Universitat Oberta de Catalunya

uoc.edu

Predicción de la gravedad de los heridos en accidentes de tráfico en Barcelona

David Vila Giménez
Máster Universitario en Ciencia de Datos
Área 2

Raul Parada Medina
Jordi Casas Roma

01/2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Predicción de la gravedad de los heridos en accidentes de tráfico en Barcelona*
Nombre del autor:	*David Vila Giménez*
Nombre del consultor/a:	*Raul Parada Medina*
Nombre del PRA:	*Jordi Casas Roma*
Fecha de entrega (mm/aaaa):	01/2020
Titulación:	*Máster Universitario de Ciencia de Datos*
Área del Trabajo Final:	*Área 2*
Idioma del trabajo:	*Castellano*
Palabras clave	*Análisis predictivo, gravedad accidentes, Smart City*

Resumen del Trabajo

¿Podemos realmente prevenir un accidente de tráfico?, y, en consecuencia, ¿está en nuestras manos evitar las víctimas derivadas? La respuesta a estas preguntas es compleja ya que los accidentes están rodeados de un gran número (prácticamente infinito) de variables independientes y, a simple vista, aleatorias.

Desde hace años, los investigadores se han centrado en utilizar técnicas de Minería de Datos para poder extraer conclusiones sobre las causas de los accidentes de tráfico y así aplicar medidas preventivas futuras. De los múltiples estudios posibles en este ámbito, el trabajo actual se centrará en las víctimas derivadas. Se pretende generar modelos de predicción que permitan predecir cuál será la gravedad de las personas involucradas en un accidente. Para ello, se estudiará el área de la ciudad de Barcelona y a partir de diferentes factores se generarán modelos de regresión logística y random forest, los cuales categorizarán la gravedad de las víctimas.

Abstract

Can we really prevent a traffic accident? and, consequently, is it in our hands to avoid the derived victims? The answer to these questions is complex since accidents are surrounded by a large (practically infinite) number of independent and, at first glance, random variables.

For years, researchers have focused on using data mining techniques to draw conclusions about the causes of traffic accidents and thus apply future preventive measures. Of the many possible studies in this area, current work will focus on derived victims. It is intended to generate prediction models that allow predicting the severity of the people involved in an accident. For this, the area of the city of Barcelona will be studied and, based on different factors, models of logistic regression and random forest will be generated, which will categorize the severity of the victims.

Índice

Índice de Figuras

Índice de Tablas

1. Introducción

En todo proyecto de Minería de Datos es esencial conocer el ámbito del problema que se presenta. Es por eso por lo que en este primer capítulo se aportará información sobre cómo han evolucionado los accidentes de tráfico en la ciudad objeto de estudio y, por consiguiente, cómo ha evolucionado la gravedad de las víctimas derivadas.

La segunda parte de este capítulo se centra más en la organización y planificación del proyecto y en los requisitos a nivel de software que son necesarios. Finalmente se definirán los objetivos, y con ellos, los productos que se pretenden obtener al concluir el proyecto.

1.1. Contexto y justificación del Trabajo

Los informes extraídos anualmente por diferentes organizaciones sobre los accidentes de tráfico son extremadamente alarmantes. Según la Organización Mundial de la Salud, cerca de 1.3 millones de personas pierden sus vidas en las carreteras de todo el mundo, siendo la principal causa de muerte entre los jóvenes de 15 y 29 años. Además, es la primera causa de muerte no relacionada con enfermedades, superando a las víctimas producidas por el VIH o la tuberculosis. A parte de ser un problema social, cabe destacar que los accidentes de vehículos tienen una repercusión económica del 1% al 3% en el PIB respecto cada país, ascendiendo a un total de $500.000 millones. Así pues, la reducción del número de heridos y muertos por accidentes de tráfico mitigará el sufrimiento, desencadenará el crecimiento y liberará recursos para la una utilización más productiva.

El caso del estado español es muy diferente. El año 2018 se cerró con un total de 1.806 muertos, 24 víctimas menos que en el 2017. Esto le hace situarse como el séptimo país europeo con menor tasa de muerte, estando muy por debajo de la media en la Unión Europea. Y es que, tal y como se aprecia en la **Figura 1-1**, desde principios de los años 90, los cuales fueron los más trágicos de los últimos 60 años, el número de fallecidos no ha hecho más que disminuir. Esto es debido a una fuerte inversión realizada por el país en la prevención de accidentes, aplicando políticas como: carné por puntos, radares móviles, etc.

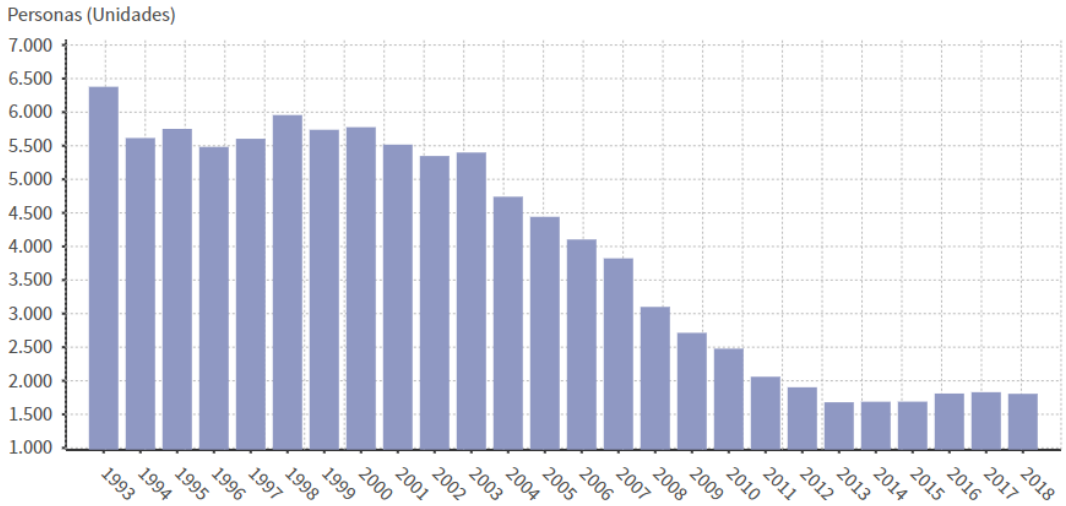

Figura 1-1. Evolución de los accidentes de tráfico en España en vías urbanas e interurbanas. (fuente: DGT, www.epdata.es)

Desgraciadamente, se puede apreciar cómo nos encontramos en una situación de estancamiento desde el año 2011 donde los datos oscilan alrededor de las 1.700 – 1.900 víctimas mortales. Esta situación también se detecta en las ciudades del país. En el caso de Barcelona, ciudad donde se centrará el estudio del Trabajo, la pendiente decreciente desde el año 1990 es menos evidente, pero se produce un descenso del número de víctimas de un 77% aproximadamente tal y como se muestra en la Figura **1-2**. Además, según datos del ayuntamiento, pese a incrementarse el número de muertos, el año 2018 ha sufrido un 3% menos de accidentes que el 2017.

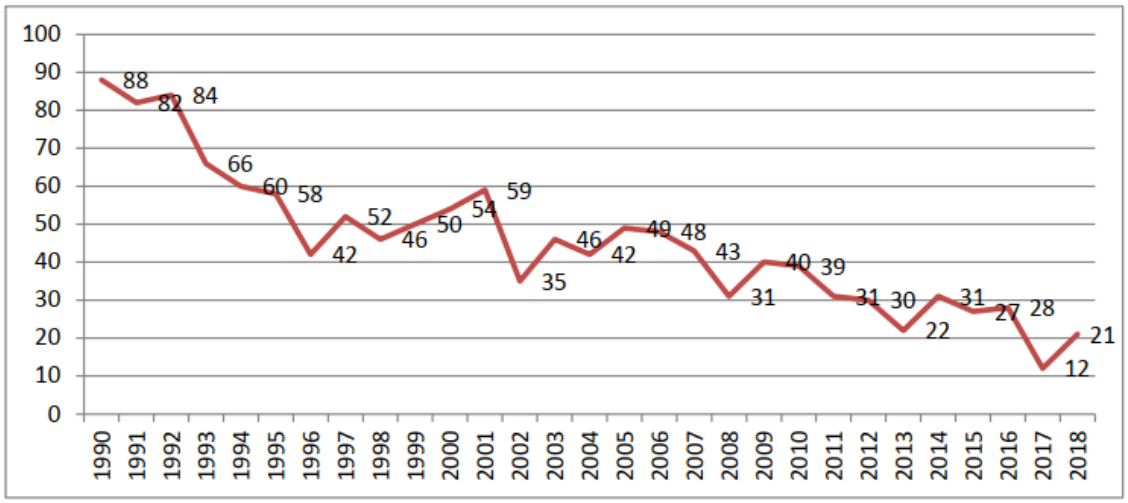

Figura 1-2. Evolución de los accidentes de tráfico en Barcelona. (fuente: Ajuntament de Barcelona)

Es evidente que los múltiples esfuerzos realizados en los últimos años han dado sus frutos en la reducción de siniestros. De todas maneras, la situación de estancamiento y la todavía elevada cifra de muertes han motivado a realizar el presente trabajo. Se intentará dar una vuelta de tuerca más en la explotación de las diferentes vías existentes de estudio sobre la severidad de heridos en los accidentes de tráfico y que sirva de complemento a las medidas tradicionales. Concretamente se pretende crear un

proyecto de Minería de Datos, generando un modelo predictivo que permita deducir la gravedad de los heridos derivados de un accidente.

1.2. Objetivos del Trabajo

1.2.1. Objetivos principales

El área de trabajo se centrará en la ciudad de Barcelona. El ayuntamiento publica de forma abierta múltiples datos sobre la ciudad en el portal web https://opendata-ajuntament.barcelona.cat/. Entre todos los datos, se pueden encontrar los relacionados con accidentes de tráfico desde el año 2010. A partir de esta y otras fuentes de datos que se explicarán más detalladamente en capítulos posteriores, se pretende generar diferentes modelos predictivos de regresión logística y random forest que permitan determinar la gravedad de los heridos implicados en un accidente. De esta manera, los cuerpos de seguridad y sanidad tendrán a su disposición una herramienta que les permita, en primer lugar, conocer los factores que influyen en la gravedad de los heridos para así actuar sobre ellos, y, en segundo lugar, en caso de que se produzca el accidente, poder estar prevenidos para una actuación más eficiente.

1.2.2. Objetivos secundarios

Para alcanzar el objetivo principal se irán realizando pequeños hitos u objetivos. De manera muy resumida se describen como:

- Adquisición de los datos. Se fijará una ventana temporal de dos años (2017 - 2018) para disponer de un número elevado de registros. A parte de los datos obtenidos del web Open Data de Barcelona, se realizará una búsqueda exhaustiva sobre todo tipo de información que pueda ser relevante en este proyecto como son datos meteorológicos o festivos.

- Procesado de los datos. Dado la naturaleza desagregada de los datos, será necesario realizar un proceso de unificación lo cual implicará modificar algunas variables. Además, se realizarán los procesos habituales en cualquier proyecto de Minería de Datos como revisar la consistencia de los datos, búsqueda de valores nulos, conversión de valores, … así como la elección de aquellos datos necesarios.

- Análisis descriptivo de los datos. Con el conjunto de datos adecuado, se procederá a realizar un estudio simple sobre su contenido. En este paso se generarán gráficas que permitan comprender mejor la naturaleza de los datos.

- Modelado y evaluación. A partir de los datos de entrenamiento y la modificación de diferentes parámetros se procederá a la creación de los modelos predictivos. Los datos se test se utilizarán para la evaluación de cada uno de los modelos extraídos.

- Conclusiones. Se analizará el producto final obtenido y se procederá a valorar la calidad del mismo.

1.3. Enfoque y método seguido

1.3.1. Metodología

Se seguirá la metodología Cross-Industry Standard Process for Data Mining (CRISP-DM) la cual está enfocada a proyectos de minería de datos como el que nos ocupa.

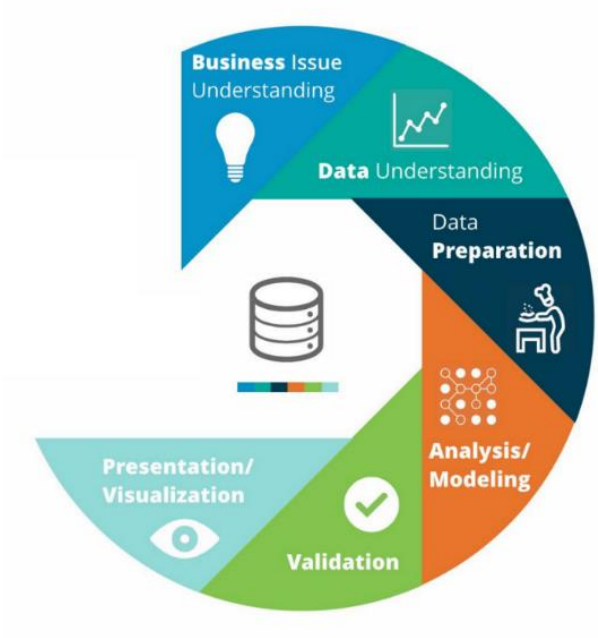

Figura 1-3. Esquema conceptual del modelo CRISP-DM.

Su principal ventaja es el retorno a fases anteriores en caso de detectar cualquier anomalía. La parte más crítica de este proyecto es la del modelado. La utilización de los atributos adecuados será clave en la obtención de buenos resultados. En caso de observa problemas en la fase de modelado, será posible regresar a la preparación de los datos para adaptarlos correctamente. Las fases que componen el esquema conceptual corresponden aproximadamente a los objetivos secundarios descritos con anterioridad.

1.3.2. Entorno de trabajo

Para este proyecto se hará uso de diferentes herramientas. Em primer lugar, para el tratamiento de datos, creación de datos y computación estadística se utilizará el lenguaje *R*. El entorno integrado de desarrollo será *RStudio* y se estructurará el proyecto en un archivo *RMarkdown*.

Carto es un software de código abierto construido sobre *PostGIS* y *PostgreSQL* el cual se utilizará para crear visualizaciones de los accidentes de tráfico sobre el mapa de Barcelona.

1.4. Planificación del Trabajo

Para planificar el trabajo, se ha creado un diagrama de Gantt estructurado en los diferentes entregables (PACs).

