Informe Minería de datos: Preprocesamiento y clasificación. Máster en Ciencia de Datos e Ingeniería de Computadores, Universidad de Granada

Adrián Calzadilla González

11/4/2017

Contents

List of Figures

Introducción

Esta práctica corresponde a la asignatura *Minería de datos: preprocesamiento y clasificación* perteneciente al *Máster en Ciencia de Datos e Ingeniería de computadores* de la *Universidad de Granada*. Y servirá para la calificación de la asignatura.

En esta práctica se usarán los métodos de preprocesamiento y aprendizaje vistos en la asignatura, para ello se hará uso de la plataforma Kaggle, que permite establecer una competición entre todos los alumnos.

Para dicha competición se analizará un dataset con información sobre accidentes de tráfico, donde el objetivo es clasisificar a partir de las variables dadas de que tipo de accidente se trata.

El mi caso el usuario utilizado para la competición kaggle ha sido *Adrián*. Quedando finalmente en el puesto número 23 con una puntuación de 0.82665.

En este informe se detallará el código realizado para la subida a kaggle con más puntuación privada 0.82817, que me hubiera colocado en la octava posición y la entrada al kaggle con mayor puntuación obtenida de las 5 entradas que había que seleccionar.

El código completo se puede ver en la siguiente dirección: https://github.com/AdCalzadilla/preproceamientoclasificacion.

Estructura del dataset

El conjunto de datos a analizar corresponde a una muestra de 30.002 instancias de accidentes que se han producido en España durante los años del 2008 al 2013. Además se cuenta con un conjunto de test de 19998 instancias donde se realizarán las predicciones.

El dataset está formada por 31 variables, siendo la variable de salida *"TIPO_ACCIDENTE"*. Las otras variables, predictoras, que componen al dataset son las siguientes:

- ANIO
- MES
- HORA
- DIASEMANA
- PROVINCIA
- COMUNIDAD_AUTONOMA
- ISLA
- TOT_VICTIMAS
- TOT_MUERTOS
- TOT_HERIDOS
- TOT_VEHICULOSIMPLICADOS
- ZONA
- ZONA_AGRUPADA
- RED_CARRETERA
- CARRETERA
- TIPO_VIA
- TRAZADO_NO_INTERSEC
- TIPO_INTERSEC
- ACOND_CALZADA
- PRIORIDAD
- SUPERFICE_CALZADA
- LUMINOSIDAD
- FACTORES_ATMOSFERICOS
- VISIBILIDAD_RESTRINGIDA
- OTRA_CIRCUNSTANCIA

- ACERAS
- DENSIDAD_CIRCULACION
- MEDIDAS_ESPECIALES

Descripción de las variables

ANIO

Año en el que se produce el accidente.

```
describe(accident.train$ANIO)
```

```
## accident.train$ANIO
##        n  missing distinct     Info     Mean      Gmd
##    30002        0        6    0.972     2010    1.974
##
## Value       2008  2009  2010  2011  2012  2013
## Frequency   5438  5091  4871  4675  4891  5036
## Proportion 0.181 0.170 0.162 0.156 0.163 0.168
```

MES

Mes en el que se produce el accidente.

```
describe(accident.train$MES)
```

```
## accident.train$MES
##        n  missing distinct
##    30002        0       12
##
## Abril (2384, 0.079), Agosto (2418, 0.081), Diciembre (2448, 0.082), Enero
## (2395, 0.080), Febrero (2362, 0.079), Julio (2757, 0.092), Junio (2649,
## 0.088), Marzo (2446, 0.082), Mayo (2605, 0.087), Noviembre (2447, 0.082),
## Octubre (2600, 0.087), Septiembre (2491, 0.083)
```

HORA

Hora a la que se produce el accidente.

```
describe(accident.train$HORA)
```

```
## accident.train$HORA
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    30002        0      448    0.998    13.93    6.153        4        7
##      .25      .50      .75      .90      .95
##       10       14       18       21       22
##
## lowest :  0.00000000  0.01666667  0.08333333  0.10000000  0.11666667
## highest: 23.66666667 23.75000000 23.83333333 23.86666667 23.91666667
```

DIASEMANA

Día de la semana en el que se produce el accidente.

```
describe(accident.train$DIASEMANA)
```

```
## accident.train$DIASEMANA
##        n  missing distinct
##    30002        0        7
##
## Value        DOMINGO    JUEVES     LUNES    MARTES MIERCOLES    SABADO
## Frequency       3597      4351      4349      4343      4394      4000
## Proportion     0.120     0.145     0.145     0.145     0.146     0.133
##
## Value        VIERNES
## Frequency       4968
## Proportion     0.166
```

PROVINCIA

Provincia en la que se produce el accidente.

```
describe(accident.train$PROVINCIA)
```

```
## accident.train$PROVINCIA
##        n  missing distinct
##    30002        0       52
##
## lowest : Albacete         Alicante/Alacant Almeria          Araba/Alava      Asturias
## highest: Toledo           Valencia         Valladolid       Zamora           Zaragoza
```

COMUNIDAD_AUTONOMA

Comunidad Autónoma en la que se produce el accidente.

```
describe(accident.train$COMUNIDAD_AUTONOMA)
```

```
## accident.train$COMUNIDAD_AUTONOMA
##        n  missing distinct
##    30002        0       18
##
## lowest : Andalucia                  Aragon                      Asturias, Principado de     Balears
## highest: Madrid, Comunidad de       Murcia, Region de           Navarra, Comunidad Foral de Pais Va
```

ISLA

Indica si es una isla o no, y si es dice cual.

```
describe(accident.train$ISLA)
```

```
## accident.train$ISLA
##        n  missing distinct
##    30002        0       10
##
## FORMENTERA (8, 0.000), FUERTEVENTURA (35, 0.001), GRAN CANARIA (199,
## 0.007), IBIZA (117, 0.004), LA PALMA (37, 0.001), LANZAROTE (53, 0.002),
## MALLORCA (608, 0.020), MENORCA (33, 0.001), NO_ES_ISLA (28476, 0.949),
## TENERIFE (436, 0.015)
```

TOT_VICTIMAS

Número total de víctimas.

```
describe(accident.train$TOT_VICTIMAS)
```

```
## accident.train$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
```

TOT_MUERTOS

Número total de muertos que hubo en el accidente.

```
describe(accident.train$TOT_MUERTOS)
```

```
## accident.train$TOT_MUERTOS
##        n  missing distinct     Info     Mean      Gmd
##    30002        0        6    0.063  0.02447    0.048
##
## Value          0     1     2     3     4     7
## Frequency  29358   579    49    10     5     1
## Proportion 0.979 0.019 0.002 0.000 0.000 0.000
```

TOT_HERIDOS_GRAVES

Número de heridos catalogados como graves.

```
describe(accident.train$TOT_HERIDOS_GRAVES)
```

```
## accident.train$TOT_HERIDOS_GRAVES
##        n  missing distinct     Info     Mean      Gmd
##    30002        0        8    0.332   0.1453   0.2582
##
## Value          0     1     2     3     4     5     6     9
## Frequency  26212  3339   369    58    17     5     1     1
## Proportion 0.874 0.111 0.012 0.002 0.001 0.000 0.000 0.000
```

