementos hay de cada tipo de accidente:

```
table(train.imputados4$TIPO_ACCIDENTE)/dim(train.imputados4)[1]
```

```
##
##          Atropello Colision_Obstaculo Colision_Vehiculos
##         0.12139191         0.03173122         0.55062996
##               Otro          Salida_Via             Vuelco
##         0.06022932         0.20041997         0.03559763
```

Por lo tanto, vamos a probar a balancear un poco más el número de los datos que tenemos, ya que por ejemplo Vuelco, Colision_obstauclo y Otro, tienen menos de un 1% de los datos. El balanceo que propongo será duplicar las instancias de cada conjunto de estos tipos.

```
rm(list=ls())
train.imputados4 <- read.csv("datasetmodificados/train-imputados4.csv")
```

```
test.imputados4 <- read.csv("datasetmodificados/test-imputados4.csv")
valores.vuelco <- train.imputados4[train.imputados4$TIPO_ACCIDENTE=="Vuelco",]
valores.colision.obstaculo <- train.imputados4[train.imputados4$TIPO_ACCIDENTE=="Colision_Obstaculo",]
valores.otro <- train.imputados4[train.imputados4$TIPO_ACCIDENTE=="Otro",]
train.imputados4 <- rbind(train.imputados4, valores.vuelco, valores.colision.obstaculo, valores.otro)
```

Quedando:

```
table(train.imputados4$TIPO_ACCIDENTE)/dim(train.imputados4)[1]
```

```
##
##          Atropello Colision_Obstaculo Colision_Vehiculos
##         0.10765911         0.05628307         0.48833841
##               Otro         Salida_Via             Vuelco
##         0.10683142         0.17774690         0.06314109
```

Veamos como funciona este modelo

```
set.seed(1234)
registerDoSNOW(makeCluster(8, type="SOCK"))
rf9 <- foreach::foreach(ntree = rep(750, 8), .combine = combine, .packages = "randomForest") %dopar% ra
rf9
randomForest::importance(rf9)
predicciones.rf9 <- predict(rf9,newdata=test.imputados4)
```

Muestro la imagen que me ha proporcionado el script de R:

```
Call:
 randomForest(x = train.imputados4[, -(dim(train.imputados4)[2])],      y = train.imputados4[, (dim(train.imputados4)
[2])], ntree = ntree)
               Type of random forest: classification
                     Number of trees: 6000
No. of variables tried at each split: 4

                         MeanDecreaseGini
ANIO                              985.9167
MES                              1807.5485
HORA                             1528.8145
DIASEMANA                        1282.0101
PROVINCIA                        1476.4366
COMUNIDAD_AUTONOMA                943.6295
TOT_VICTIMAS                      283.6946
TOT_HERIDOS_LEVES                 476.8119
TOT_VEHICULOS_IMPLICADOS         7336.4196
ZONA                              496.6514
ZONA_AGRUPADA                     432.7088
RED_CARRETERA                     410.4496
TIPO_VIA                          571.5148
TRAZADO_NO_INTERSEC               630.8712
TIPO_INTERSEC                     531.8381
ACOND_CALZADA                     635.4105
PRIORIDAD                         825.1161
SUPERFICIE_CALZADA                461.4751
LUMINOSIDAD                       487.1301
FACTORES_ATMOSFERICOS             359.1316
```