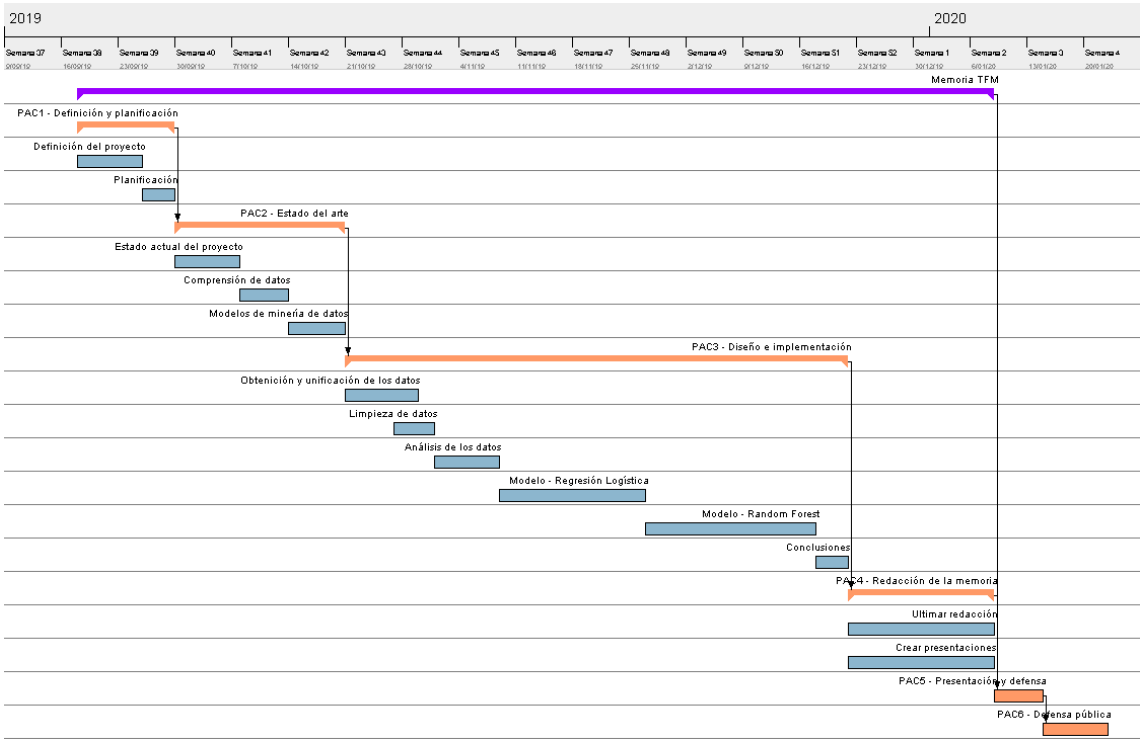

Figura 1-4. Planificación del Trabajo mediante un diagrama de Gantt - Gantt Project.

En la siguiente tabla se aprecian mejor los objetivos de cada PAC y los plazos temporales.

Nombre	Fecha inicio	Fecha fin	Duración
Memoria TFM	**18/09/2019**	**08/01/2020**	**113**
PAC1 - Definición y planificación	18/09/2019	29/09/2019	12
Definición del proyecto	18/09/2019	25/09/2019	8
Planificación	26/09/2019	29/09/2019	4
PAC2 - Estado del arte	30/09/2019	20/10/2019	21
Estado actual del proyecto	30/09/2019	07/10/2019	8
Comprensión de datos	08/10/2019	13/10/2019	6
Modelos de minería de datos	14/10/2019	20/10/2019	7
PAC3 - Diseño e implementación	21/10/2019	21/12/2019	62
Obtención y unificación de los datos	21/10/2019	29/10/2019	9
Limpieza de datos	27/10/2019	31/10/2019	5
Análisis de los datos	01/11/2019	08/11/2019	8
Modelo – Regresión Logística	09/11/2019	26/11/2019	18
Modelo – Random Forest	27/11/2019	17/12/2019	21
Conclusiones	18/12/2019	21/12/2019	4
PAC4 - Redacción de la memoria	22/12/2019	08/01/2020	18
Ultimar redacción	22/12/2019	08/01/2020	18
Crear presentaciones	22/12/2019	08/01/2020	18
PAC5 - Presentación y defensa	09/01/2020	14/01/2020	6
PAC6 - Defensa pública	15/01/2020	22/01/2020	8

Tabla 1-1. Planificación del proyecto.

1.5. Sumario de productos obtenidos

La idea inicial es obtener dos modelos predictivos para poder evaluar las diferencias en sus resultados: regresión logística y random forest. Como se observará en capítulos

anteriores, durante el análisis de los datos se comprobará como existe la problemática de desbalanceo de clases. Esto provocará la implementación en cada modelo, de diferentes técnicas de balanceo de clases: Upsampling, Downsampling, SMOTE y ROSE. De esta manera se generarán hasta diez modelos predictivos. A continuación, se listan:

Modelo	***Técnica aplicada sobre train***
Regresión Logística	Ninguna – Clases desbalanceadas
Regresión Logística	Upsampling
Regresión Logística	Downsampling
Regresión Logística	SMOTE
Regresión Logística	ROSE
Random Forest	Ninguna – Clases desbalanceadas
Random Forest	Upsampling
Random Forest	Downsampling
Random Forest	SMOTE
Random Forest	ROSE

Tabla 1-2. Modelos que se generarán en el Trabajo.

1.6. Descripción de los capítulos del Trabajo

Capítulo 1. Introducción.
Se analiza el ámbito de trabajo y se establecen las bases del proyecto. Finalmente se describen los productos que se deberán obtener al finalizar.

Capítulo 2. Estado del arte.
Se realizará un estudio general sobre qué trabajos y aplicaciones se han realizado relacionados en el ámbito de este proyecto.

Capítulo 3. Datos de interés.
Se explicarán las fuentes de datos utilizadas para la extracción de información, así como una descripción de todas las variables encontradas y que son susceptibles a ser utilizadas en el trabajo.

Capítulo 4. Procesado de los datos.
Se procede a unificar y procesar todos los datos que serán utilizados para la creación de los modelos.

Capítulo 5. Estudio descriptivo de los datos.
Con los datos correctamente formateados y limpios, se procederá a analizarlos descriptivamente para poder extraer las primeras conclusiones antes de generar los modelos.

Capítulo 6. Preparación de los datos.
En este capítulo se extraerán los conjuntos de entrenamiento y test utilizados para crear los modelos y evaluarlos. Habitualmente esto procesos se incluyen junto los modelados, pero en este caso, como se explicará detalladamente, el hecho de trabajar con clases desbalanceadas implicará aplicar técnicas de balanceo sobre los datos de entrenamiento.

Capítulo 7. Regresión logística.
Generación del modelo de regresión logística.

Capítulo 8. Random forest.
Generación del modelo de random forest.

Capítulo 9. Métricas.
Se explicará qué métricas se emplearán en la evaluación de los modelos. Además, se incluye una pequeña explicación de la función creada para poder evaluar todos los modelos generados.

Capítulo 10. Resultados.
Evaluación de los resultados obtenidos de los diez modelos. Se realiza un estudio individual de los modelos generados de regresión logística y random forest para que finalmente se puedan comparar mutuamente.

Capítulo 11. Conclusiones y trabajo futuro.
Conclusiones del proyecto y posibles vías futuras para mejorar o ampliar los resultados obtenidos.

2. Estado del arte

Las medidas en seguridad vial han sido tradicionalmente enfocadas al trinomio formado por infraestructura – conductor – vehículo promoviendo acciones como la instalación de radares, controles de alcoholemia, mejoras en pavimentos o aplicando nuevas tecnologías en vehículos. Todas estas medidas han sido positivas y han provocado un descenso en el número de accidentes de tráfico y, en consecuencia, de las víctimas que se derivan, pero las cifras todavía siguen siendo preocupantes.

Hasta finales del siglo pasado, todavía se seguía considerando la definición de un accidente de tráfico según J.S. Baker:

> *"… hecho, suceso o acontecimiento inesperado o impremeditado, que contiene un elemento de azar o probabilidad y cuyos resultados son indeseables o infortunados..."*

J.S.Baker. (1970). Manual de Investigación de Accidentes de Tráfico. **[1]**

En ella se afirma que un accidente de tráfico tiene un componente de azar, lo que provoca que sea inevitable al no poder ser previsible o predecible. Afortunadamente, en la actualidad, esta teoría ha sido ampliamente rechazada y ha abierto la puerta a considerar el máximo de factores que pueden intervenir en un accidente. Por ello, los países y ciudades comprometidas con la reducción de la accidentalidad están recogiendo desde hace años una gran cantidad de datos sobre los accidentes sucedidos para su posterior análisis y extracción de conocimiento.

En este capítulo se enunciarán algunos Trabajos realizados a partir de todos estos datos recogidos durante años. Además, se explicarán qué proyectos están siendo aplicados actualmente para combatir la siniestralidad.

2.1. Estudios y trabajos realizados

El considerar múltiples factores en un accidente de tráfico implica un aumento en cuanto al tamaño de los datos (volumen), a un ritmo elevado (velocidad), y una estructura diferente entre ellos (variabilidad). Estos tres conceptos son los que llevan a los profesionales a utilizar tecnologías Big Data. Relacionado este concepto, y con la necesidad de estudiar el gran volumen de datos almacenados para extraer conocimiento, surgen el uso de técnicas de Minería de datos.

En este sentido son múltiples los estudios realizados en los últimos tiempos. Los investigadores se han centrado sobre todo en factores directamente relacionados en el momento del accidente como son: la velocidad de los vehículos, límites de velocidades de las vías, situación meteorológica, sexo y edad del conductor, vehículo, infraestructuras, … todos estos factores sirven para crear modelos que dan respuestas a las causas y, por lo tanto, permiten a los profesionales establecer medidas para evitar los accidentes. De los múltiples estudios realizados, destaco unos pocos realizados en nuestro país durante la última década y que reflejan el estado actual de la implementación de la Minería de datos en el estudio de los accidentes de tráfico.

David Úbeda González **[2]**, en su Tesis Doctoral presenta un proyecto de detección de la gravedad en los accidentes de tráfico. Para ello genera dos modelos clasificadores: Bayes GLM y Random Forest. Un aspecto interesante es el trabajo previo a la hora de detectar los factores con mayor relación con la gravedad del accidente, para así utilizarlos en la creación de los modelos. Los resultados son notables ya que consigue alcanzar una tasa de acierto del 84% en la predicción de accidentes graves o fallecimientos y un 74% en los accidentes leves.

Joaquín Montesinos Muñoz **[3]**, presenta en el año 2018 su trabajo final de Máster donde a partir de diferentes modelos de regresión lineal consigue predecir el número de accidentes que se producirán en las carreteras periféricas de la ciudad de Valencia.

Luis Cruz Bellas **[4]**, también en su trabajo final de Máster, intenta calcular la probabilidad que un accidente de tráfico tenga lugar en un punto concreto de Madrid. Sus resultados son negativos debido a la baja representación de observaciones en las que se produce un accidente y al tipo de variables.

Blanca Arenas **[5]**, responsable de la unidad de estudios de transporte e impacto ambiental de los vehículos en el instituto universitario de investigación del automóvil (*INSIA*) argumenta como los modelos de regresión de Poisson y binominal negativa son los más utilizados y efectivos en la predicción de frecuencias de los accidentes de tráfico ya que permiten utilizar factores no relacionados con la conducción como las características de las carreteras, características del tráfico o condiciones climáticas y ambientales.

En el 2015, Rosario Cintas y Jose Luis Brita-Paja **[6]**, presentan un caso de estudio. Al igual que David Úbeda, realizan un trabajo previo de detección de los factores más relevantes. Se muestran los resultados de diferentes modelos: Regresión Logística con Boosting, Redes Neuronales, Random Forest, Gradient Boosting y modelo Bayesiano.

La revisión de todos estos trabajos hace concluir, que, como se explicará en futuros capítulos, los modelos de regresión logística son los más utilizados debido a su efectividad. También, en el caso de clasificadores, los modelos random forest superan con creces los resultados ofrecidos por otros modelos como los árboles simples o las Redes Neuronales. Es por ello por lo que en este trabajo se intentará extraer conclusiones a partir de estos dos modelos.

2.2. Aplicaciones

Los estudios realizados han servido para extraer conclusiones importantes y han permitido aplicar medidas más específicas para la reducción de la accidentalidad y, en consecuencia, del número de víctimas. A continuación, se expondrán algunos proyectos donde la Minería de datos se está aplicando como pilar fundamental.

2.2.1. Waycare

Waycare (https://waycaretech.com/) **[7]** es un startup israelí destinada a organismos públicos para la gestión del tráfico y la mejora de movilidad. El año 2017 dio un paso más allá e inició un test en la Interestatal número 15 que transcurre por Las Vegas (EEUU) para la predicción de los accidentes de tráfico. Además de datos históricos sobre accidentes, se recogió información en tiempo real de los vehículos y del estado de las carreteras. Algunos factores recogidos fueron: cambios bruscos de dirección en

el vehículo, frenadas, atascos, datos meteorológicos, eventos especiales en los alrededores, construcciones, cierres de carreteras, …

A partir de estos datos se aplicaron algoritmos clasificación de Minería de datos para determinar si se iba a producir un accidente con una previsión de dos horas.

En caso de predecir un posible accidente, se abría el protocolo de actuación descrito en la **Figura 2-1**

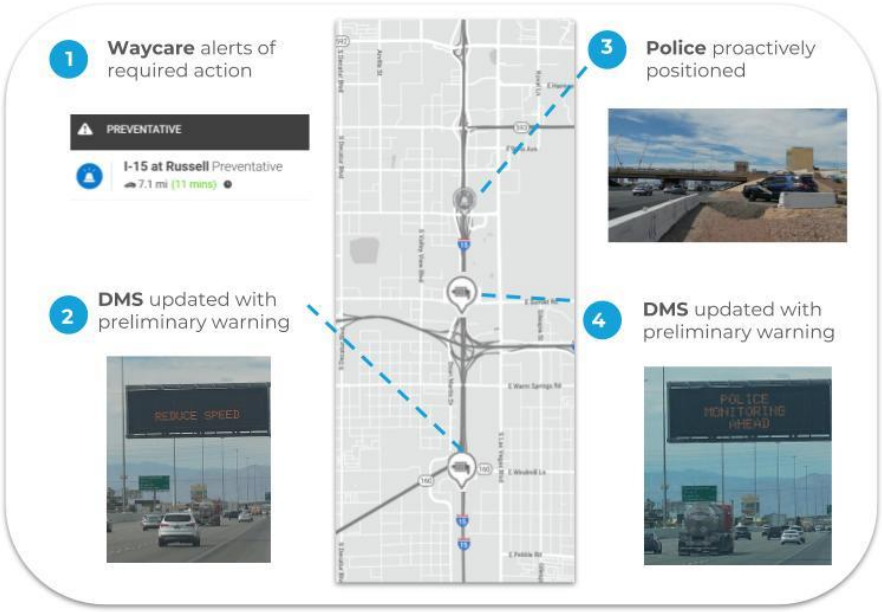

Figura 2-1. Esquema de funcionamiento de Waycare. (fuente: Waycare)

Una vez la plataforma recibe un aviso sobre un posible accidente, envía una alerta a la Regional Transportation Commission RTC. Estos activan los paneles luminosos situados antes del punto predicho con mensajes informativos para los conductores del estilo: reducción de la velocidad, o realizar una conducción más cuidadosa en ese tramo. Para acabar, la policía y los equipos sanitarios se sitúan en el punto donde se prevé que suceda el accidente.

La aplicación de estas medidas redujo en un 17% el número de accidentes de tráfico en tramo de la interestatal 15 que transcurre por las Vegas, reduciendo a su vez el número de víctimas.

2.2.2. Crash Risk Map

Uno de los proyectos pioneros y más conocidos en el ámbito de la predicción de la accidentalidad es el Indiana Crash Risk Map **[8]**; creado el año 2017 y desplegado conjuntamente por la policía del estado de Indiana ISP y el Management Performance Hub MPH. Consta de un mapa interactivo como el que se puede observar en la Figura **2-2**, creado con la tecnología Esri al que se accede desde: https://www.in.gov/isp/ispCrashApp/main.html. Se visualiza el mapa de estado de Indiana dividido en áreas de diferentes colores, los cuales representan el riesgo de accidente. Estos resultados se obtienen a partir de modelos predictivos los cuales se alimentan múltiples datos históricos sobre accidentes. Se trabajan con factores directamente relacionados con el accidente como: velocidad, conductor, vehículo, etc.

pero también otro tipo de información como: posición del sol, estado del tráfico, proximidad de carteles publicitarios, precio del combustible, etc.

A parte de ser una herramienta de consulta para usuarios de las carreteras del estado, también permite extraer información a las autoridades. Por ejemplo, se observó que cuando hay un descenso en el precio del combustible, la población utiliza mucho más el coche y, por lo tanto, hay más accidentes.