TOT_HERIDOS_LEVES

Número de heridos etiquetados con categoría leve.

```
describe(accident.train$TOT_HERIDOS_LEVES)
```

```
## accident.train$TOT_HERIDOS_LEVES
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    30002        0       16    0.712     1.26   0.7914        0        0
##      .25      .50      .75      .90      .95
##        1        1        1        2        3
##
```

```
## Value          0     1     2     3     4     5     6     7     8     9
## Frequency   3160 19682  4945  1316   557   191    71    40    18    10
## Proportion 0.105 0.656 0.165 0.044 0.019 0.006 0.002 0.001 0.001 0.000
##
## Value         10    11    12    13    15    18
## Frequency      5     2     1     1     1     2
## Proportion 0.000 0.000 0.000 0.000 0.000 0.000
```

TOT_VEHICULOS_IMPLICADOS

Número total de vehículos implicados en el accidente.

```
describe(accident.train$TOT_VEHICULOS_IMPLICADOS)
```

```
## accident.train$TOT_VEHICULOS_IMPLICADOS
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    30002        0       14    0.796    1.738   0.7055        1        1
##      .25      .50      .75      .90      .95
##        1        2        2        2        3
##
## Value          1     2     3     4     5     6     7     8     9    10
## Frequency  11583 15788  1956   455   130    43    27    10     4     2
## Proportion 0.386 0.526 0.065 0.015 0.004 0.001 0.001 0.000 0.000 0.000
##
## Value         11    12    17    21
## Frequency      1     1     1     1
## Proportion 0.000 0.000 0.000 0.000
```

ZONA_AGRUPADA

Agrupa la localización del accidente en Vías interurbanas o urbanas.

```
describe(accident.train$ZONA_AGRUPADA)
```

```
## accident.train$ZONA_AGRUPADA
##        n  missing distinct
##    30002        0        2
##
## Value      VIAS INTERURBANAS      VIAS URBANAS
## Frequency              13335             16667
## Proportion             0.444             0.556
```

RED_CARRETERA

Responsable de la carretera en la que se produzco el accidente.

```
describe(accident.train$RED_CARRETERA)
```

```
## accident.train$RED_CARRETERA
##        n  missing distinct
##    30002        0        5
##
## OTRAS TITULARIDADES (318, 0.011), TITULARIDAD AUTONOMICA (3890, 0.130),
## TITULARIDAD ESTATAL (4021, 0.134), TITULARIDAD MUNICIPAL (19077, 0.636),
## TITULARIDAD PROVINCIAL (DIPUTACION, CABILDO O CONSELL) (2696, 0.090)
```

TIPO_VIA

Tipo de vía en la que se produzco el accidente

```
describe(accident.train$TIPO_VIA)
```

```
## accident.train$TIPO_VIA
##        n  missing distinct
##    30002        0        9
##
## AUTOPISTA (723, 0.024), AUTOVIA (2941, 0.098), CAMINO VECINAL (519,
## 0.017), OTRO TIPO (15527, 0.518), RAMAL DE ENLACE (101, 0.003), VIA
## CONVENCIONAL (10044, 0.335), VIA CONVENCIONAL CON CARRIL LENTO (51,
## 0.002), VIA DE SERVICIO (52, 0.002), VIA PARA AUTOMOVILES (44, 0.001)
```

TIPO_INTERSEC

Describe si es una intersección el tipo de intersección donde se produjo el accidente.

```
describe(accident.train$TIPO_INTERSEC)
```

```
## accident.train$TIPO_INTERSEC
##        n  missing distinct
##    30002        0        7
##
## EN T O Y (3350, 0.112), EN X O + (4714, 0.157), ENLACE DE ENTRADA (421,
## 0.014), ENLACE DE SALIDA (223, 0.007), GIRATORIA (2006, 0.067),
## NO_ES_INTERSECCION (18983, 0.633), OTROS (305, 0.010)
```

ACOND_CALZADA

Refleja que tipo de calzada es.

```
describe(accident.train$ACOND_CALZADA)
```

```
## accident.train$ACOND_CALZADA
##        n  missing distinct
##     6303    23699        6
##
## CARRIL CENTRAL DE ESPERA (193, 0.031), NADA ESPECIAL (4645, 0.737), OTRO
## TIPO (791, 0.125), PASO PARA PEATONES O ISLETAS EN CENTRO DE VIA PRINCIPAL
## (397, 0.063), RAQUETA DE GIRO IZQUIERDA (109, 0.017), SOLO ISLETAS O PASO
## PARA PEATONES (168, 0.027)
```

PRIORIDAD

Normativa que se aplica al lugar donde se produjo el accidente.

```
describe(accident.train$PRIORIDAD)
```

```
## accident.train$PRIORIDAD
##        n  missing distinct
##    21880     8122        8
##
## AGENTE (26, 0.001), NINGUNA (SOLO NORMA) (13495, 0.617), OTRA SEÆAL (695,
## 0.032), PASO PARA PEATONES (848, 0.039), SEÆAL DE CEDA EL PASO (1629,
## 0.074), SEÆAL DE STOP (1750, 0.080), SEMAFORO (1778, 0.081), SOLO MARCAS
```

```
## VIALES (1659, 0.076)
```

SUPERFICIE_CALZADA

Estado de la calzada en el momento del accidente.

```
describe(accident.train$SUPERFICIE_CALZADA)
```

```
## accident.train$SUPERFICIE_CALZADA
##        n  missing distinct
##    30002        0        9
##
## ACEITE (83, 0.003), BARRILLO (27, 0.001), GRAVILLA SUELTA (150, 0.005),
## HELADA (72, 0.002), MOJADA (3895, 0.130), NEVADA (47, 0.002), OTRO TIPO
## (327, 0.011), SECA Y LIMPIA (25236, 0.841), UMBRIA (165, 0.005)
```

LUMINOSIDAD

Luminosidad que había en el momento del accidente.

```
describe(accident.train$LUMINOSIDAD)
```

```
## accident.train$LUMINOSIDAD
##        n  missing distinct
##    30002        0        5
##
## CREPUSCULO (1330, 0.044), NOCHE: ILUMINACION INSUFICIENTE (1067, 0.036),
## NOCHE: ILUMINACION SUFICIENTE (4793, 0.160), NOCHE: SIN ILUMINACION (1815,
## 0.060), PLENO DIA (20997, 0.700)
```

FACTORES_ATMOSFERICOS

Descripción meteorológica en el momento del accidente.

```
describe(accident.train$FACTORES_ATMOSFERICOS)
```

```
## accident.train$FACTORES_ATMOSFERICOS
##        n  missing distinct
##    30002        0        9
##
## BUEN TIEMPO (25852, 0.862), GRANIZANDO (50, 0.002), LLOVIZNANDO (2524,
## 0.084), LLUVIA FUERTE (499, 0.017), NEVANDO (66, 0.002), NIEBLA INTENSA
## (57, 0.002), NIEBLA LIGERA (83, 0.003), OTRO (715, 0.024), VIENTO FUERTE
## (156, 0.005)
```