Figure 8: 46 modelo

Veamos la puntuación en kaggel

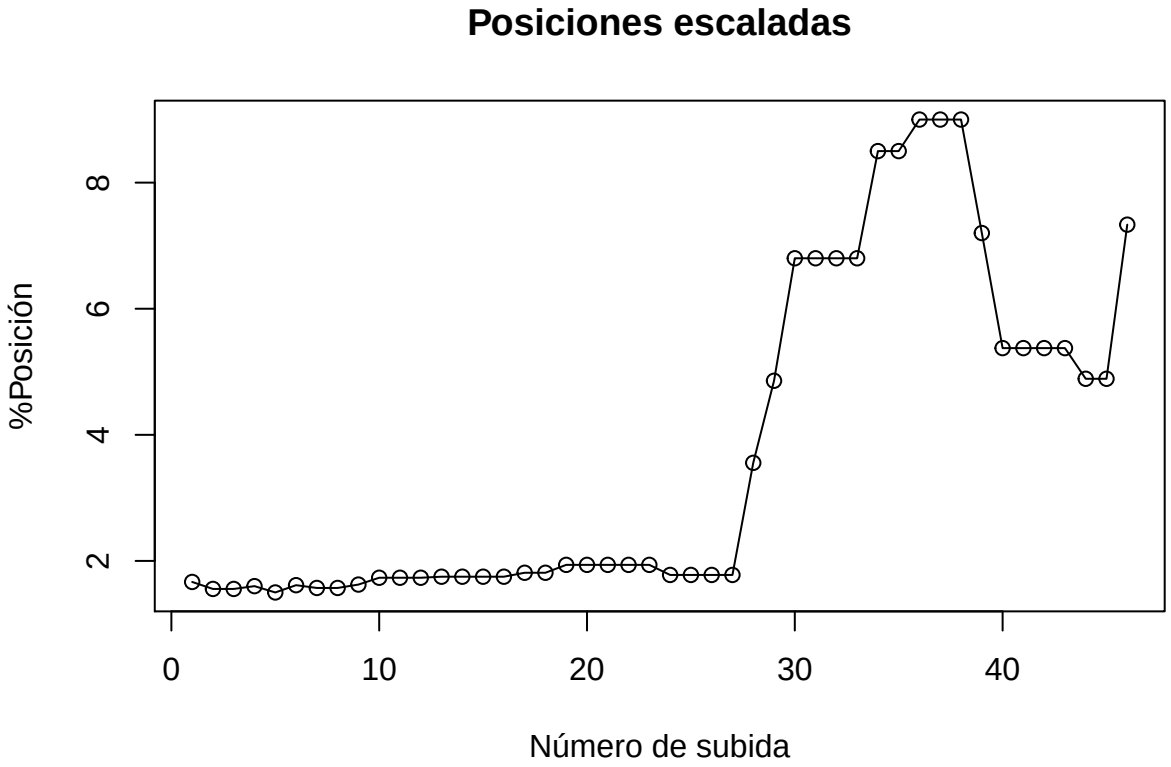
```
salida.modelo.46 <- as.matrix(predicciones.rf9)
salida.modelo.46 <- cbind(c(1:(dim(salida.modelo.46)[1])), salida.modelo.46)
```

```
colnames(salida.modelo.46) <- c("Id","Prediction")
write.table(salida.modelo.46,file="predicciones/Prediccion46.txt",sep=",",quote = F,row.names = F)
```