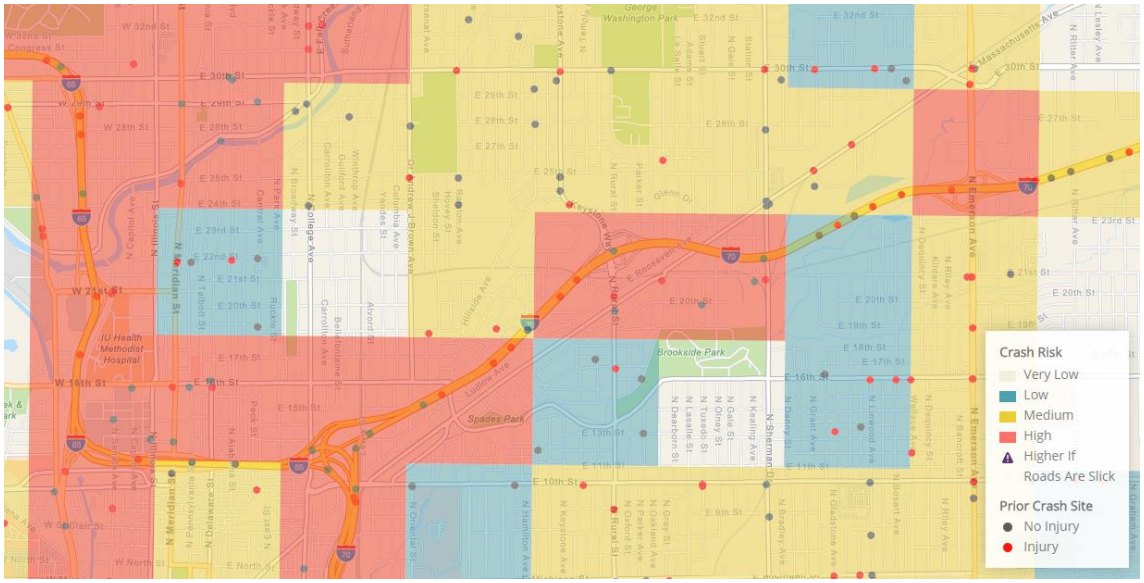

Figura 2-2. Captura de pantalla de la herramienta Indiana Crash Risk Map.

2.2.3. Proyectos en Barcelona

A nivel local, recientemente, se ha publicado una noticia en “el Periodico de Catalunya” **[9]** en el que se entrevista a Manuel Haro, jefe de la unidad de investigación y prevención de accidentes de la Urbana (UIPA) y a Juanjo Vilanova, intendente mayor de la Guardia Urbana y responsable de la división de Tráfico. En este artículo se explica cómo se ha iniciado un proyecto conjunto entre la Guardia Urbana y Accenture con la intención de reducir los accidentes de tráfico en la ciudad de Barcelona. La idea inicial es la de establecer alertas sobre los perfiles, ubicaciones y metodologías de actuación, para que en un futuro se puedan predecir los puntos dónde puede producirse un accidente. Para alcanzar todos estos objetivos se procederá a analizar los datos de los accidentes sucedidos durante los 10 últimos años. Además, se unirá al proyecto la Unitat d’Investigació i Prevenció d’Accidentalitat (UIPA) la cual aportará la experiencia y conocimientos adquiridos desde su formación, sobre el año 1999.

Como se puede observar, el proyecto que justo ahora inicia Barcelona tiene un gran parecido con el ya realizado por el estado de Indiana de EEUU.

2.3. Trabajo actual y estudios ya realizados

El Trabajo actual que justo ahora da inicio se ha nutrido de los diferentes artículos, trabajos y proyectos que se han ido citando en este capítulo. Como se ha podido comprobar, estudiar un accidente de tráfico y sus víctimas es una tarea de enorme complejidad. Solo hace falta comprobar el número y tipos de factores que se tienen en cuenta en los estudios y la gran variedad de algoritmos empleados. En consecuencia, se observa una cierta lentitud en la mejora e implementación de Big Data en el ámbito

de la accidentalidad vial, pero ya hay proyectos en explotación (Waycare o Crahs Risk Map) que están empezando a dar sus resultados.

Sin llegar al nivel de detalle tan grande al que se alcanza en las aplicaciones anteriormente descritas, en este trabajo se intentará aplicar algunas de las recomendaciones leídas en los trabajos mencionados. Por ejemplo, Luis Cruz Bellas **[4]** concluye en su trabajo que en el 94% de las observaciones, la variable dependiente adquiere valor 0 (no hay accidente). Este hecho implica que el modelo no sea correctamente entrenado. En este Trabajo, se intentará dar respuesta a esta problemática llamada clases desbalanceadas.
En cuanto a los factores desencadenantes de los accidentes, se intentarán considerar el mayor número posible que se puedan conseguir de forma abierta (webs Open Data).

3. Extracción de los datos

El punto de partida de cualquier proyecto parecido al actual, parte de la definición de los datos de interés. Esto implica varias fases, siendo la primera la del planteamiento de los datos que son necesarios para conseguir los objetivos fijados. Una vez decididos, empieza la búsqueda de la fuente de estos datos. En cualquier empresa, habitualmente, estos datos se encuentran dentro de la propia organización, y pese a tener que adaptarlos, el proceso de interpretación es mucho más simple. En cambio, en un proyecto de recerca como el actual se debe estudiar diferentes fuentes de datos hasta encontrar aquellas que se adaptan mejor a nuestros requisitos.

En este capítulo, se pretende explicar de forma teórica los datos que van a ser empleados durante todo el proyecto. Se partirá detallando las fuentes de donde se han adquirido y a continuación, se explicará qué información contiene y como puede ayudar a la realización del trabajo. Además, en este primer contacto con los datos, se descartan algunas variables que ya sea por repetición o por poco interés, serán eliminadas del estudio.

3.1. Fuentes de datos

¿Qué tipo de datos reflejan mejor la gravedad de los involucrados en un accidente de tráfico? Como se ha visto en el capítulo **2** se han realizado múltiples trabajos en esta área. En todos ellos se consideran factores directamente relacionados con los implicados como edad, sexo, estado del conductor y otros externos como la meteorología, tipología del accidente, etc. Así pues, se ha pretendido seguir en esta línea y se han extraído de diferentes fuentes de datos informaciones directamente relacionadas con los accidentes sucedidos entre los años 2017 y 2018 en Barcelona (tipologías de accidentes, involucrados, vehículos), así como otros datos que pueden influir en la ocurrencia de estos siniestros como son factores meteorológicos, o un estudio de los días festivos y pre/post festivos de la ciudad.

3.1.1. Accidentes en Barcelona

El elemento generador de las víctimas es el accidente de tráfico; razón por la cual, es vital tener el máximo de información posible sobre el evento para poder afrontar el proyecto. En el portal web Open Data de Barcelona https://opendata-ajuntament.barcelona.cat encontramos diferentes archivos *comma-separated values* (csv) con información sobre los accidentes sucedidos en la ciudad por años. La información se encuentra desagregada de la siguiente forma:

Información sobre las personas involucradas:
https://opendata-ajuntament.barcelona.cat./data/es/dataset/accidents-persones-gu-bcn

Personas involucradas en un accidente gestionado por la Guardia Urbana en la ciudad de Barcelona y que han sufrido algún tipo de lesión (herido leve, herido grave o muerte). Incluye descripción de la persona (conductor, pasajero o peatón), sexo, edad, vehículo asociado a la persona y si la causa ha sido del peatón. Las coordenadas se expresan en el sistema de referencia ED50 y en el sistema geográfico (longitud y latitud).

Información sobre las causas de los accidentes:
https://opendata-ajuntament.barcelona.cat./data/es/dataset/accidents-causes-gu-bcn

Listado de la causalidad de los accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona. Un accidente puede tener una o más causas mediatas las cuales hacen referencia a factores externos del resultado en tiempo, lugar o grado (ej.: Alcoholemia y Exceso de velocidad o inadecuada). Las coordenadas se expresan en el sistema de referencia ED50 y en el sistema geográfico (longitud y latitud).

Información sobre la tipología de los accidentes:
https://opendata-ajuntament.barcelona.cat./data/es/dataset/accidents-tipus-gu-bcn

Listado de los tipos de accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona. Un accidente puede ser tipificado como más de un tipo (ej.: caída (dos ruedas) y abarque). Las coordenadas se expresan en el sistema de referencia ED50 y en el sistema geográfico (longitud y latitud).

Información sobre los vehículos implicados en los accidentes:
https://opendata-ajuntament.barcelona.cat./data/es/dataset/accidents-vehicles-gu-bcn

Listado de los vehículos implicados en accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona. Un accidente puede tener asociado más de un vehículo. Incluye si la causa es del peatón, el tipo de vehículo, modelo, marca, color y tipo carnet y antigüedad de la persona que lo conducía. Las coordenadas se expresan en el sistema de referencia ED50 y en el sistema geográfico (longitud y latitud).

Información sobre los accidentes con valores globales por accidente:
https://opendata-ajuntament.barcelona.cat./data/es/dataset/accidents-gu-bcn

Listado de los accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona. Incorpora el número de lesionados según gravedad, el número de vehículos y el punto de impacto. Las coordenadas se expresan en el sistema de referencia ED50 y en el sistema geográfico (longitud y latitud).

3.1.2. Meteorología

La meteorología puede ser un factor importante en el suceso de los accidentes. Pavimentos mojados, fuertes ráfagas de vientos, etc. aparte de generar accidentes, pueden ser un factor importante en la gravedad de los heridos. La Agencia Estatal de Meteorología (AEMET) ofrece un histórico de los datos meteorológicos diarios. Mediante el uso de una API key, se ha podido descargar un documento *JavaScript Object Notation* (json) con diferentes datos meteorológicos sucedidos en la ciudad de Barcelona entre el 1 de enero de 2017 y el 31 de diciembre de 2018.

Figura 3-1. Captura de los datos meteorológicos de AEMET.

Se ha capturado la información de la estación 0200E (Observatori Fabra) ya que es el punto con menor número de NAs en sus observaciones.

3.1.3. Festividades

Los festivos y pre-festivos acostumbran a ser días complicados en cuanto a la movilidad. Esto puede ser in factor importante en cuanto a los accidentes y la gravedad de los mismos. Así pues, se decide crear de forma manual un documento csv con todos los días festivos, pre-festivos y post-festivos de la ciudad de Barcelona.

3.2. Descripción de los datos

Como hemos visto, se han obtenido un total de siete documentos (incluidos los 5 archivos csv con los datos de accidentes en Barcelona) con parte de la información que se va a emplear para las predicciones. Del total de documentos, no se va a utilizar toda la información, ya que en algunos casos la información es redundante, irrelevante o poco explícita. Es por eso, que en las tablas que se mostrarán a continuación, se marcan de color verde aquellas variables que serán utilizadas durante el proyecto; ya sea para crear las predicciones o bien para obtener información descriptiva (como las variables de geolocalización).

En este apartado se describen las variables que se pueden encontrar en cada uno de estos documentos. Comentar que los nombres de las variables de los archivos obtenidos de la web Open Data de Barcelona pueden diferir entre años; por ejemplo, la variable que indica el número de expediente en el archivo *2017_accidents_causes_gu_bcn_*, es `Número_d.expedient`, mientras que en el documento del 2018 *2018_accidents_causes_gu_bcn_* es `Numero expedient`. En esencia describen lo mismo, y para explicar este apartado se emplearán los nombres de las variables del año 2018.

3.2.1. Información sobre las personas

2017_accidents_persones_gu_bcn_.csv
2018_accidents_persones_gu_bcn_.csv

ID	Tipo	Descripción
Numero_expedient	factor	ID del accidente
Codi_districte	int	ID del distrito
Nom_districte	factor	Nombre del distrito
Codi_barri	int	ID del barrio
Nom_barri	factor	Nombre del barrio
Codi_carrer	int	ID de la calle
Nom_carrer	factor	Nombre de la calle
Num_postal	factor	Número postal
Descripcio_dia_setmana	factor	Nombre del día de la semana
Dia_setmana	factor	Abreviatura del día de la semana
Descripcio_tipus_dia	factor	Definición si el día es laboral o no
Any	int	Año
Mes_.any	int	Número del mes
Nom_mes	factor	Nombre del mes
Dia_mes	int	Número del día del mes
Descripcio_torn	factor	Tipo de turno: mañana, tarde o noche
Hora_dia	int	Hora sin minutos ni segundos
Descripcio_causa_vianant	factor	Causa del peatón en caso que sea culpable
Desc_Tipus_vehicle_implicat	factor	Tipo de vehículo donde iba el afectado

Descripcio_sexe	factor	Sexo del afectado
Edat	factor	Edad del afectado
Descripcio_tipus_persona	factor	Tipo de afectado: conductor, pasajero o peatón
Descripcio_situacio	factor	Descripción si el afectado ha sido citado o presentado
Descripcio_victimitzacio	factor	Tipo de herido
Coordenada_UTM_X	num	Coordenada X del accidente
Cooddenada_UTM_Y	num	Coordenada Y del accidente
Longitud	num	Longitud del accidente
Latitud	num	Latitud del accidente

Tabla 3-1. Variables del archivo csv sobre las víctimas de los accidentes.

3.2.2. Información sobre las causas

*2017_accidents_causes_gu_bcn_*
*2018_accidents_causes_gu_bcn_*

ID	Tipo	Descripción
Numero_expedient	factor	ID del accidente
Codi_districte	int	ID del distrito
Nom_districte	factor	Nombre del distrito
Codi_barri	int	ID del barrio
Nom_barri	factor	Nombre del barrio
Codi_carrer	int	ID de la calle
Nom_carrer	factor	Nombre de la calle
Num_postal	factor	Número postal
Descripcio_dia_setmana	factor	Nombre del día de la semana
Dia_setmana	factor	Abreviatura del día de la semana
Descripcio_tipus_dia	factor	Definición si el día es laboral o no
Any	int	Año
Mes_any	int	Número del mes
Nom_mes	factor	Nombre del mes
Dia_mes	int	Número del día del mes
Hora_dia	int	Hora sin minutos ni segundos
Descripcio_torn	factor	Tipo de turno: mañana, tarde o noche
Descripcio_causa_mediata	factor	Descripción de la causa del accidente
Coordenada_UTM_X	num	Coordenada X del accidente
Cooddenada_UTM_Y	num	Coordenada Y del accidente
Longitud	num	Longitud del accidente
Latitud	num	Latitud del accidente

Tabla 3-2. Variables del archivo csv sobre las causas de los accidentes.

3.2.3. Información sobre las tipologías

2017_accidents_tipus_gu_bcn_.csv
2018_accidents_tipus_gu_bcn_.csv

ID	Tipo	Descripción
Codi_expedient	factor	ID del accidente
Codi_districte	int	ID del distrito
Nom_districte	factor	Nombre del distrito
Codi_barri	int	ID del barrio
Nom_barri	factor	Nombre del barrio

Codi_carrer	int	ID de la calle
Nom_carrer	factor	Nombre de la calle
Num_postal.	factor	Número postal
Dia_setmana	factor	Nombre del día de la semana
ID_Dia_setmana	factor	Abreviatura del día de la semana
Tipus_dia	factor	Definición si el día es laboral o no
Any	int	Año
Mes	int	Número del mes
Nom_mes	factor	Nombre del mes
Dia_de_mes	int	Número del día del mes
Hora_de_dia	int	Hora sin minutos ni segundos
Torn	factor	Tipo de turno: mañana, tarde o noche
Tipus_accident	factor	Descripción de la tipología del accidente
Desc_Tipus_vehicle_implicat	factor	Tipo de vehículo donde iba el afectado
Coordenada_UTM_X	num	Coordenada X del accidente
Cooddenada_UTM_Y	num	Coordenada Y del accidente
Longitud	num	Longitud del accidente
Latitud	num	Latitud del accidente

Tabla 3-3. Variables del archivo csv sobre las tipologías de los accidentes.