VISIBILIDAD_RESTRINGIDA

Describe si hubo algún condicionante que redujo la visibilidad en el momento del accidente.

```
describe(accident.train$VISIBILIDAD_RESTRINGIDA)
```

```
## accident.train$VISIBILIDAD_RESTRINGIDA
##        n  missing distinct
##    19317    10685        8
##
## CONFIGURACION DEL TERRENO (989, 0.051), DESLUMBRAMIENTO (123, 0.006),
```

```
## EDIFICIOS (229, 0.012), FACTORES ATMOSFERICOS (374, 0.019), OTRA_CAUSA
## (491, 0.025), POLVO O HUMO (13, 0.001), SIN RESTRICCION (16982, 0.879),
## VEGETACION (116, 0.006)
```

OTRA_CIRCUNSTANCIA

```
describe(accident.train$OTRA_CIRCUNSTANCIA)
```

```
## accident.train$OTRA_CIRCUNSTANCIA
##        n  missing distinct
##    26763     3239       14
##
## BACHES (88, 0.003), BADEN (41, 0.002), CAMBIO DE RASANTE (100, 0.004),
## ESCALON (8, 0.000), ESTRECHAMIENTO (59, 0.002), FIN CARRIL LENTO (11,
## 0.000), FIRME DESLIZANTE SEÆALIZADO (18, 0.001), FUERTE DESCENSO (227,
## 0.008), INUNDACION (19, 0.001), NINGUNA (24967, 0.933), OBRAS (263,
## 0.010), OTRA (942, 0.035), PASO A NIVEL (13, 0.000), PERALTE INVERTIDO (7,
## 0.000)
```

ACERAS

Refleja si la vía en la que se produjo el accidente tiene acera o no.

```
describe(accident.train$ACERAS)
```

```
## accident.train$ACERAS
##        n  missing distinct
##    26853     3149        2
##
## Value      NO HAY ACERA SI HAY ACERA
## Frequency         21416         5437
## Proportion        0.798        0.202
```

DENSIDAD_CIRCULACION

```
describe(accident.train$DENSIDAD_CIRCULACION)
```

```
## accident.train$DENSIDAD_CIRCULACION
##        n  missing distinct
##    19292    10710        3
##
## Value      CONGESTIONADA         DENSA        FLUIDA
## Frequency            308          1479         17505
## Proportion         0.016         0.077         0.907
```

MEDIDAS_ESPECIALES

Describe si se ha tomado alguna medida adicional a la hora del accidente.

```
describe(accident.train$MEDIDAS_ESPECIALES)
```

```
## accident.train$MEDIDAS_ESPECIALES
##        n  missing distinct
##    21327     8675        4
##
```

```
## Value       CARRIL REVERSIBLE HABILITACION ARCEN     NINGUNA MEDIDA
## Frequency                  17                  8             21024
## Proportion              0.001              0.000             0.986
##
## Value             OTRA MEDIDA
## Frequency                 278
## Proportion              0.013
```

Análisis de la variable de salida

El objetivo de la práctica es clasificar los accidentes según las clases que ofrece esta variable. Para ello se ha dispuesto a realizar un análisis más detallado de esta variable. En la *Figura 1* se puede observar como están repartidas las instancias entre las diferentes clases.

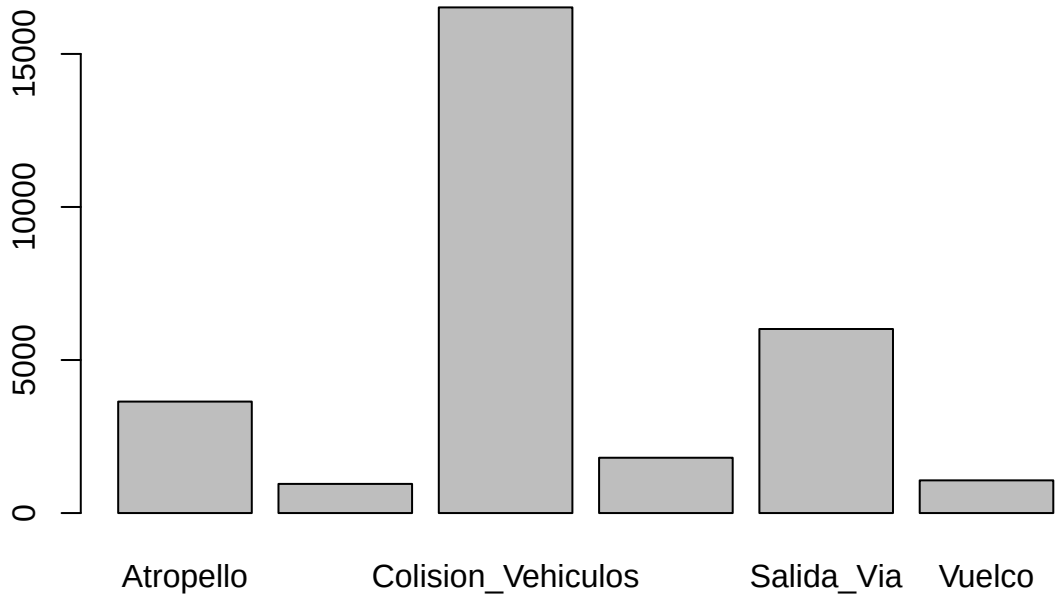

Figure 1: Instancias clasificadas según TIPO_ACCIDENTE

```
describe(accident.train$TIPO_ACCIDENTE)
```

```
## accident.train$TIPO_ACCIDENTE
##        n  missing distinct
##    30002        0        6
##
## Value               Atropello Colision_Obstaculo Colision_Vehiculos
## Frequency                3642               952             16520
## Proportion              0.121             0.032             0.551
##
## Value                    Otro         Salida_Via             Vuelco
## Frequency                1807               6013               1068
## Proportion              0.060              0.200              0.036
```

Al analizar la variable se puede observar claramente que existe un desbalanceo entre las clases siendo la mayoritaria *Colisión de Vehículos* con el 55% de los casos estudiados. Le siguen las clases: *Salida Vía* con el 20% y *Atropello* con el 12%. Las demás tienen un porcentaje inferior al 7%.

Análisis preliminar de los datos

Después de realizar un análisis variable por variable se ha decidido eliminar las variables:

- *CARRETERA*, demasiados NAs.
- *ACOND_CALZADA*, demasiados NAs.
- *TOT_VICTIMAS*, porque es la suma de las variables: *TOT_MUERTOS*, *TOT_HERIDOS_LEVES* y *TOT_HERIDOS_GRAVES*.

```
# Se elimina la variable CARRETERA
full.data$CARRETERA <- NULL
# Se elimina la variable ACOND_CALZADA
full.data$ACOND_CALZADA <- NULL
# Se elimina la variable TOT_VICTIMAS
full.data$TOT_VICTIMAS <- NULL
```

Imputación de valores

Un problema común en los datos suele ser la existencia de valores perdidos. El tratamiento de estos datos se puede tratar desde distintas aproximaciones:

- Usar algunas técnicas de aprendizaje. Por ejemplo, los métodos de clasificación basados en árbol.
- Eliminar instancias que contengan muchos valores perdidos.
- Asignar valores utilizando imputación.