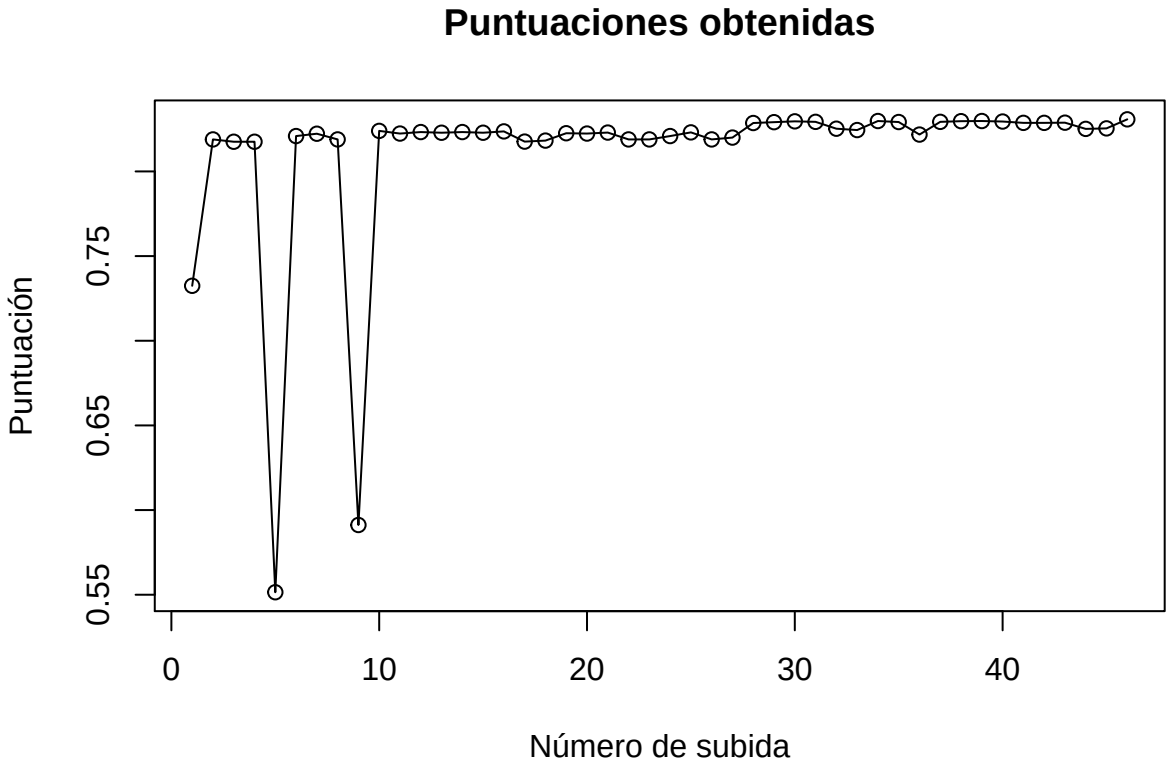
46 El resultado de este modelo para la competición de Kaggle, subido el 18/03/2017 a las 11:02, con un total de 44 personas entregadas, se ha quedado en la posición 6 con una puntuación del 0.83076. El mejor obtenido hasta el momento.

18 Gráfica de resultados

```
indices = c(1:46)
posicion = c(3/5, 9/14, 9/14, 10/16, 12/18, 13/21, 14/22, 14/22, 16/26, 15/26,
             15/26, 15/26, 16/28, 16/28, 16/28, 16/28, 16/29, 16/29, 16/31, 16/31,
             16/31, 16/31, 16/31, 18/32, 18/32, 18/32, 18/32, 9/32, 7/34, 5/34,
             5/34, 5/34, 5/34, 4/34, 4/34, 4/36, 4/36, 4/36, 5/36, 8/43,
             8/43, 8/43, 8/43, 9/44, 9/44, 6/44)
posicion2 = 1/posicion
puntuacion = c(0.73246, 0.81891, 0.81753, 0.81753, 0.55147, 0.82089, 0.82227, 0.81891, 0.59119, 0.82395
               0.82237, 0.82326, 0.82286, 0.82326, 0.82286, 0.82365, 0.81762, 0.81822, 0.82256, 0.82237,
               0.82296, 0.81891, 0.81891, 0.82089, 0.82306, 0.81891, 0.82000, 0.82859, 0.82909, 0.82958,
               0.82928, 0.82523, 0.82444, 0.82978, 0.82918, 0.82177, 0.82928, 0.82968, 0.82978, 0.82948,
               0.82869, 0.82869, 0.82879, 0.82513, 0.82543, 0.83076)
plot(indices,posicion2,main="Posiciones escaladas",xlab="Número de subida",ylab="%Posición",type="o")
```