3.2.4. Información individual de los accidentes

2017_accidents_gu_bcn.csv
2018_accidents_gu_bcn.csv

ID	**Tipo**	**Descripción**
Numero_expedient	factor	ID del accidente
Codi_districte	int	ID del distrito
Nom_districte	factor	Nombre del distrito
Codi_barri	int	ID del barrio
Nom_barri	factor	Nombre del barrio
Codi_carrer	int	ID de la calle
Nom_carrer	factor	Nombre de la calle
Num_postal	factor	Número postal
Descripcio_dia_setmana	factor	Nombre del día de la semana
Dia_setmana	factor	Abreviatura del día de la semana
Descripcio_tipus_dia	factor	Definición si el día es laboral o no
Any	int	Año
Mes_any	int	Número del mes
Nom_mes	factor	Nombre del mes
Dia_mes	int	Número del día del mes
Hora_dia	int	Hora sin minutos ni segundos
Descripcio_torn	factor	Tipo de turno: mañana, tarde o noche
Descripcio_causa_vianant	factor	Descripción de la causa del peatón en caso que así sea
Numero_morts	int	Número de muertos derivados
Numero_lesionats_lleus	int	Número de heridos leves derivados
Numero lesionats_greus	int	Número de lesionados graves derivados
Numero_victimes	int	Número de víctimas totales
Numero_vehicles_implicats	int	Número de vehículos implicados
Coordenada_UTM_X	num	Coordenada X del accidente
Cooddenada_UTM_Y	num	Coordenada Y del accidente

Longitud	num	Longitud del accidente
Latitud	num	Latitud del accidente

Tabla 3-4. Variables del archivo csv sobre los datos generales de cada accidente.

3.2.5. Información sobre los vehículos

*2017_accidents_vehicles_gu_bcn_*
*2018_accidents_vehicles_gu_bcn_*

ID	Tipo	Descripción
Codi_expedient	factor	ID del accidente
Codi_districte	int	ID del distrito
Nom_districte	factor	Nombre del distrito
Codi_barri	int	ID del barrio
Nom_barri	factor	Nombre del barrio
Codi_carrer	int	ID de la calle
Nom_carrer	factor	Nombre de la calle
Num_postal	factor	Número postal
Descripcio_dia_setmana	factor	Nombre del día de la semana
Dia_setmana	factor	Abreviatura del día de la semana
Descripcio_tipus_dia	factor	Definición si el día es laboral o no
Any	int	Año
Mes_any	int	Número del mes
Nom_mes	factor	Nombre del mes
Dia_mes	int	Número del día del mes
Hora_dia	int	Hora sin minutos ni segundos
Descripcio_torn	factor	Tipo de turno: mañana, tarde o noche
Descripcio_causa_vianant	factor	Descripción de la causa del peatón en caso que así sea
Desc_tipus_vehicle	factor	Tipo de vehículo
Descripcio_model	factor	Modelo del vehículo
Descripcio_marca	factor	Marca del vehículo
Descripcio_color	factor	Color del vehículo
Descripcio_carnet	factor	Tipo del carnet dependiendo del vehículo conducido
Antiguitat_carnet	factor	Años de antigüedad del carné de conducir
Coordenada_UTM_X	num	Coordenada X del accidente
Cooddenada_UTM_Y	num	Coordenada Y del accidente
Longitud	num	Longitud del accidente
Latitud	num	Latitud del accidente

Tabla 3-5. Variables del archivo csv sobre vehículos involucrados en los accidentes.

Como resumen de los datos extraídos de la web Open Data de Barcelona, podemos observar como en todos los documentos tienen variables repetidas; y realmente, hay una variable que tiene relación con lo que describe el data frame. Por ejemplo, del data frame de las tipologías únicamente interesa la variable que indica la tipología, en el de las causas, solo extraemos la variable que indica la causa del accidente, etc. A parte de estas variables, en cada conjunto de datos también se extrae el ID para poder relacionar los data frames. De esta manera, el grosor de la información se extrae el primer data frame con información sobre la víctimas, y del resto, se extraen el identificador y una variable.

Otro detalle importante es que se obvia el data frame sobre los vehículos. La variable sobre el tipo de vehículo ya está descrita en el primer documento sobre las personas involucradas, pero además se hubiera querido añadir información sobre el modelo. Lamentablemente, a medida que se iba avanzando con el proyecto, ha sido imposible poder relacionar este tipo de información. Por ejemplo, en un accidente donde hay involucrados dos turismos, es imposible saber qué víctima conducía un modelo u otro. Esta problemática se ha explicado con más detalle en el siguiente capítulo, cuando se ha procedido a unificar toda la información.

3.2.6. Información sobre la meteorología

meteo_2017_2018_bcn.json

ID	Tipo	Descripción
fecha	chr	Fecha AA-MM-DD
indicativo	chr	Indicativo de la medida
nombre	chr	Población de la estación
provincia	chr	Provincia de la estación
altitud	chr	Altitud de la estación en metros
tmed	chr	Temperatura media diaria en ºC
prec	chr	Precipiación media diaria en mm
tmin	chr	Temperatura mínima del día en ºC
horatmin	chr	Hora y minuto de la temperatura mínima
tmax	chr	Temperatura máxima del día en ºC
horatmax	chr	Hora y minuto de la temperatura máxima
dir	chr	Dirección de la racha máxima en decenas de grado
velmedia	chr	Velocidad media del viento en m/s
racha	chr	Racha máxima del viento en m/s
horaracha	chr	Hora y minuto de la racha máxima

Tabla 3-6. Variables del archivo json sobre información meteorológica.

Se ha focalizado el estudio en las precipitaciones, la temperatura y la velocidad del viento. Como se puede observar, los datos son medias diarias y por lo tanto estamos perdiendo cierta precisión en nuestro estudio.

3.2.7. Información sobre los pre/post y festivos

calendario_festivos_17_18.csv

ID	Tipo	Descripción
fecha	factor	Fecha AAAA-MM-DD
festividad	factor	Festivo, Post_festivo o Pre_festivo

Tabla 3-7. Variables del archivo csv sobre los días festivos.

Este data frame únicamente incluye el campo fecha para poder relacionarlo con los demás, y un valor asociado a cada fecha indicando si es festivo, post festivo o pre festivo.

4. Procesado de los datos

Este es el primer capítulo donde se realizará la primera toma de contacto con los datos y el entorno de trabajo en RStudio. Una vez se han definido las variables de interés de cada una de la fuente de datos, se procede a su carga en R.

El primer problema que debe ser solucionado es la condición de desegregación que tienen los datos de entrada. Una vez se unifiquen todos en un único data frame, se realizarán las tareas estrictamente de procesado (limpieza, reducción, tratamiento NAs, etc) para obtener un conjunto de datos final apto para generar los modelos.

En este capítulo se explicarán partes muy concretas del código creado. Si se desea comprobar el código entero, se pueden encontrar en el ***Anexo: Códigos*** *en las secciones* ***A.Código 1: Unificación de datos.*** *y* ***B.Código 2: Procesado de los datos.***

4.1. Unificación de los datos

Los datos utilizados para este trabajo se encuentran desagregados debido a que la información sobre accidente en la web OpenData la información se encuentra en diferentes archivos y porque se incorporan datos externos como información meteorológica o sobre festivos.

Para poder aplicar los modelos predictivos es necesario obtener un dataframe unificado con toda esta información. Es por ello por lo que el primer paso a realizar sea la unificación de todos los archivos.

4.1.1. Relaciones entre data frames

Para unificar los datos se debe determinar qué variables son las que permiten asociar las diferentes observaciones de cada archivo. En el caso que nos ocupa los datos del data frame *persona* (descripción de la víctima), *tipo* (tipología del accidente) y *causa* (causa del accidente) se relacionan mediante la variable `Numero expedient` o `Codi expedient` las cuales contienen un identificador único sobre el accidente. Utilizando estas variables se consigue un único data frame llamado *data* que unifica los tres documentos: *persona*, *tipo* y *causa*. Para facilitar el proceso, se renombran las dos variables con el nombre `id`.

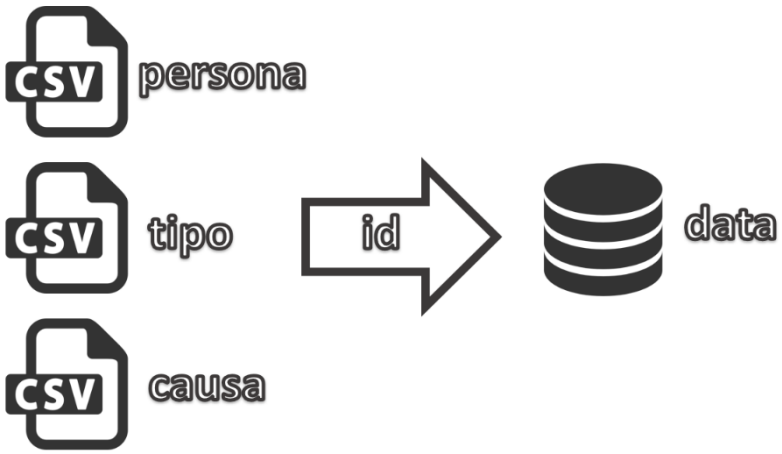

Figura 4-1. Unificación de persona, tipo y causa mediante la variable id.

El archivo json, el cual contiene la información meteorológica, al provenir de una fuente de datos diferente a la anterior, no tiene ninguna variable que permita relacionar directamente los datos. Dado que la información meteorológica es a nivel diario, se ha optado que la fecha sea el identificador único que permita relacionar los datos. Así pues, se ha generado una nueva variable fecha en el conjunto *data* (conjunto obtenido a partir de *persona*, *tipo* y *causa*) a partir de las variables `dia`, `mes` y `año`.

Los datos sobre los días festivos han sido generados manualmente, por lo que se ha aprovechado a crear el identificador único con el mismo formato que la variable `fecha` del archivo json. Finalmente, se obtiene los datos de: *persona*, *tipo*, *causa*, *meteo* (meteorología) y *cal* (festividades) unificados en el data frame*data*.

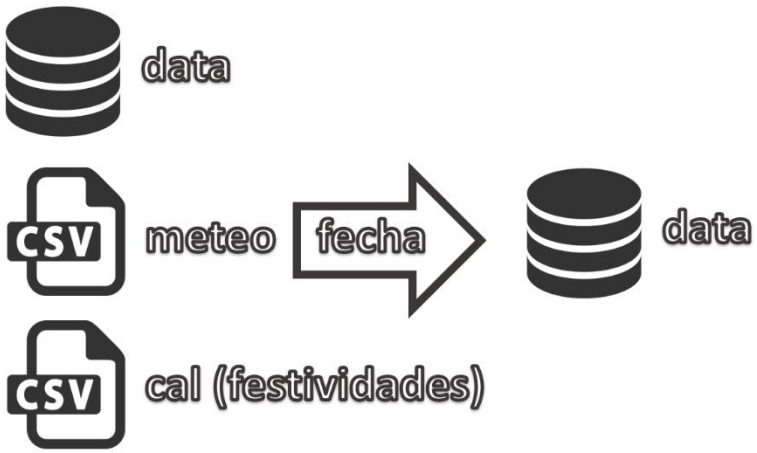

Figura 4-2. Unificación de data, meteo y cal mediante la variable fecha.

Estos dos procesos explicados teóricamente e ilustrados en la **Figura 4-1** y Figura **4-2**, adquieren una complicación extra debido a la falta de información de datos a la hora de unificar. Se explica a continuación.

4.1.2. Proceso de unificación

Recordemos que la finalidad de este trabajo es la de predecir la gravedad de cada uno de los implicados en un accidente de tráfico. Así pues, el data frame *personas* servirá de referencia para construir el data frame definitivo *data*. A medida que se empieza a diseñar el proceso de unión de datos es cuando nos

damos cuenta de la limitación que existe al coger como referencia la variable `id`. Un accidente puede tener más de una víctima (ya sea leve o grave) y, por lo tanto, cada uno de los registros contendrá la misma `id`. Gracias a variables como `sexo` o `edad`, se pueden diferenciar las víctimas dentro del mismo data frame *personas*; pero al no existir estas variables en los data frames *tipo* y *causa*, hace imposible, en muchos casos, asociar las observaciones para cada herido. Explicaremos el problema y las soluciones a partir de los siguientes ejemplos reales.

Ejemplo 1: Varias tipologías de accidentes para varios heridos.

En el data frame *persona* con `id = 2017S000147` encontramos el siguiente resultado (se obvian variables irrelevantes para mostrar el ejemplo):

```
##              id    vehiculo sexo edad tipo_persona   gravedad
## 174 2017S000147 Motocicleta Home   31    Conductor Ferit greu
## 175 2017S000147 Motocicleta Home   45    Conductor Ferit greu
## 176 2017S000147     Turismo Dona   33    Conductor Ferit lleu
```

En el data frame *tipo* con `id = 2017S000147` encontramos el siguiente resultado:

```
##                id                        tipo
## 3683  2017S000147       Caiguda (dues rodes)
## 5071  2017S000147 Xoc contra element estàtic
## 10556 2017S000147   Col.lisió fronto-lateral
```

Con los datos mostrados es imposible juntar esta información sin asumir cierto error ya que no podemos asegurar qué tipo de accidente corresponde a cada registro de persona. La lógica nos hace deducir que la tipología Caiguda (dues rodes) está relacionada con una de las dos motocicletas; pero tampoco podemos asegurar que corresponda a la primera motocicleta o a la segunda.

Ejemplo 2: Una tipología de accidente para varios heridos.

En el data frame *persona* con `id = 2017S000987` encontramos el siguiente resultado (se obvian variables irrelevantes para mostrar el ejemplo):

```
##              id vehiculo sexo edad tipo_persona   gravedad
## 1155 2017S000987  Turismo Dona   32    Conductor Ferit lleu
## 1156 2017S000987  Turismo Dona   38    Conductor Ferit lleu
```

En el data frame *tipo* con `id = 2017S000987` encontramos el siguiente resultado:

```
##              id  tipo
## 5508 2017S000987 Abast
```

Con los datos mostrados no podemos saber qué vehículo ha sufrido la tipología descrita.

Ejemplo 3: Un herido para varias tipologías.

En el data frame *persona* con `id = 2017S000026` encontramos el siguiente resultado (se obvian variables irrelevantes para mostrar el ejemplo):

```
##             id vehiculo sexo edad tipo_persona   gravedad
## 29 2017S000026  Turismo Dona   70    Conductor Ferit lleu
```

En el data frame *tipo* con `id = 2017S000026` encontramos el siguiente resultado:

```
##                id                          tipo
## 3553  2017S000026           Col.lisió lateral
## 9044  2017S000026 Bolcada (més de dues rodes)
## 10406 2017S000026  Xoc contra element estàtic
```

En este caso tenemos tres tipologías para un único herido. Tenemos un claro ejemplo de sobre-información ya que de un accidente donde se ha producido una colisión lateral, un vuelco y un choque contra un elemento estático, se debe reducir a una tipología.

Se puede observar que, tratando los tres problemas descritos en los ejemplos anteriores, se introduce cierto error que pude condicionar los resultados. Debido al grado de error que se considera solucionar los diferentes problemas, se ha decidido trabajar únicamente el siguiente tipo de accidentes:

- Sólo existe una tipología en el accidente.
- Existen varias tipologías, pero un único herido.
- Existen varias tipologías, pero un único conductor y un único vehículo implicado. Esta condición difiere de la anterior ya que, en este caso, puede haber varios heridos (conductor, pasajero o peatón).