Esta última es la que se va a aplicar en el dataset. Para ello se utilizará el paquete *mice*,que proporciona una implementación del algoritmo *MICE* tal como se describe en Van Buuren y Groothuis-Oudshoorn (2011). Para su ejecución se usarán los siguientes parámetros:

- method = pmm
- m = 10
- maxit = 5

La imputación se hará en dos fases. La primera se aplicará el algoritmo al dataset de "train" y en la segunda parte se realizará al dataset completo.

```
## --- > MICE < --- ##
set.seed(179385)
# Intentar train primero y después juntar el test y volver hacer la imputación
full.train <- full.data[1:30002,]
full.test <- full.data[30003:nrow(full.data),]

#se realiza la imputacion (m = 10)
imputados <-  mice::mice(full.train, m=10, method="pmm", maxit = 5)
# se completa el conjunto de datos con las imputaciones
datosImputados <-  mice::complete(imputados)

# Pasamos a full.data el nuvo data.frame con todos los datos imputados
full.train <- datosImputados
full.data <- rbind(full.train, full.test)

#############################################################################################################
# Volvemos a realizar la imputación, pero esta vez a todo el conjunto quitando la variable clase
# perdidos.

#se realiza la imputacion (m = 10)
```

```
imputados <-  mice::mice(full.data[,-27], m=10, method="pmm", maxit = 5)
# se completa el conjunto de datos con las imputaciones
datosImputados <-  mice::complete(imputados)

full.data[,1:28] <- datosImputados
```

Selección de características

Una vez se han imputado los valores perdidos el siguiente paso es la selección de las variables adecuadas para que resulte el mejor modelo posible.

Selección de varaibles utilizando Random Forest

Método de cálculo de importancia de atributos calculados sobre un modelo usando el algoritmo Random Forest, combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Los nodos del modelo se analizan para aportar información de importancia de cada variable. Además indica que variables utilizar para conseguir el mejor modelo.

```
# Sacado del fichero caret-randomForest.R
set.seed(74749572)
# define el control usando la funcion de seleccion mediante random forest
control <- caret::rfeControl(functions=rfFuncs, method="cv", number=10)
# ejecuta el metodo
results <- caret::rfe(trainData[,1:26], trainData[,27], sizes=c(1:26), rfeControl=control)
```

A continuación se muestran los resultados obtenidos:

```
# muestra los resultados
print(results)
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold)
##
## Resampling performance over subset size:
##
##  Variables Accuracy  Kappa AccuracySD KappaSD Selected
##          1   0.7286 0.5389    0.01060 0.02360
##          2   0.8175 0.6960    0.02452 0.04218
##          3   0.8175 0.6959    0.02452 0.04205
##          4   0.8165 0.6942    0.02273 0.03907
##          5   0.8184 0.6975    0.02294 0.03932
##          6   0.8174 0.6961    0.02159 0.03711
##          7   0.8165 0.6943    0.02237 0.03836
##          8   0.8165 0.6944    0.02196 0.03722
##          9   0.8184 0.6978    0.02132 0.03670
##         10   0.8174 0.6963    0.02105 0.03609
##         11   0.8165 0.6945    0.02318 0.04049
##         12   0.8145 0.6913    0.02278 0.03924
##         13   0.8144 0.6914    0.02258 0.03933
##         14   0.8164 0.6948    0.02251 0.03932
```

```
##         15   0.8145 0.6915    0.02104 0.03705
##         16   0.8096 0.6840    0.01890 0.03342
##         17   0.8136 0.6900    0.02630 0.04444
##         18   0.8136 0.6899    0.02621 0.04500
##         19   0.8136 0.6898    0.02356 0.04005
##         20   0.8175 0.6961    0.02657 0.04568
##         21   0.8214 0.7024    0.02900 0.04951        *
##         22   0.8214 0.7023    0.02376 0.04067
##         23   0.8194 0.6991    0.02116 0.03633
##         24   0.8185 0.6975    0.02237 0.03858
##         25   0.8214 0.7023    0.02206 0.03789
##         26   0.8194 0.6991    0.02347 0.04004
##
## The top 5 variables (out of 21):
##    TOT_VEHICULOS_IMPLICADOS, ZONA, ZONA_AGRUPADA, HORA, PRIORIDAD
```

```r
# muestra las caracteristicas elegidas
conjVariables <- predictors(results)
```

En la *Figura 2* se muestra el número de variables que el algoritmo ha considerado como las mejores para realizar el ajuste.

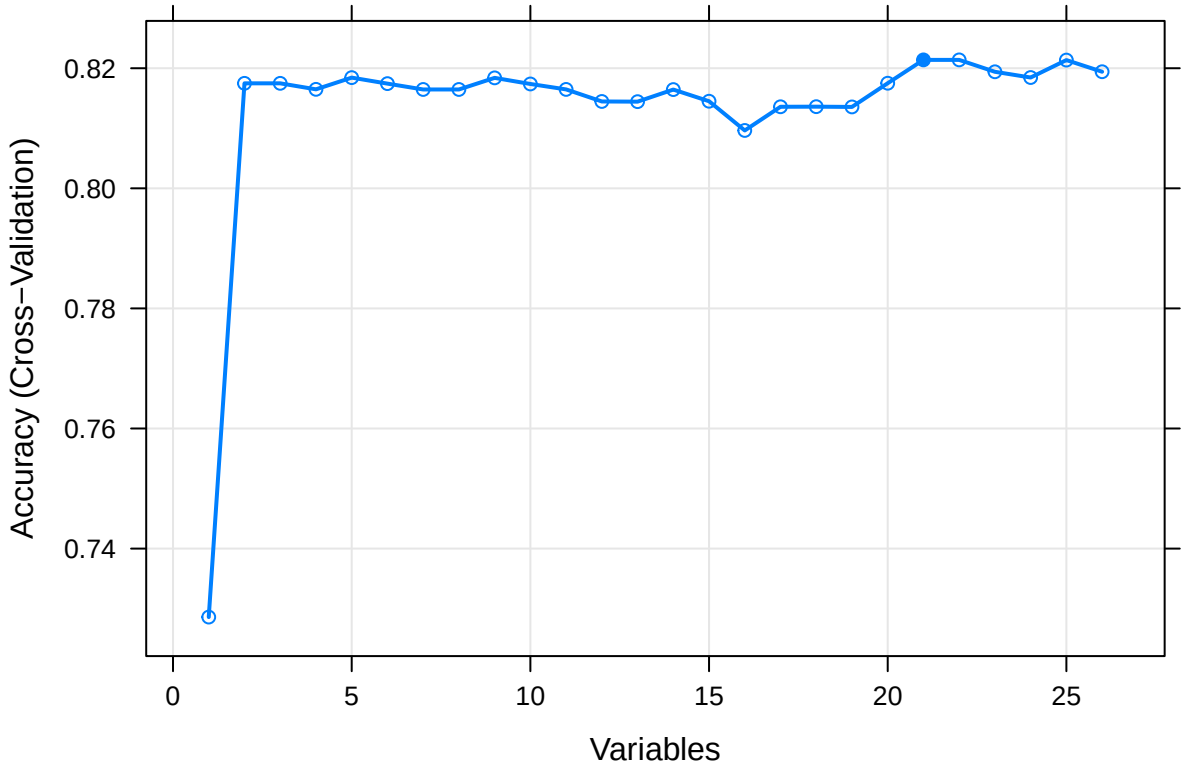

Figure 2: Gráfica donde se muestra el número de variables adecuadas para el ajuste