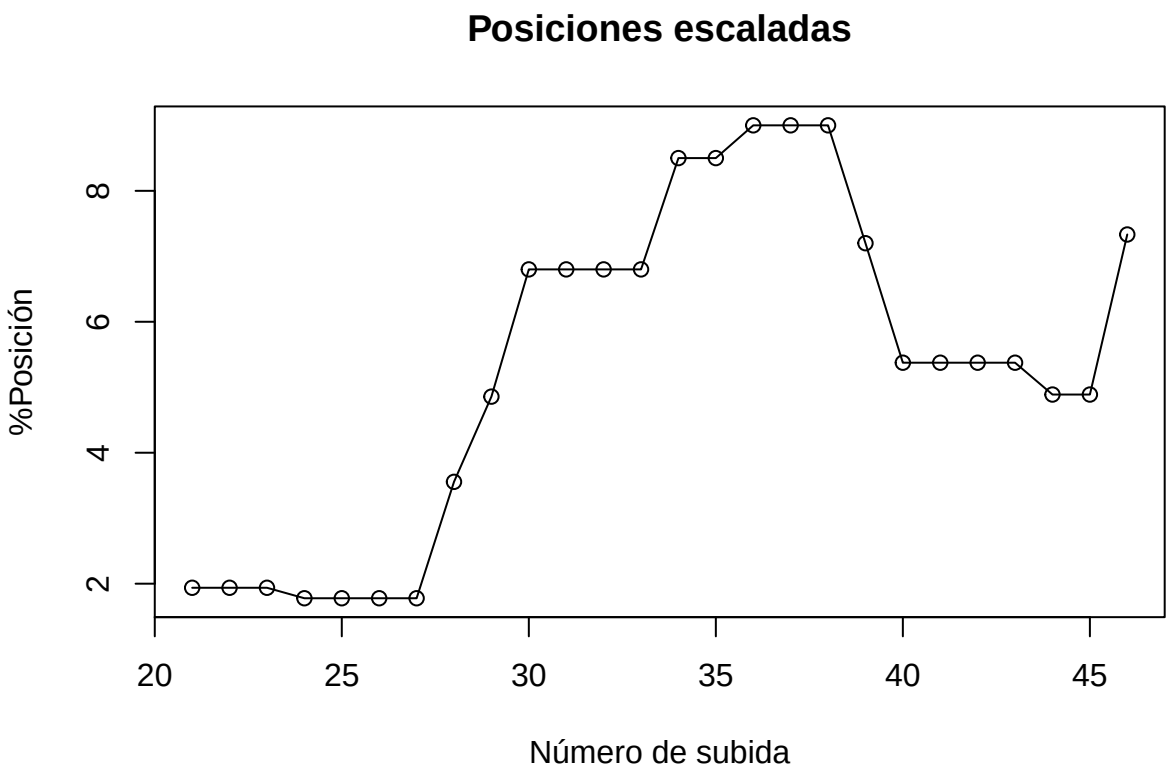



```
plot(indices,puntuacion,main="Puntuaciones obtenidas",xlab="Número de subida",ylab="Puntuación",type="o
```



Muestro la gráfica de la subida 21 en adelante para apreciar mejor la diferencia:

```
indices = c(21:46)
posicion = c(16/31, 16/31, 16/31, 18/32, 18/32, 18/32, 18/32, 9/32, 7/34, 5/34,
             5/34, 5/34, 5/34, 4/34, 4/34, 4/36, 4/36, 4/36, 5/36, 8/43,
             8/43, 8/43, 8/43, 9/44, 9/44, 6/44)
posicion2 = 1/posicion
puntuacion = c(0.82296, 0.81891, 0.81891, 0.82089, 0.82306, 0.81891, 0.82000, 0.82859, 0.82909, 0.82958
               0.82928, 0.82523, 0.82444, 0.82978, 0.82918, 0.82177, 0.82928, 0.82968, 0.82978, 0.82948,
               0.82869, 0.82869, 0.82879, 0.82513, 0.82543, 0.83076)
plot(indices,posicion2,main="Posiciones escaladas",xlab="Número de subida",ylab="%Posición",type="o")
```




```
plot(indices,puntuacion,main="Puntuaciones obtenidas",xlab="Número de subida",ylab="Puntuación",type="o
```