Los accidentes que no cumplan una de las tres condiciones, no serán evaluados en este Trabajo para evitar el error que comporta su tratamiento. De hecho, el primer ejemplo, no cumple ninguna condición y por lo tanto no ha sido considerado en este Trabajo.

El segundo ejemplo se resuelve fácilmente asignando la misma tipología a todos los heridos. Así pues, se asigna la tipología `Abast` a los dos turismos.

El tercer ejemplo se ha resuelto reduciendo el número de tipologías a 1. Para aplicar la reducción del estilo $X \rightarrow 1$ se genera una lista con las 17 posibles tipologías ordenadas de mayor impacto a menor impacto. Para cada accidente se extraen las diferentes tipologías y se comparan con las de la lista. Se escogerá, aquella que ocupe la posición más alta de la lista (mayor impacto). La lista utilizada es la siguiente:

```
tipologias <- c("Abast", "Encalç", "Abast multiple", "Col.lisió latera
l", "Col.lisió fronto-lateral", "Xoc contra element estàtic", "Col.lis
ió frontal", "Sortida de via amb bolcada", "Sortida de via amb xoc o c
ol.lisió", "Xoc amb animal a la calçada", "Resta sortides de via", "Bo
lcada (més de dues rodes)", "Atropellament", "Caiguda (dues rodes)", "
Caiguda interior vehicle", "Altres", "Desconegut")
```

Con esta solución estamos asumiendo dos errores. El primero corresponde a la reducción, ya que estamos quitando de forma inevitable información sobre el accidente. El segundo corresponde a la interpretación que se ha hecho al generar el orden de la

lista anterior. Para ello se ha analizado la cantidad de heridos graves que se asocian a cada tipología; y de ahí se ha hecho una lista agrupando por familia. Por ejemplo, `Abast`, `Encalç`, i `Abast multiple`, al pertenecer a accidentes donde han sucedido Alcances (choque por detrás) se ponen juntas. De esta manera, en el ejemplo 3, de las tres tipologías, se ha asignado `Col·lisió lateral` a todos los heridos.

El fragmento de código utilizado para solucionar estos problemas es el siguiente:

```
for (x in 1: length(ids)){

  subp <- data[as.character(data$id) == ids[x],]
  subt <- tipo[as.character(tipo$id) == ids[x],]
  subg <- general[as.character(general$id) == ids[x],]

  heridos <- nrow(subp)
  tipos <- nrow(subt)
  vehiculos <- subg$n_vehic

  rol<-table(subp$tipo_persona)
  # Sólo una tipología -> todos comparten tipología
  if(tipos == 1){
    data[data$id == ids[x],]$tipo <- as.character(subt$tipo)
  } else{
  # Sólo un herido o sólo un conductor, se elige la tipología más rest
rictiva
    if(heridos == 1 || (rol[1] == 1 && vehiculos == 1)){
      i <- na.omit(match(subt$tipo, tipologias))
      tipo_reducida <- subt$tipo[which(i == min(i))]
      data[data$id == ids[x],]$tipo <- as.character(tipo_reducida)
    }
  }
}
# Si la víctima es un peatón, únicamente ha podido ser atropellado
data[data$tipo_persona == "Vianant",]$tipo <- "Atropellament"
```

Para acabar con el proceso de unificación, comentar que el data frame *causa* tiene una problemática parecida a la del data frame *tipo* ya que hay observaciones que comparten `id` y por lo tanto se debe hacer una reducción. En este caso, el proceso es mucho más simple ya que únicamente hay 8 casos en todo el data frame donde sucede este problema y siempre se repiten las causas: `Alcoholèmia` y `Excès de velocidad o inadequada`. Para solucionar este caso y reducir 2 →1, se ha aplicado el siguiente razonamiento: El conductor iba a una velocidad inadecuada debido a la ingesta excesiva de alcohol. Es por eso por lo que la causa principal elegida ha sido `Alcoholèmia`.

Las demás uniones son simples y se realizan mediante la variable `fecha`. El data frame *data* obtenido está compuesto por las siguientes variables:

```
## 'data.frame':    23860 obs. of  22 variables:
##  $ fecha       : Factor w/ 730 levels "2017-01-01","2017-01-02",..: 1 1 1 1
1 1 1 1 1 1 ...
##  $ id          : Factor w/ 18507 levels "2017S000001",..: 10 12 11 3 2 4 14
10 9 8 ...
##  $ distrito    : Factor w/ 4326 levels "","A Zona Franca",..: 1278 3297 135
8 1003 2559 2991 81 1278 3552 1727 ...
##  $ dia_nombre  : Factor w/ 7 levels "Dc","Dg","Dj",..: 2 2 2 2 2 2 2 2 2 2
...
##  $ año         : int  2017 2017 2017 2017 2017 2017 2017 2017 2017 2017 ...
##  $ mes         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dia         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hora        : int  10 16 15 7 2 7 16 10 15 12 ...
##  $ causa_peaton: Factor w/ 6 levels "Altres","Creuar per fora pas de vianan
ts",..: 5 5 4 5 5 5 5 5 5 5 ...
##  $ vehiculo    : Factor w/ 32 levels "Altres vehicles amb motor",..: 13 20
32 20 20 13 3 13 8 3 ...
##  $ sexo        : Factor w/ 3 levels "Desconegut","Dona",..: 2 3 2 2 3 2 2 3
3 2 ...
##  $ edad        : Factor w/ 101 levels "0","1","10","11",..: 18 48 55 22 44
37 77 21 24 93 ...
##  $ tipo_persona: Factor w/ 3 levels "Conductor","Passatger",..: 2 1 3 1 1 2
2 1 1 2 ...
##  $ gravedad    : Factor w/ 8 levels "Ferit greu","Ferit greu: hospitalitzac
ió superior a 24h",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Lat         : num  41.4 41.4 41.4 41.4 41.4 ...
##  $ Long        : num  2.18 2.14 2.16 2.18 2.19 ...
##  $ tipo        : Factor w/ 17 levels "0","Abast","Abast multiple",..: 11 11
5 11 10 1 8 11 11 8 ...
##  $ causa       : Factor w/ 8 levels "Alcoholèmia",..: 7 7 7 7 7 1 7 7 7 7 .
..
##  $ festividad  : Factor w/ 3 levels "Festivo","Post_festivo",..: 1 1 1 1 1
1 1 1 1 1 ...
##  $ tmed        : Factor w/ 220 levels "-0,4","10,0",..: 199 199 199 199 199
199 199 199 199 199 ...
##  $ prec        : Factor w/ 99 levels "0,0","0,1","0,2",..: 1 1 1 1 1 1 1 1
1 1 ...
##  $ velmedia    : Factor w/ 34 levels "0,6","0,8","1,1",..: 23 23 23 23 23 2
3 23 23 23 23 ...
```

4.2. Procesado de los datos

Una vez obtenidos los datos unificados, se procede a su procesado para que sean aptos y puedan ser utilizados en la creación de los modelos. La información anterior todavía no está lista para poder trabajar con ella. Por ejemplo, la variable `festividad` no está completa ya que la mayoría de sus valores son `NA`; o las variables relacionadas con la meteorología han sido interpretadas como factores en vez de números. En este apartado se solucionarán estos y otros problemas que se irán comentando detalladamente.

4.2.1. Corrección de variables

El software define automáticamente la tipología de cada una de las variables que conforman el data frame. En la mayoría de los casos la asignación es correcta, pero en otras situaciones, o bien es incorrecta, o bien interesa modificar el tipo de variable para mejorar el modelo.

Según el resumen obtenido de la unificación de los datos, las variables meteorológicas (`tmed`, `velmedia` y `prec`) han sido definidas como factores. Evidentemente, no tiene ningún sentido ya que contienen valores de magnitudes como grados centígrados, m/s o mm. Es por eso que se convierten las tres en variables numéricas.
En el caso de las precipitaciones, existe un valor `Ip` el cual significa que ha llovido menos de 0.1mm. En nuestro caso se considera que no ha llovido; es decir, 0mm.

El resumen también muestra como la variable `edad` ha sido definida como factor. En este caso se realiza la conversión a *integer* o entero.

En el apartado de unificación de los datos se ha tenido que hacer una corrección de variables previa para poder generar el data frame *data*. Las variables de geolocalización `Long` y `Lat` del año 2017 estaban definidas como factores mientras que las del año 2018 estaban correctamente definidas como numéricas. Esto imposibilitaba que se unieran los dos años 2017 y 2018; por lo tanto, se ha realizado un proceso previo de procesado de datos convirtiendo los datos del año 2017 en numéricos. La conversión no ha sido correcta debido a la interpretación de los decimales por comas o por puntos. Por eso, en este apartado de corrección de variables se ha procedido a reajustar los valores erróneos dividiéndolos entre 1000.

4.2.2. Reducción de niveles en variables categóricas

El procesado de variables categóricas cuando queremos aplicar modelos predictivos, se convierte en un elemento esencial para lograr un modelo lo más ajustado posible. Este tipo de variables pueden adquirir un número limitado de valores, llamados niveles. Supongamos que una variable está formada por 15 niveles; al crear el modelo, se estarán añadiendo 15 grados adicionales de libertad, lo cual complica enormemente el análisis y puede conducir a un sobreajuste. Es por eso que uno de los procesos clave en este apartado es la reducción de los niveles de algunas variables.

En total, y teniendo en cuenta el procesado realizado anteriormente, el data frame *data* tiene 8 variables definidas como categóricas, incluida la variable dependiente: `distrito`, `dia_nombre`, `vehiculo`, `tipo_persona`, `gravedad`, `tipo`, `causa`, `festividad`. Se han descartado las variables `id` y `fecha` ya que su única función ha sido unir los data frames.

Los niveles de `distrito`, `dia_nombre`, `sexo`, `tipo_persona` y `festividad` no pueden ser reducidos ya que describen todas las posibilidades y no permite agrupación. En cambio, las tres variables restantes sí que pueden ser tratadas.

La variable vehiculo contiene un total de 32 niveles.

```
##  [1] "Altres vehicles amb motor"    "Altres vehicles sense motor"
##  [3] "Autobús"                      "Autobús articulado"
##  [5] "Autobús articulat"            "Autocar"
```

```
##  [7] "Autocaravana"                    "Bicicleta"
##  [9] "Camió rígid <= 3,5 tones"        "Camió rígid > 3,5 tones"
## [11] "Camión <= 3,5 Tm"                "Camión > 3,5 Tm"
## [13] "Ciclomotor"                      "Cuadriciclo <75cc"
## [15] "Desconegut"                      "Furgoneta"
## [17] "Maquinària d'obres i serveis"    "Microbus <= 17"
## [19] "Microbús <= 17"                  "Motocicleta"
## [21] "Otros vehíc. a motor"            "Quadricicle < 75 cc"
## [23] "Quadricicle > 75 cc"             "Taxi"
## [25] "Todo terreno"                    "Tot terreny"
## [27] "Tractocamión"                    "Tractor camió"
## [29] "Tranvía o tren"                  "Tren o tramvia"
## [31] "Turisme"                         "Turismo"
```

Por defecto, existe una falta de unificación de niveles ya que dependiendo de la observación encontramos un mismo vehículo descrito en catalán y en castellano. La primera solución a este problema pasa por unificar los idiomas, pero el número de niveles obtenido sigue siendo muy elevado: 21. Finalmente se ha decidido clasificar los vehículos por tipologías. En la web de la ITEUVE **[10]** encontramos una tabla con los vehículos clasificados en 4 tipologías (en la tabla ya se resuelve el problema de niveles duplicados por el idioma):

Tipo 1	Bicicleta, Ciclomotor, Cuadriciclo < 75cc, Cuadriciclo > 75 cc, Motocicleta
Tipo 2	Taxi, Todo terreno, Turismo
Tipo 3	Camión <= 3.5T, Furgoneta, Microbús <= 17, Autocaravana
Tipo 4	Autobús articulado, Autobús, Autocar, Maquinaria de obras y servicios, Tractocamión, Camión > 3.5T.

Tabla 4-1. Reducción de vehículos a tipos de vehículos (de 32 a 4 niveles).

Hay vehículos como la Autocaravana o Todo terreno que no aparecen en la lista de la ITEUVE. En estos casos se le ha asignado una tipología según tamaños y pesos. Por ejemplo, una Autocaravana se asemeja más en cuanto a peso y tamaño a una furgoneta que a un turismo. Por otro lado, se ha decidido descartar del estudio los Tranvías o trenes, otros vehículos tanto con motor como sin motor y aquellas observaciones donde el vehículo es desconocido.

Seguimos con la variable *tipo* la cual está compuesta por 17 niveles.

```
##  [1] "0"                                 "Abast"
##  [3] "Abast multiple"                    "Altres"
##  [5] "Atropellament"                     "Bolcada (més de dues rodes)"
##  [7] "Caiguda (dues rodes)"              "Caiguda interior vehicle"
##  [9] "Col.lisió frontal"                 "Col.lisió fronto-lateral"
## [11] "Col.lisió lateral"                 "Desconegut"
## [13] "Encalç"                            "Sortida de via amb bolcada"
## [15] "Sortida de via amb xoc o col.lisió" "Xoc amb animal a la calçada"
## [17] "Xoc contra element estàtic"
```

De nuevo nos encontramos con un problema de exceso de niveles. En este caso se decide agrupar las tipologías en 8 grandes grupos:

Alcance	Abast, Abast múltiple, Encalç
Salida	Sortida de via amb volcada, Sortida de via amb xoc o col·lisió
Choque	Xoc amb animal a la calçada, Xoc contra element estàtic
Atropello	Atropellament
Vuelco	Bolcada (més de dues rodes)
Caida	Caiguda (dues rodes)
Caida_interior	Caiguda interior vehicle
Colision	Col·lisió frontal, Col·lisió fronto-lateral, Col·lisió lateral

Tabla 4-2. Reducción de tipologías de accidentes (de 17 a 8 niveles).

La variable *causa* está compuesta por 8 niveles.

```
## [1] "Alcoholèmia"                      "Calçada en mal estat"
## [3] "Drogues o medicaments"            "Estat de la senyalització"
## [5] "Excés de velocitat o inadequada" "Factors meteorològics"
## [7] "No hi ha causa mediata"           "Objectes o animals a la calçada"
```

Con tal de reducir los niveles, se ha optado por realizar las siguientes agrupaciones.

Drogas	Alcoholemia, Drogas o medicamentos
Mal estado elementos viarios	Estado de la señalización, Calzada en mal estado
Velocidad	Exceso de velocidad oinadecuada
Meteorologia	Factores metereologicos
Obstaculos	Objetos o animales en la calzada
Sin causa	Sin causa

Tabla 4-3. Reducción de las causas de los accidentes (de 8 a 6 niveles).

Las variables categóricas restantes: `causa_peaton`, `sexo`, `gravedad` y han sido binarizadas; es decir, se han reducido de tal forma que únicamente puedan adquirir dos valores.

La variable `causa_peaton`. Esta estaba formada por 6 niveles que describían qué acción había realizado el peatón para causar el accidente. Para nuestro estudio nos interesa saber únicamente si el peatón ha sido culpable o no, por lo que se ha asignado `No` a todas las observaciones donde aparceia el nivel `No és causa del vianant` y `Si` en el resto.

El caso de la variable `sexo` es parecido tiene tres niveles: `Hombre`, `Mujer` y `Desconocido`. Se procede a asignar `NA` cuando la variable tiene valor `Desconocido`. De esta manera, únicamente existen dos valores conocidos: `Hombre` y `Mujer`.