Variables Correladas

En primer lugar se mira la correlación entre las variables predictoras y la salida. Para ello se ha creado una función, *filterMethods* que devuelve una matriz con diferentes medidas de correlación para variables categóricas.

```
## Crear la matriz y el caluculo de los estadísticos a partir de una función
filterMethods <- function(vs, df){
  w.chi.squared <- FSelector::chi.squared(vs~., df)
  w.gain.ratio <-  gain.ratio(vs~., df)
  w.information.gain <- FSelector::information.gain(vs~., df)
  w.symetrical <-  symmetrical.uncertainty(vs~., df)
  w.oneR <- FSelector::oneR(vs~., df)

  mat <- as.matrix(w.chi.squared)
  mat <- cbind(mat, w.gain.ratio)
  mat <- cbind(mat, w.information.gain)
  mat <- cbind(mat, w.symetrical)
  mat <- cbind(mat, w.oneR)
  names(mat) <- c("chi.squared", "gain.ratio", "information.gain", "symetrical", "oneR")
  return(mat)
}

###### ------------ CORRELACIÓN PARA SELECCIONAR VARIABLES ------------------
matrix.all.result <- filterMethods(accident.train$TIPO_ACCIDENTE, accident.train)
head(matrix.all.result)
```

```
##                    chi.squared   gain.ratio information.gain  symetrical
## ANIO                0.00000000 0.0000000000      0.000000000 0.000000000
## MES                 0.03094819 0.0009591009      0.002382215 0.001257794
## HORA                0.08687182 0.0114549085      0.016926571 0.012169534
## DIASEMANA           0.06981337 0.0060174810      0.011684879 0.007199673
## PROVINCIA           0.14327070 0.0156839612      0.050353152 0.022306748
## COMUNIDAD_AUTONOMA  0.12198994 0.0158565275      0.036747025 0.020293284
##                        oneR
## ANIO                487.4126
## MES                 482.4646
## HORA                484.9513
## DIASEMANA           484.6924
## PROVINCIA           462.0844
## COMUNIDAD_AUTONOMA 479.0556
```

Para la selección de características de todas las medidas de correlación se ha utilizado *chi cuadrado*.

A continuación se mide la correlación entre variables mediante la función *compareItems* y *correlationExit*:

```
## Función para comparar la correlación existente entre dos variables
compareItems <- function(x, y){
  mat <- matrix(data = c(matrix.all.result[x,], matrix.all.result[y,]),
                nrow = 2,
                byrow = T,
                dimnames = list(c(x, y), names(matrix.all.result)))
}

## Función para calcular la correlación entre dos variables
correlationExit <- function(value, vs, dataset){
  w.chi <- FSelector::chi.squared(vs~value,dataset)
  w.gain <- FSelector::gain.ratio(vs~value,dataset)
  w.information <- FSelector::information.gain(vs~value,dataset)
  w.symetrical <- FSelector::symmetrical.uncertainty(vs~value,dataset)
  x <- c(w.chi, w.gain, w.information, w.symetrical)
```

```
  names(x) <- c("chi.squared", "gain.ratio", "information.gain", "symetrical")
  return(x)
}
```

Se aplica esta función a cada una de las variables. En el siguiente fragmento de código sólo se muestran algunos ejemplos.

```
###### ------------ CORRELACIÓN PARA QUITAR VARIABLES ------------------
# ZONA ~ ZONA_AGRUPADA
w.ZONA <- correlationExit(accident.train$ZONA_AGRUPADA, accident.train$ZONA, accident.train)
w.ZONA
```

```
## $chi.squared
## [1] 1
##
## $gain.ratio
## [1] 1
##
## $information.gain
## [1] 0.6869674
##
## $symetrical
## [1] 0.9620051
```

```
# TRAZADO_NO_INTERSEC ~ TIPO_INTERSEC
w.INTERSEC <- correlationExit(accident.train$TRAZADO_NO_INTERSEC, accident.train$TIPO_INTERSEC, acciden
w.INTERSEC
```

```
## $chi.squared
## [1] 0.4466049
##
## $gain.ratio
## [1] 0.548521
##
## $information.gain
## [1] 0.6528263
##
## $symetrical
## [1] 0.5581687
```

```
##RELACION INTERSECION PROVINCIA/COMUNIDAD AUTONOMA
w.PROVINCIA.COM <- correlationExit(accident.train$PROVINCIA, accident.train$COMUNIDAD_AUTONOMA)
w.PROVINCIA.COM
```

```
## $chi.squared
## [1] 1
##
## $gain.ratio
## [1] 0.7218437
##
## $information.gain
## [1] 2.31747
##
## $symetrical
## [1] 0.8384544
```

```
# PROVINCIA ~ COMUNIDAD_AUTONOMA
comparePyC <- compareItems("PROVINCIA", "COMUNIDAD_AUTONOMA")
comparePyC
```

```
##                    chi.squared gain.ratio information.gain symetrical
## PROVINCIA          0.1432707   0.01568396 0.05035315       0.02230675
## COMUNIDAD_AUTONOMA 0.1219899   0.01585653 0.03674703       0.02029328
##                    oneR
## PROVINCIA          462.0844
## COMUNIDAD_AUTONOMA 479.0556
```

Transformación a algunas variables

Además de calcular la correlación entre variables y la importancia de estas respecto a la salida, también se han realizado cambios en variables como la *HORA*. En este caso se discretiza la variable.

```
# Pasamos la hora a entero
full.data$HORA <- trunc(full.data$HORA)
```

El siguiente paso es la construcción de modelos teniendo en cuenta los resultados obtenidos a partir de los calculos de correlación y de importancia de las variables realizado.

Modelos

Para realizar los modelos se ha utilizado el conjunto de entrenamiento y de ahí se ha seleccionado el conjunto de variables que se ha obtenido en el puto anterior.

```
sub.full.data <- full.data[,conjVariables]
sub.full.data$TIPO_ACCIDENTE <- full.data$TIPO_ACCIDENTE
trainData <- sub.full.data[1:30002,]
```

Después de probar con varios algoritmos el que mejor resultados da es el algoritmo *Random Forest*. Una vez seleccionado el algoritmo se construyen varios modelos alternando variables dependiendo de los cálculos realizados anteriormente.