Finalmente, también se ha modificado la variable dependiente de este Trabajo `gravedad`, la qual tiene 8 niveles:

```
##                                                  Ferit greu
##                                                          46
##                 Ferit greu: hospitalització superior a 24h
##                                                         428
##                                                Ferit lleu
##                                                        2961
## Ferit lleu: Amb assistència sanitària en lloc d'accident
##                                                        6250
##                     Ferit lleu: Hospitalització fins a 24h
##                                                       13321
##                  Ferit lleu: Rebutja assistència sanitària
##                                                         821
##                                                       Mort
##                                                           3
##                       Mort (dins 24h posteriors accident)
##                                                          30
```

Para la variable objetivo se ha decidido clasificarla en dos opciones: `Grave` o `Leve`. La tabla siguiente muestra cómo se han clasificado los diferentes niveles.

Grave	Herido grave, herido grave: hospitalización superior a 24h, Muerto, Muerto (dentro 24h posteriores accidente).
Leve	Herido leve, Herido leve: Hospitalización hasta 24 h, Herido leve: Con asistencia sanitaria en el lugar del accidente, Herido leve: Rechaza asistencia sanitaria.

Tabla 4-4. Binarización de la variable dependiente gravedad (de 8 niveles a 2).

4.2.3. Creación de nuevas variables

Pese a no formar parte de este apartado, se ha hecho un análisis rápido de todas las variables numéricas para detectar el número de valores atípicos. La variable más conflictiva ha sido `prec`, referente a los mm de lluvia caídos por día.

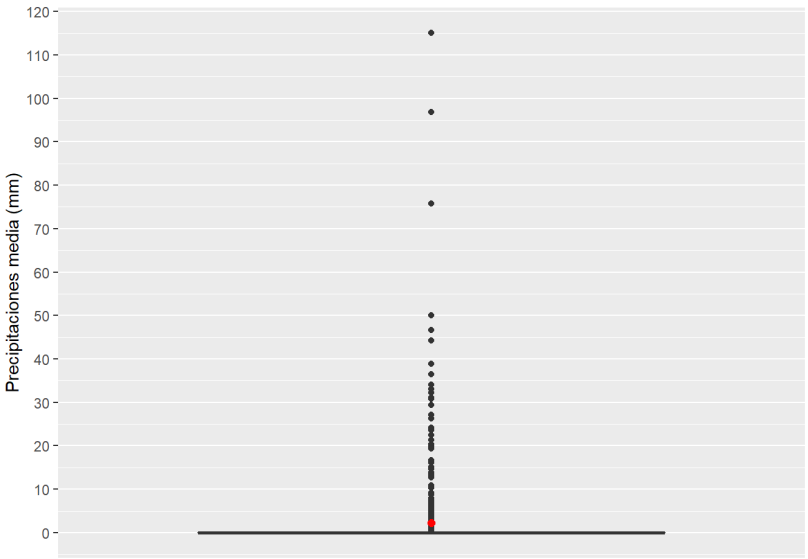

Figura 4-3. Boxplot de la variable prec.

Observamos una gran cantidad de valores atípicos, razón por la cual se decide crear una nueva variable con el tipo de lluvia dependiendo de los mm caídos. Los diferentes

niveles se han extraído de la página web *meteolobios* **[11]** donde se puede encontrar una tabla clasificatoria dependiendo los mm caídos.

Sin lluvia	0 mm
Débil	(0, 2] mm
Moderada	(2, 15] mm
Fuerte	(15, 30] mm
Muy fuerte	(30, 60] mm
Torrencial	>60 mm

Tabla 4-5. Clasificación de la lluvia según los mm caídos (fuente: http://www.meteolobios.es/).

La variable `velmedia` no contiene tantos valores atípicos como la variable `prec`, pero también se ha optado por generar una nueva variable `viento` que categorice el tipo de viento a partir de su velocidad media. Para obtener esta nueva variable se ha consultado la escala *Beaufort y Douglas* **[12]** la cual clasifica los vientos a partir de su velocidad y el oleaje generado.

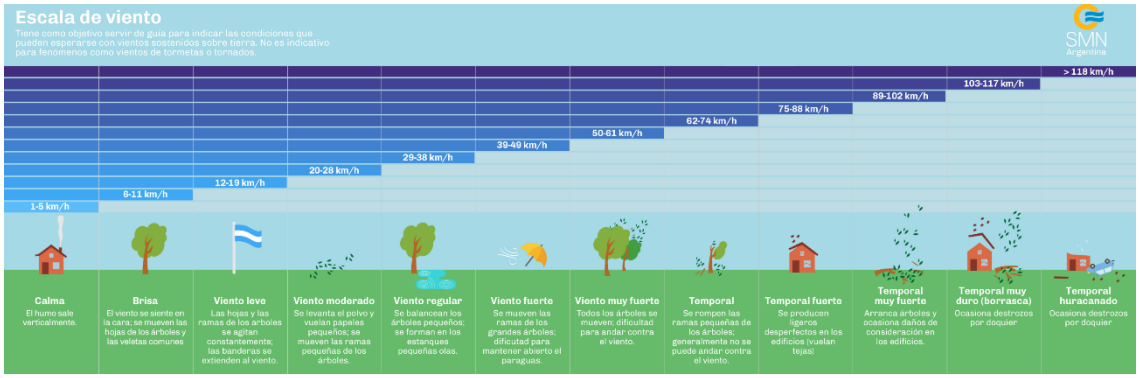

Figura 4-4. Representación gráfica de la escala de Beaufort y Douglas **[12]**.

La variable se descompone en cuatro niveles:

Calma	[0, 6)
Leve	[6, 20)
Moderado	[20, 29)
Regular	[29, 39)

Tabla 4-6. Clasificación del viento según la velocidad media.

4.2.4. Tratamiento de los valores desconocidos.

El data frame data contiene observaciones con diferentes valores desconocidos. A ellos se les debe sumar los que se han generado en los apartados anteriores al reducir los niveles de la variable `sexo`. Estos valores deben ser tratados, de lo contrario, los modelos aplicados pueden aportar errores.

La variable `festividad` está formada, en su mayoría, por valores desconocidos. Recordemos que proviene de un archivo csv generado manualmente en el que únicamente aparecen las fechas festivas, pre-festivas y post-festivas. Todos los demás días del año no son considerados y por lo tanto se les asigna por defecto el valor `NA`. Es por eso por lo que todos los valores desconocidos de esta variable se les asignará el nivel `Laborable`.

Las variables de geolocalización `Long` y `Lat` contienen 17 valores desconocidos. En este caso no se tratarán ya que el uso de estas variables está limitado a la representación de la información a nivel visual en mapas y por lo tanto no se utilizarán para los modelados.

Para acabar, se eliminarán todas las observaciones que contengan valores desconocidos en alguna de las variables: `vehiculo`, `sexo` y `tipo`. En total son 1541 `NA's` que se traducen en 1504 observaciones eliminadas.

```
##   vehiculo          sexo                 tipo
##  Tipo1:15545   Min.   :0.00   Colisión      :10200
##  Tipo2: 6237   1st Qu.:0.00   Alcance       : 6715
##  Tipo3:  761   Median :1.00   Atropello     : 2887
##  Tipo4: 1213   Mean   :0.61   Caida         : 1697
##  NA's :  104   3rd Qu.:1.00   Caída interior:  780
##                Max.   :1.00   (Other)       :  598
##                NA's   :454    NA's          :  983
```

Eliminar 1504 observaciones es un factor que considerar. De todas maneras, el data frame original tiene 23860 observaciones, lo que comportaría una reducción del 6%. Se prefiere obviar este porcentaje de observaciones, antes que asignar valores mediante diferentes técnicas de tratamiento de valores desconocidos e introducir un error en el estudio.

4.2.5. Traducción de variables

Los datos de entrada están en catalán. Ya que el trabajo se ha planteado en castellano, se procede a traducir todos los elementos al castellano. En los apartados anteriores, cuando se han definido nuevas variables, reducido niveles, modificado nombres, etc. ya se han realizado algunas traducciones. Todas las variables que seguían en catalán como `dia_nombre` y `tipo_persona` se han acabado de traducir.

4.2.6. Eliminación de variables

En este punto ya se han realizado todas las modificaciones sobre las variables. El último aspecto de limpieza antes de obtener el data frame definitivo, es el de reducción de las dimensiones. Cabe destacar que en este apartado no se eliminarán variables a partir de conclusiones estadísticas como el estudio de correlaciones, por ejemplo. Simplemente se obviarán variables por su irrelevancia en el estudio.

Las variables `id` y `fecha` se han utilizado para unificar los datos, por lo que ya no son relevantes en el estudio y todavía menos en la creación de los modelos. Para crear las fechas, se han utilizado las variables `dia`, `mes` y `año`. De estas, la última es la menos relevante ya que no interesa hacer un estudio anual, por lo que también se elimina.

Finalmente, las variables numéricas `prec` y `velmedia` también son eliminadas ya que se han utilizado para la creación de las nuevas variables `lluvia` y `viento`.

5. Estudio descriptivo de los datos

Una vez obtenido el data frame *data* definitivo, en este capítulo se realizará un estudio de cada una de las variables que conforman el conjunto de datos. El estudio se dividirá en dos grandes bloques: variables categóricas y variables numéricas.

En este capítulo se explicarán partes muy concretas del código creado. Si se desea comprobar el código entero, se puede encontrar en el ***Anexo: Códigos*** *en la sección* ***C.Código 3: Estudio de los datos.***

5.1. Variables categóricas.

Este tipo de variables se analizarán desde dos puntos de vista: individualmente y por gravedad. Para ello se han empleado medidas de frecuencias.

El primer diagrama muestra la frecuencia de aparición de cada nivel y cada barra se divide entre accidentes graves y leves. En algunos casos, es complejo apreciar el fragmento de gravedad `Grave` debido a la poca proporción; razón por la cual, en el segundo diagrama se muestran los mismos datos de frecuencia, pero por tipo de gravedad, dividiendo el gráfico en dos.

A continuación, se inicia el estudio con la variable dependiente del estudio `gravedad`. En este caso únicamente hay un diagrama de barras mostrando su frecuencia de aparición en el conjunto de datos.

5.1.1. Variable dependiente: `gravedad`

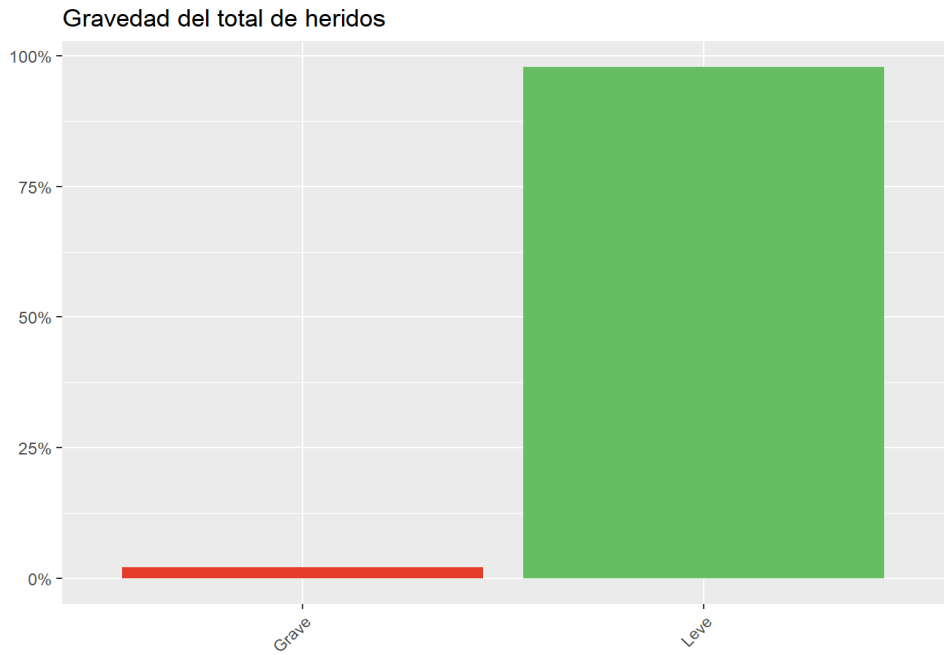

Figura 5-1. Frecuencia de la variable gravedad.

Podemos apreciar como la proporción de heridos graves es insignificante comparada con los heridos leves. En total, en el conjunto de datos hay 466 heridos graves por 21890 leves; es decir, la proporción de heridos graves es únicamente del 2.1%. Éste fenómeno

nada deseado para cualquier estudio, recibe el nombre de desbalanceo de datos, y como se verá en el capítulo **6, apartado 6.2. Desbalanceo de clases**, implica una dificultad extra en el estudio.

5.1.2. Variable `distrito`.

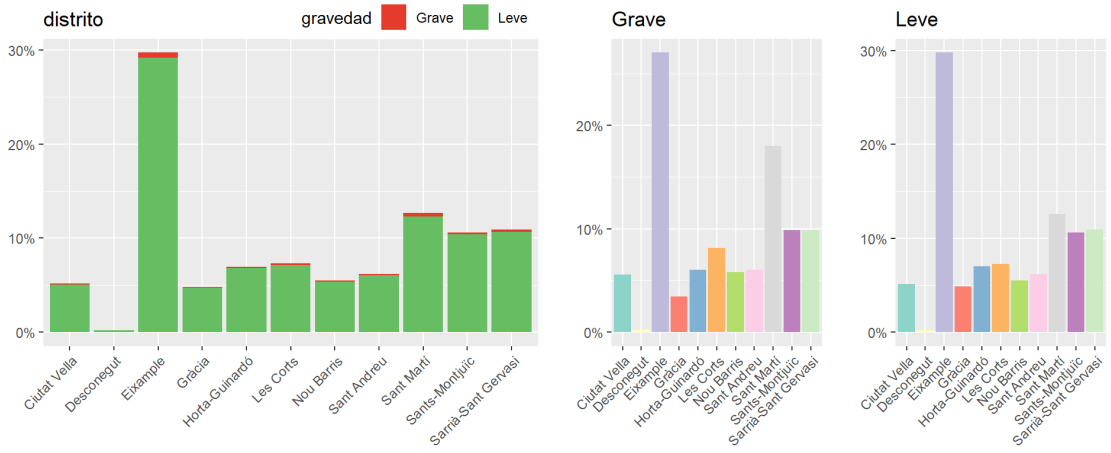

Figura 5-2. Frecuencia de la gravedad por cada distrito.

El distrito que acumula un mayor número de heridos es l'Eixample. A priori, es un dato curioso ya que con 7.46 km^2 no es de los distritos más grandes. Por ejemplo, Sants-Monjuïc y Sarrià-Sant Gervasi superan los 21 km^2 y tienen, aproximadamente, un tercio de los heridos de l'Eixample.

Con la ayuda de las variables de geolocalización `Lat` y `Long`, se han creado visualización utilizando la herramienta Carto.