El primero modelo es con todas las varaibles seleccionadas:

```
# ***** Random Forest *****
rf.model.29M.var <- randomForest::randomForest(TIPO_ACCIDENTE ~ ., data=trainData, ntree=500)
```

Se muestran los resultados obtenidos para el modelo *rf.model.29M.var*.

```
print(rf.model.29M.var)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = trainData,      ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.05%
## Confusion matrix:
##                    Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello               3063                 7                134   23
## Colision_Obstaculo       278                49                366   26
## Colision_Vehiculos         1                 0              16514    2
```

```
## Otro                     604                12                745   91
## Salida_Via               389                 3                461   20
## Vuelco                   310                11                190   17
##                    Salida_Via Vuelco  class.error
## Atropello                 404     11 0.1589785832
## Colision_Obstaculo        232      1 0.9485294118
## Colision_Vehiculos          3      0 0.0003631961
## Otro                      339     16 0.9496402878
## Salida_Via               5119     21 0.1486778646
## Vuelco                    490     50 0.9531835206
```

```
rf.importancia.29M.var <- randomForest::importance(rf.model.29M.var)
rf.importancia.29M.var
```

```
##                          MeanDecreaseGini
## TOT_VEHICULOS_IMPLICADOS        7549.6941
## ZONA                             497.6287
## ZONA_AGRUPADA                    448.7911
## HORA                            1021.9966
## PRIORIDAD                        626.0012
## TIPO_VIA                         483.9410
## TRAZADO_NO_INTERSEC              574.4463
## TIPO_INTERSEC                    408.9798
## RED_CARRETERA                    334.2343
## DENSIDAD_CIRCULACION             182.5221
## TOT_HERIDOS_LEVES                381.3644
## SUPERFICIE_CALZADA               312.6819
## COMUNIDAD_AUTONOMA               595.3067
## FACTORES_ATMOSFERICOS            231.9064
## OTRA_CIRCUNSTANCIA               185.2197
## ACERAS                           163.1862
## LUMINOSIDAD                      354.1605
## VISIBILIDAD_RESTRINGIDA          201.4701
## DIASEMANA                        805.0109
## TOT_HERIDOS_GRAVES               168.1129
## PROVINCIA                        925.1752
```

En la *Figura 3* se muestra ordenadamante la importancia de las variables para la producción del modelo.

El siguiente modelo realizado es quitar al dataset una de las variables más correladas entre ellas, en este caso *ZONA*:

```
# ***** Random Forest *****
rf.model.30M.Z <- randomForest::randomForest(TIPO_ACCIDENTE ~ ., data=trainData, ntree=1500)
```

Se muestran los resultados obtenidos para el modelo *rf.model.30MZ*.

```
print(rf.model.30M.Z)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = trainData,      ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.06%
```

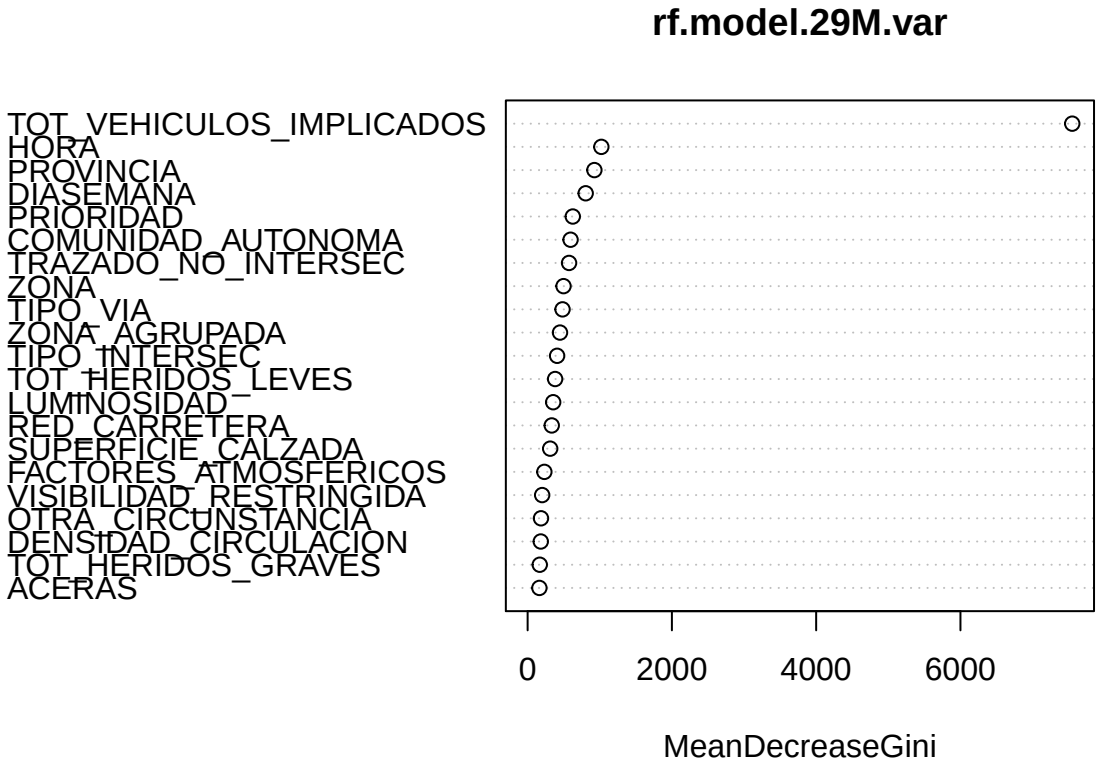

Figure 3: Importancia de las variables para el modelo rf.model.29M.var

```
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3057                  7                134   30
## Colision_Obstaculo        286                 53                366   28
## Colision_Vehiculos          1                  0              16515    1
## Otro                      603                 10                744   97
## Salida_Via                385                  4                461   22
## Vuelco                    310                 13                190   20
##                     Salida_Via Vuelco  class.error
## Atropello                  401     13 0.1606260297
## Colision_Obstaculo         216      3 0.9443277311
## Colision_Vehiculos           3      0 0.0003026634
## Otro                       338     15 0.9463198672
## Salida_Via                5111     30 0.1500083153
## Vuelco                     483     52 0.9513108614
```

```
rf.importancia.30M.Z <- randomForest::importance(rf.model.30M.Z)
rf.importancia.30M.Z
```

```
##                           MeanDecreaseGini
## TOT_VEHICULOS_IMPLICADOS        7543.0695
## ZONA                             456.6594
## ZONA_AGRUPADA                    455.1822
## HORA                            1015.2677
## PRIORIDAD                        617.3041
## TIPO_VIA                         513.9105
## TRAZADO_NO_INTERSEC              532.7025
```

```
## TIPO_INTERSEC                   440.1644
## RED_CARRETERA                   365.5110
## DENSIDAD_CIRCULACION            184.4039
## TOT_HERIDOS_LEVES               381.4746
## SUPERFICIE_CALZADA              314.1549
## COMUNIDAD_AUTONOMA              594.5648
## FACTORES_ATMOSFERICOS           230.9608
## OTRA_CIRCUNSTANCIA              183.9614
## ACERAS                          161.7210
## LUMINOSIDAD                     354.8714
## VISIBILIDAD_RESTRINGIDA         202.9174
## DIASEMANA                       805.3445
## TOT_HERIDOS_GRAVES              167.9851
## PROVINCIA                       924.6529
```

La importancia de las variables se puede observar en la *Figura 4*.

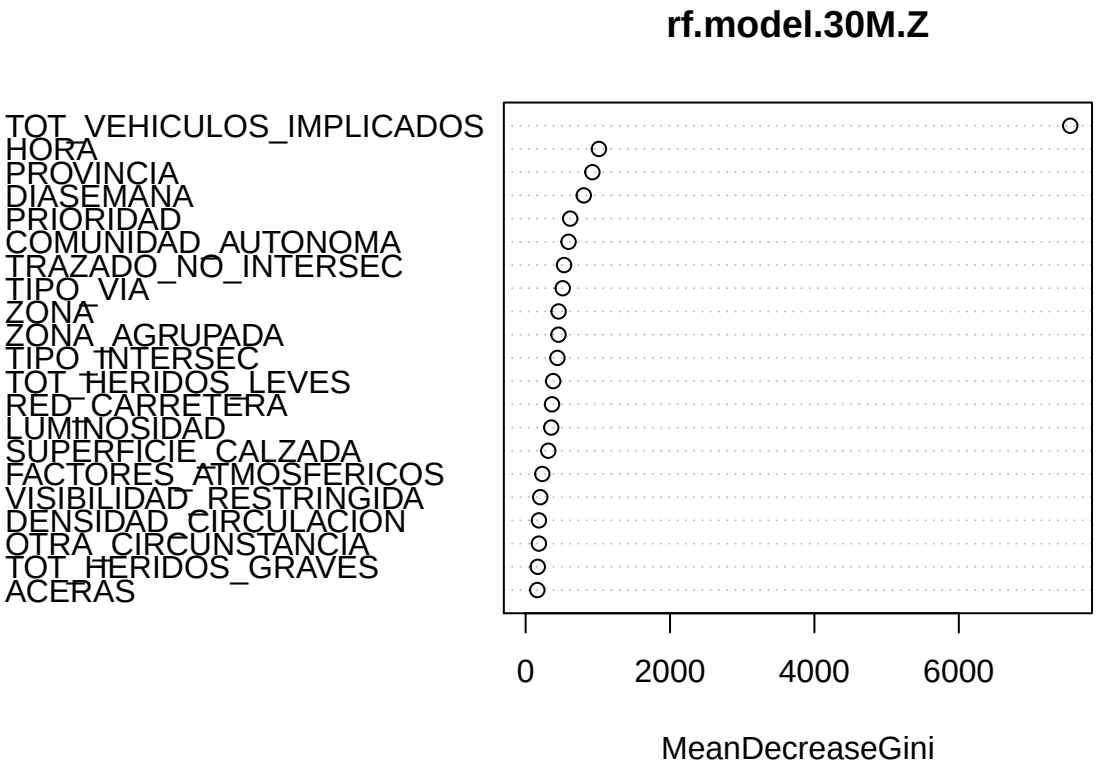

Figure 4: Importancia de las variables para el modelo rf.model.30M.Z

Otro modelo propuesto es el utilizando el mismo dataset que en los anteriores pero eliminando las variables *ZONA* y *COMUNIDAD_AUTONOMA*.

```
# ***** Random Forest *****
rf.model.30MZCA <- randomForest::randomForest(TIPO_ACCIDENTE ~ ., data=trainData, ntree=500)
```

Se muestran los resultados obtenidos para el modelo *rf.model.30MZCA*.

```
print(rf.model.30MZCA)
```

```
##
## Call:
##  randomForest(formula = TIPO_ACCIDENTE ~ ., data = trainData,      ntree = 500)
```

```
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 17.05%
## Confusion matrix:
##                     Atropello Colision_Obstaculo Colision_Vehiculos Otro
## Atropello                3053                 10                134   24
## Colision_Obstaculo        284                 50                366   24
## Colision_Vehiculos          1                  0              16516    0
## Otro                      598                 10                744   92
## Salida_Via                377                  4                461   26
## Vuelco                    310                 12                190   20
##                     Salida_Via Vuelco  class.error
## Atropello                  409     12 0.1617243273
## Colision_Obstaculo         225      3 0.9474789916
## Colision_Vehiculos           3      0 0.0002421308
## Otro                       346     17 0.9490868843
## Salida_Via                5118     27 0.1488441710
## Vuelco                     477     59 0.9447565543
```

```
rf.importancia.30MZCA <- randomForest::importance(rf.model.30MZCA)
rf.importancia.30MZCA
```

```
##                           MeanDecreaseGini
## TOT_VEHICULOS_IMPLICADOS        7570.7914
## ZONA                             453.6962
## ZONA_AGRUPADA                    503.4900
## HORA                            1015.1819
## PRIORIDAD                        607.9010
## TIPO_VIA                         481.2252
## TRAZADO_NO_INTERSEC              550.8613
## TIPO_INTERSEC                    415.2585
## RED_CARRETERA                    350.4974
## DENSIDAD_CIRCULACION             186.3455
## TOT_HERIDOS_LEVES                381.4240
## SUPERFICIE_CALZADA               315.2411
## COMUNIDAD_AUTONOMA               592.3210
## FACTORES_ATMOSFERICOS            232.1587
## OTRA_CIRCUNSTANCIA               183.9331
## ACERAS                           168.9988
## LUMINOSIDAD                      350.2444
## VISIBILIDAD_RESTRINGIDA          203.2547
## DIASEMANA                        810.0044
## TOT_HERIDOS_GRAVES               167.5109
## PROVINCIA                        916.8417
```

El orden de importancia de las variables se muestra en la *Figura 5*.