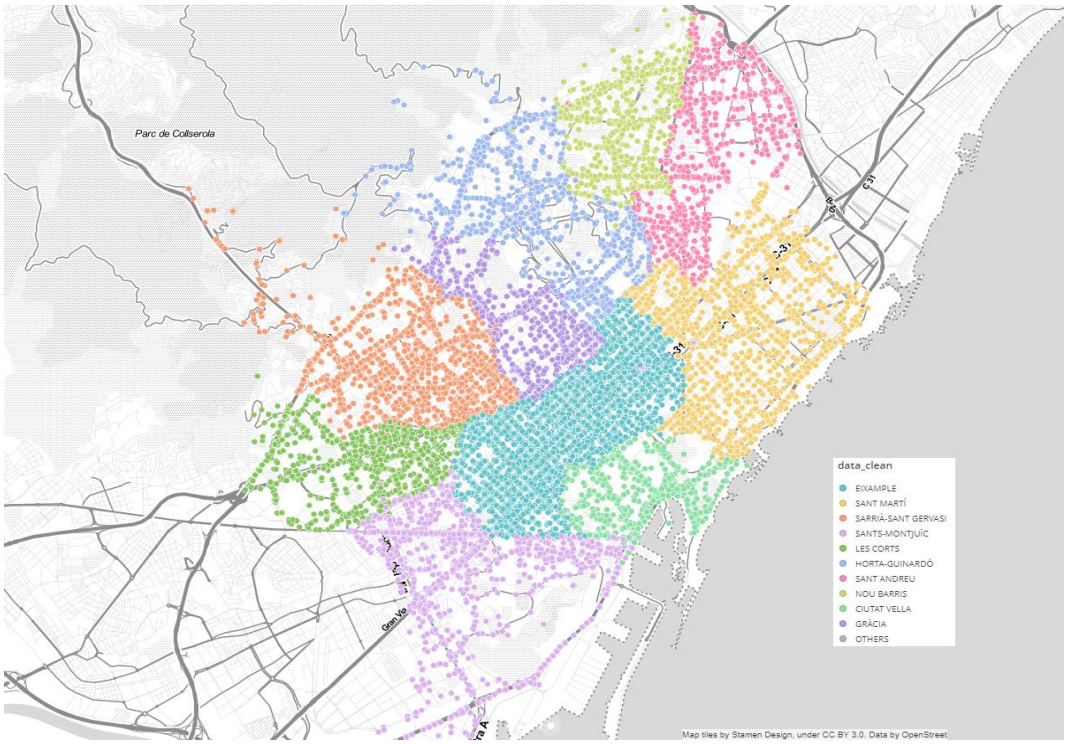

Figura 5-3. Acumulación de heridos por distritos de Barcelona. Imagen creada con Carto.

En la **Figura 5-3** apreciamos claramente que, pese a las dimensiones del distrito de l'Eixample, la densidad de los accidentes es realmente elevada. Debido al elevado volumen de datos, es muy complejo extraer más información de la imagen anterior. Dado que es más importante ubicar los heridos graves que los leves, a continuación, se muestra un mapa de calor con la distribución de los heridos graves en toda la ciudad.

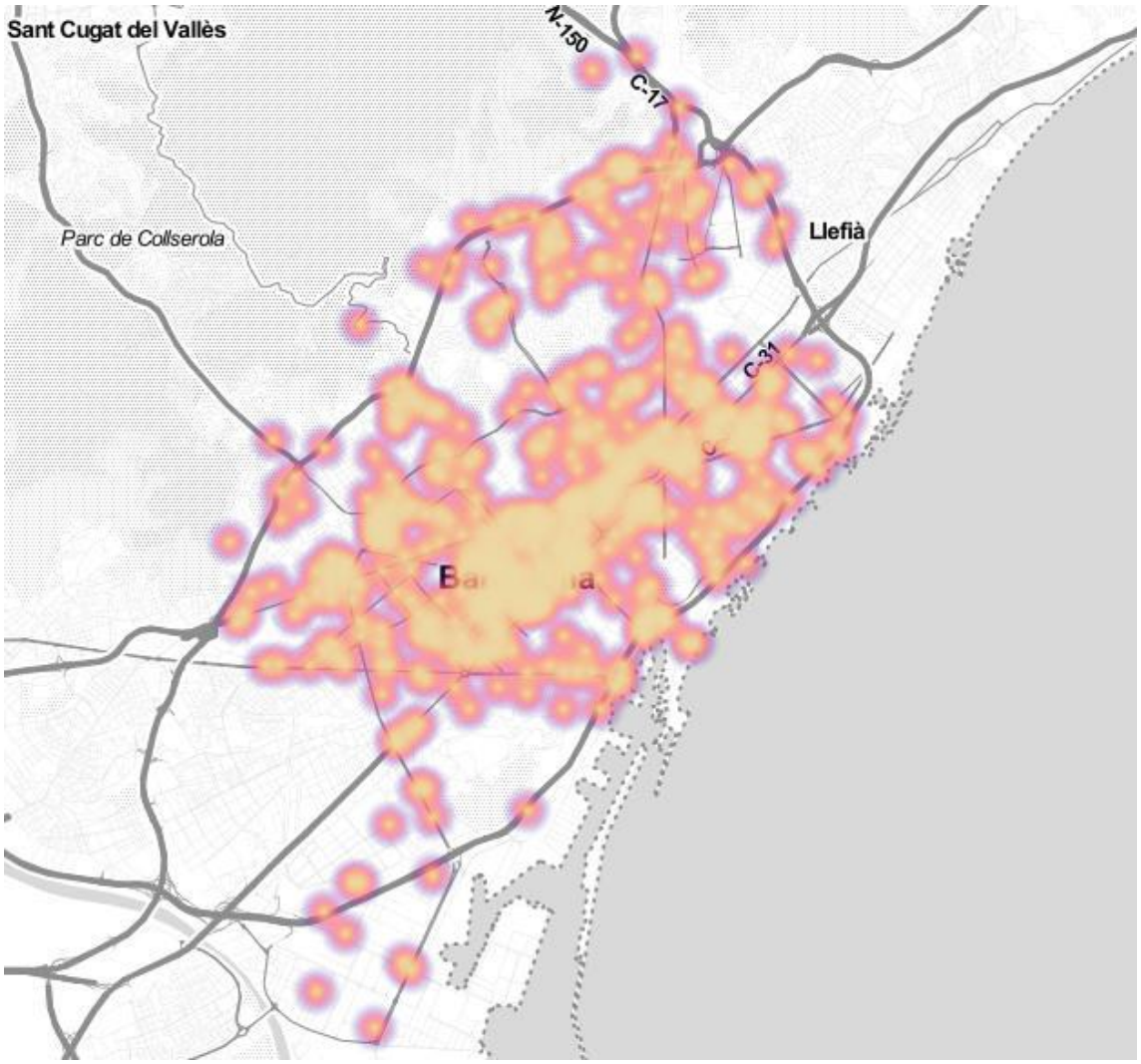

Figura 5-4. Mapa de calor de los heridos graves en Barcelona. Imagen creada con Carto.

Comprobamos que además de ser la zona con mayor densidad de heridos, l'Eixample también acumula el mayor número de heridos graves. Caber destacar que este distrito es de los más importantes de la ciudad ya que por él transcurren vías tan importantes como: Calle Aragón, Meridiana, Gran Vía, Diagonal, Balmes, etc. las cuales acumulan una gran densidad de circulación diaria. Así pues, confirmamos que los accidentes dependen más de las calles que de los distritos. Todo parece indicar que un buen estudio sería a nivel de calle en vez de distrito, pero recordemos que variables tipo factor con muchos niveles provocan un elevado número de grados de libertad, deteriorando el modelo. En el data frame original existían un total de 1401 calles lo cual hace inviable un estudio de la totalidad de la ciudad. Por esta razón, se decide seguir trabajando con los datos de los distritos en vez de las calles.

Siguiendo con el estudio de los accidentes por distritos, es interesante analizar el volumen de heridos en las dos Rondas de la ciudad. La Ronda de Dalt y la Ronda Litoral son dos vías de circunvalación de Barcelona. La primera de ellas recorre el norte de la ciudad y la segunda, todo el litoral marítimo. Debido a su extensión, transcurre por varios distritos. La Ronda de Dalt pasa por: Sarrià-Sant Gervasi, Gràcia, Les Corts, Horta-Guinardó y Nou Barris; mientras que la Ronda Litoral pasa por: Sants-Montjuïc, Ciutat Vella y Sant Martí.

Analizando los datos numéricamente, es interesante ver como la suma de heridos en los distritos de la Ronda Litoral, es muy similar a los heridos únicamente en l'Eixample: 6645. En los tres distritos por donde transcurre la Ronda de Litoral se suman 6358. La Ronda de Dalt, pese a transcurrir por cinco de los diez distritos de la ciudad, tiene 7930 heridos; con 1285 heridos más que l'Eixample.

5.1.3. Variable `dia_nombre`

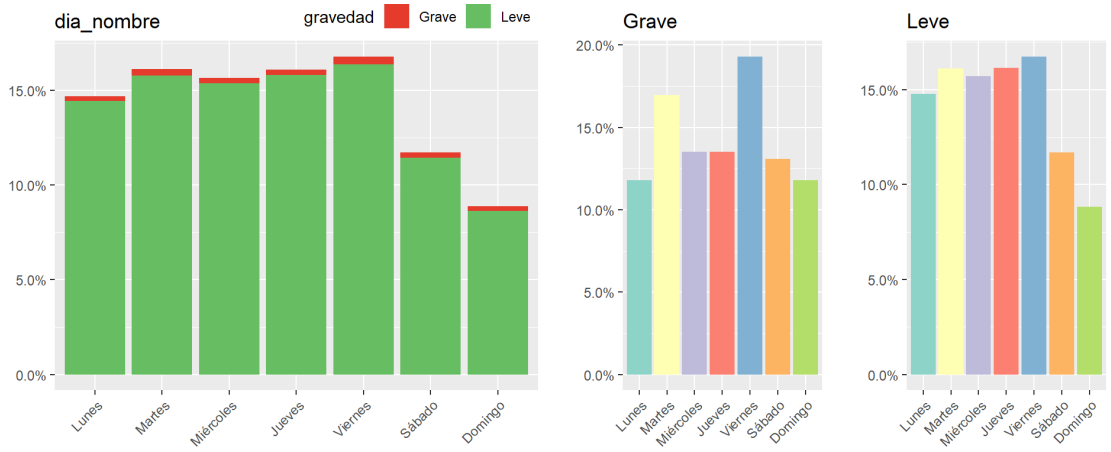

Figura 5-5. Frecuencia de la gravedad por cada día de la semana.

Los fines de semana son los días con menos heridos en los accidentes de tráfico; tanto graves como leves. Durante la semana se mantiene una tendencia constante, pero si nos fijamos por gravedades, podemos comprobar como los martes y viernes, son los dos días donde se acumulan más heridos graves.

5.1.4. Variable `vehiculo`

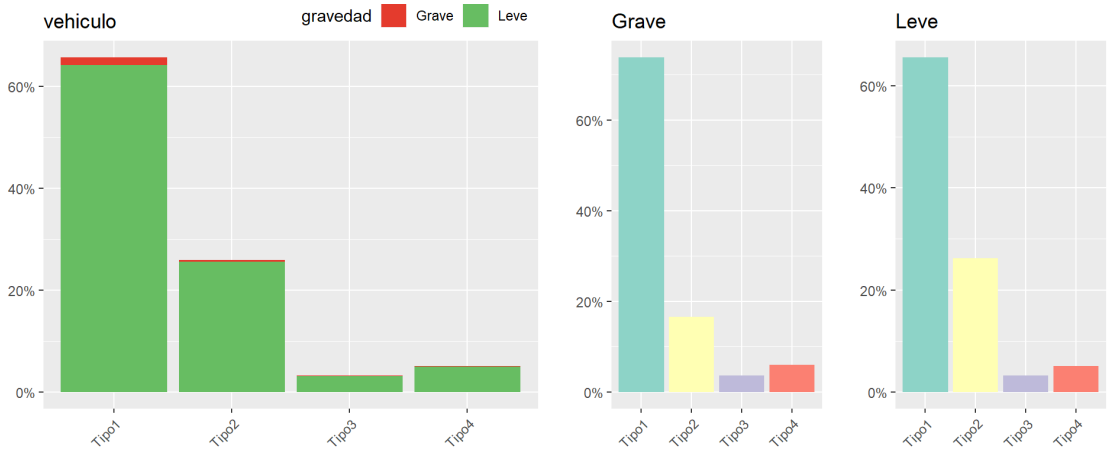

Figura 5-6. Frecuencia de la gravedad por tipo de vehículos.

La tipología de vehículos 1 supera el 60% en la frecuencia de heridos totales; seguido por la tipología 2 que no llega al 30%. Estos datos son coherentes ya que recordemos que la tipología 1 hace referencia a los vehículos de dos ruedas, los ocupantes de los cuales son más susceptibles de sufrir daños.

5.1.5. Variable `tipo_persona`

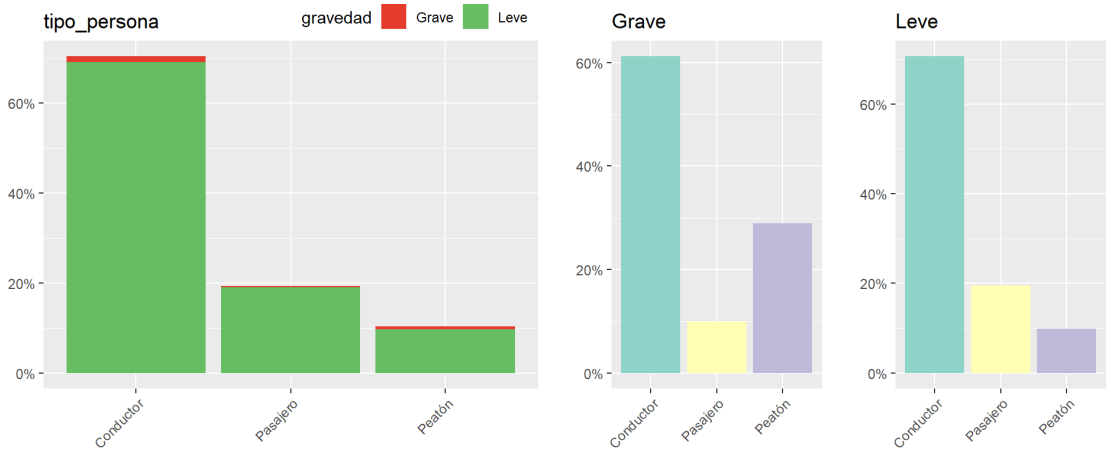

Figura 5-7. Frecuencia de la gravedad por tipos de personas heridas.

Los heridos más frecuentes son los conductores con un 70% aproximadamente. Les siguen los pasajeros con un 20% y los peatones con un 10%. Observando los gráficos por gravedad, los conductores siguen ocupando la primera posición en cuanto a número de heridos; pero hay que destacar que los peatones ocupan el 30% de los heridos graves; es decir casi la mitad de los conductores. De nuevo, encontramos resultados lógicos, ya un atropello tiene un alto grado de gravedad.

Pese a lo dicho en el párrafo anterior, cabe destacar el bajo número de peatones heridos; los cuales son la mitad en número que los pasajeros.

5.1.6. Variable `tipo`

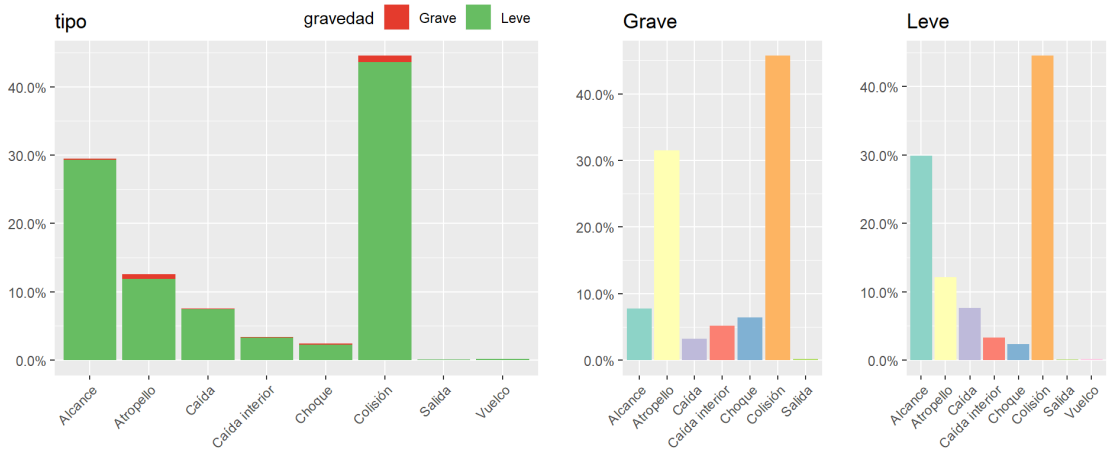

Figura 5-8. Frecuencia de la gravedad por tipología de accidente.