Conclusiones modelos realizados

Se puede observar que los modelos con mejor resultados son:

- *rf.model.29M.var*
- *rf.model.30MZCA*

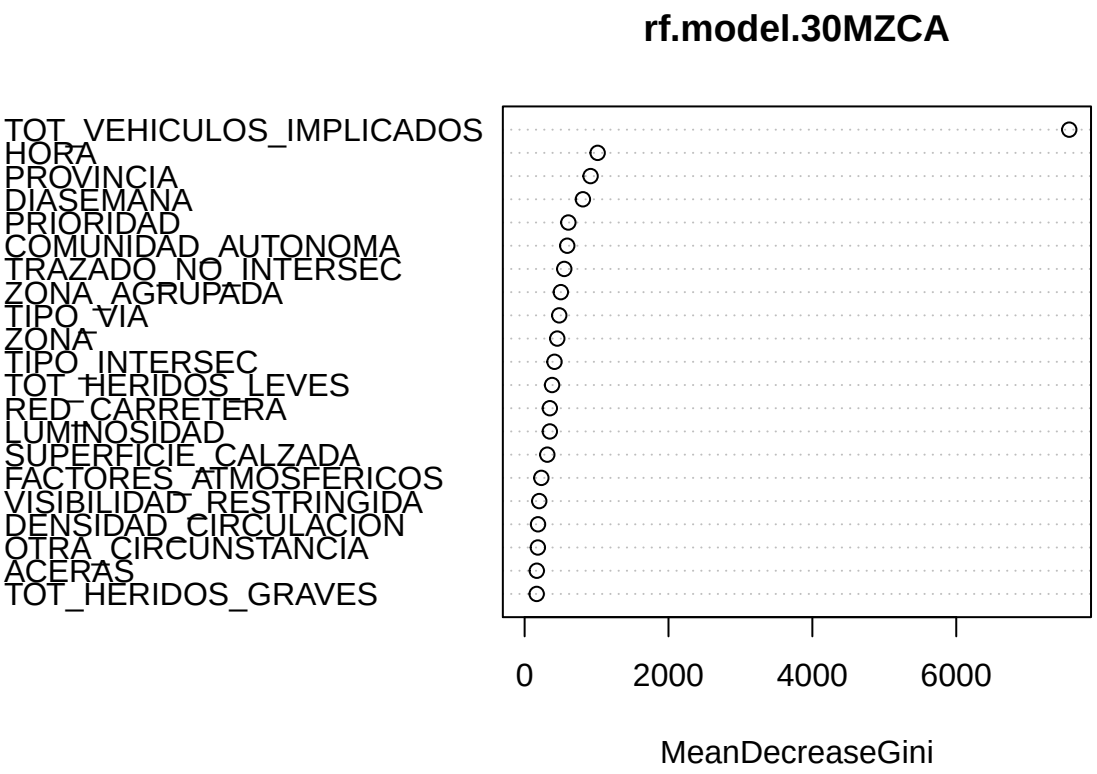

Figure 5: Importancia de las variables para el modelo rf.model.30M.Z

Ambos con un error del 17.05. El modelos *rf.model30MZ* es una décima peor, pero a su vez es el que mejor resultados da en kaggle.

Kaggle

Para subir los resultados obtenidos a kaggle se ha de realizar una predicción y extraer la clase obtenida en un fichero con extensión *csv*. Se puede ver el ejemplo del *rf.model.30M.Z* en el siguiente fragmento de código:

```
# Creamos la predicción para la entrega en kaggle
prediction <- predict(rf.model.30M.Z, full.final.test)
full.final.test$id <- seq.int(nrow(full.final.test))
submit <- data.frame(id = full.final.test$id, Prediction = prediction)
write.csv(submit, file = "./resultados/result30M.Z.csv", row.names = FALSE)
```

Subidas kaggle

Para esta práctica se han realizado 26 entradas en kaggle siendo las primeras modelos generales con diferentes algoritmos para ver como funcionaba la plataforma y como se adaptaban diferentes modelos al dataset.

Los modelos que se han explicado en el informe son aquellos con mejores puntuación en la clasificación final.

El resto de subidas fueron cambios realizados en el dataset y en los modelos para intentar llegar a 0.83 en la clasificación pública, hecho que nunca se produzco.

Para la competición se seleccionaron 5 modelos, 4 de ellos por intentar llegar a 0.83 estaban sobreajustados. El modelo *rf.model.29M.var* siendo el peor resultado de los modelos elegidos en la clasificación pública con

0.82909, dio el mejor de los elegidos en la clasificación privada con 0.82665, siendo a su vez el quinto mejor modelo generado de todos los realizados.

En la *Figura 6* se puede observar el modelo con mejor puntuación obtenida en la competición y el modelo con mejor puntuación en la clasificación pública claramente sobreajustado.

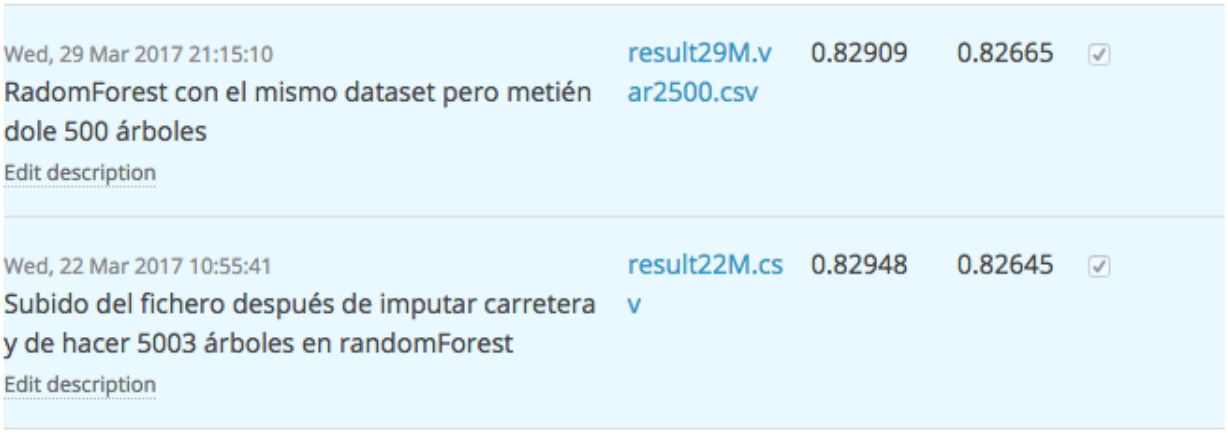

Figure 6: Mejor puntuación en la clasificación privada y pública en kaggle

El modelo con mejor puntuación de todos fue *rf.model.30M.Z* con una puntuación de 0.82899 en la clasificación pública y 0.82817 en la privada. Aunque no fue elegido entre los 5 mejores. En la *Figura 7* se puede ver la puntuación obtenida en kaggle.

Figure 7: Mejor puntuación privada en kaggle

Conclusiones

Conclusiones técnicas

Los modelos con mejores puntuaciones han sido aquellos en los que he quitado y transformado variables. Es importante recalcar que en estos modelos la diferencia entre la clasificación pública y privada ha sido mínima, muestra de que el modelo se estaba ajustando bien al problema.

Sin embargo, los modelos con un número excesivo de árboles y con más varaibles, siendo los que mejor resultado me daban en la clasificación pública, sufrieron un descenso considerable al desbloquearse la clasificación privada.

Conclusiones personales

Como puntos negativos tengo que decir que me dejé llevar por el ímpetú de la competición y sobreajusté demasido los modelos intentando mejorar la puntuación más que por realizar una buena predicción. Este hecho produzco que eligiera mal los modelos dejando 4 modelos por fuera mejores que los seleccionados.

Por otro lado he aprendido a enfrentarme a un dataset. Además he conocido el funcionamiento de la herramienta kaggle y la importancia que tiene dentro de la ciencia de datos.