Las colisiones son la tipología de accidentes que causan más heridos tanto graves como leves. A nivel global, los accidentes por alcance son los segundos con más heridos y los atropellos los terceros. En cuanto a la gravedad, el atropello es la segunda tipología más frecuente en los heridos graves, y en los leves, esa posición la ocupa los accidentes por alcance con el mismo porcentaje aproximadamente (30%).

Estos datos coinciden justamente con el análisis hecho en la variable anterior `tipo_persona`, donde los peatones eran el segundo colectivo de heridos con más frecuencia en sufrir lesiones graves. Por otro lado, el alcance (choque trasero), es un tipo de colisión muy peculiar y por eso destaca en los heridos leves.

5.1.7. Variable `causa`

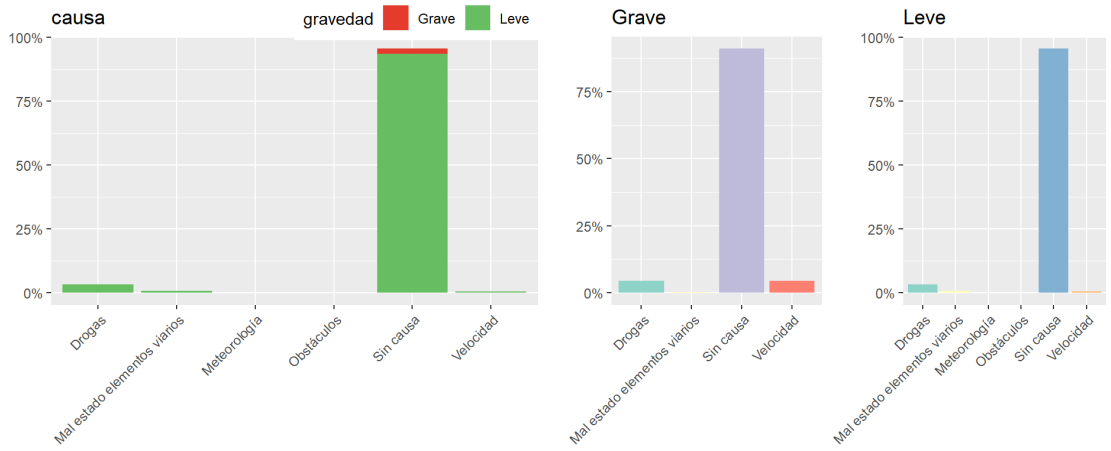

Figura 5-9. Frecuencia de la gravedad por causa.

La mayoría de accidentes no tienen una causa definida tal y como se observan en los gráficos. Dejando de lado este aspecto, las drogas y la velocidad, con menos de un 5%, son las dos causas definidas que aparecen con más frecuencia en los accidentes donde hay heridos graves. Además, no hay observaciones donde las causas meteorológicas o los obstáculos en la vía hayan provocado accidentes graves.

5.1.8. Variable `festividad`

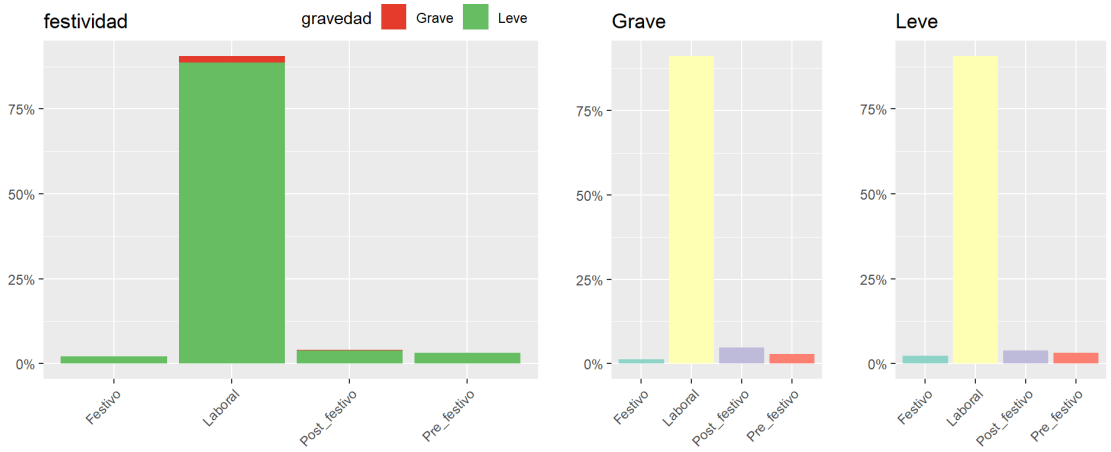

Figura 5-10. Frecuencia de la gravedad por días festivos.

El gran volumen de heridos se produce en días laborables. Se destaca que los días post-festivos sobresalen levemente respecto los días festivos y post-festivos.

5.1.9. Variable `lluvia`

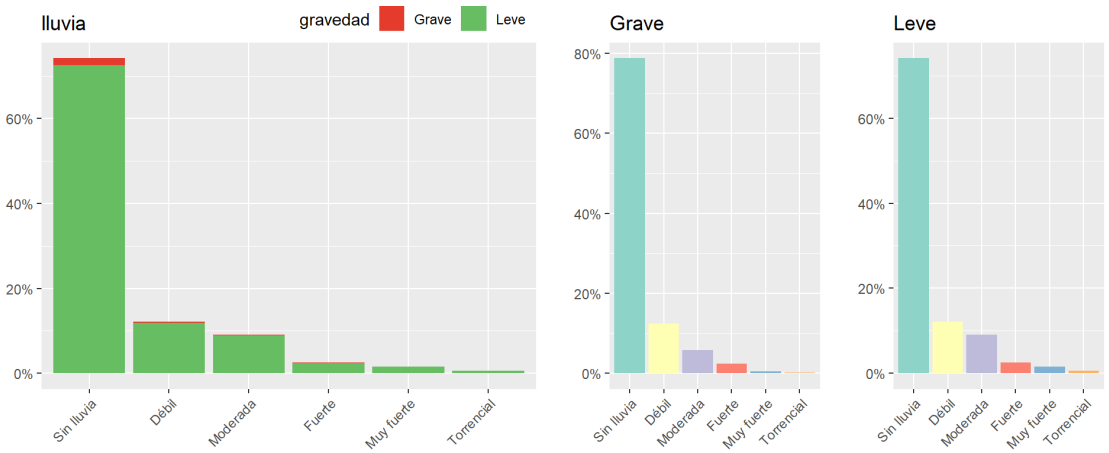

Figura 5-11. Frecuencia de la gravedad por tipo de lluvia.

Los días sin lluvia, son los días donde hay más heridos tanto graves como leves. La lluvia débil es la tipología de lluvia que aparece en segundo lugar con más heridos con un poco más del 10% respecto los otros tipos. A partir de aquí se produce un descenso en la frecuencia de heridos. Es decir; en cuanto más fuerte es la lluvia, menos heridos hay; y más en cuanto menor es la intensidad de la lluvia. Este dato tiene sentido ya que los conductores prestan más atención y tienen una conducción más moderada en cuanto más fuerte es la lluvia.

5.1.10. Variable `viento`

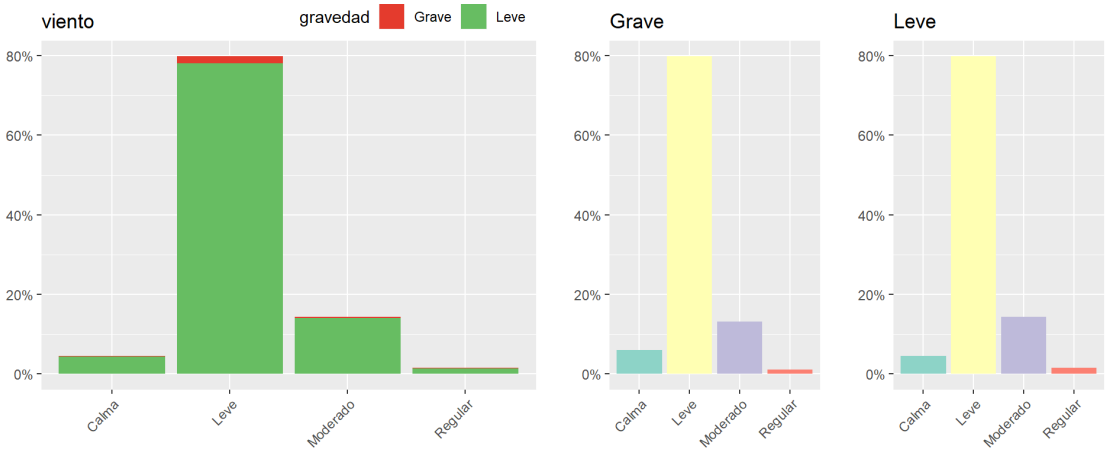

Figura 5-12. Frecuencia de la gravedad por tipo de viento.

El viento no tiene una gran importancia sobre la gravedad de los heridos. Observamos como la mayoría heridos se producen cuando el viento era leve en el momento del accidente.

5.1.11. Variable `causa_peaton`

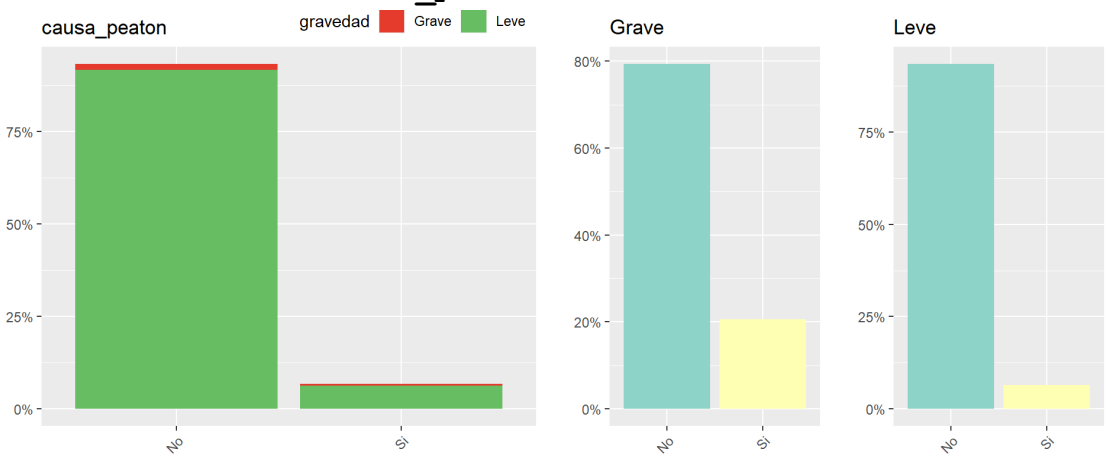

Figura 5-13. Frecuencia de la gravedad por culpabilidad del peatón.

Se observa claramente como los peatones no son los causantes de accidentes con heridos.

5.1.12. Variable `sexo`

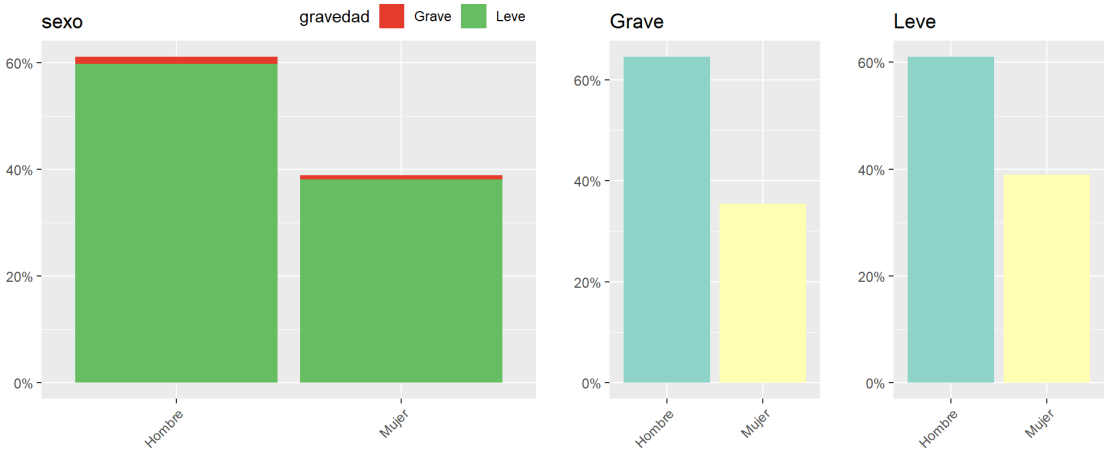

Figura 5-14. Frecuencia de la gravedad por sexo.

De nuevo, la variable binaria sexo, aporta resultados muy claros: El 60% de los heridos en accidentes de tráfico son hombres, respecto al 40% que son mujeres. La misma tendencia se observa en cuanto a la gravedad.

5.2. Variables numéricas

Para el análisis de las variables numéricas, nos basaremos en la exploración de tres gráficos: histograma, boxplot y gráfico de densidad. Éste último, no deja de ser una variación del histograma, ya que muestra la distribución de los datos en un intervalo continuo de tiempo. La diferencia entre ambos es que el diagrama de densidad aplica un suavizado para reducir el ruido.

El histograma mostrará los valores globales; mientras que el gráfico de densidad y el boxplot muestra los valores de la variable por gravedad: `Leve` o `Grave`.

5.2.1. Variable mes

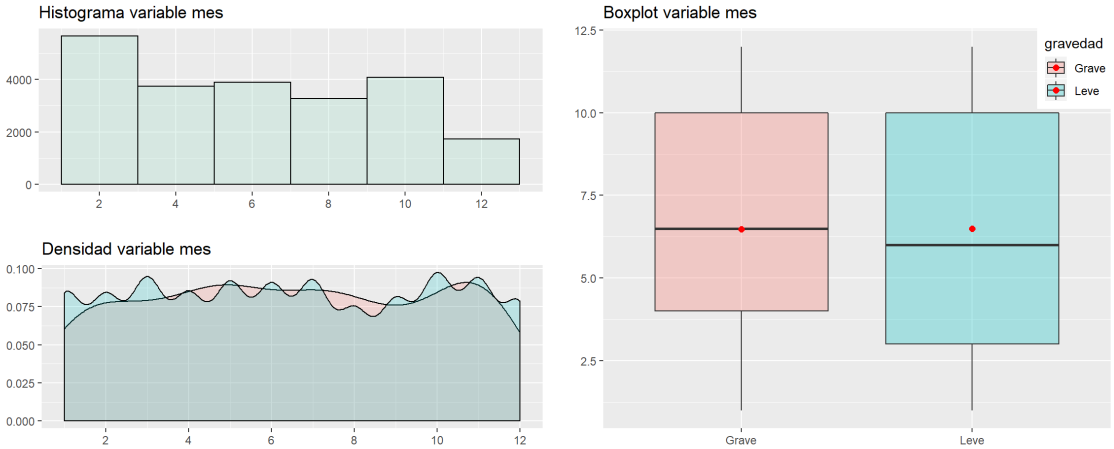

Figura 5-15. Histograma, densidad y boxplot de cada mes.

Durante los meses intermedios se observa una cierta regularidad. A principios de año, el número de heridos augmenta, mientras que a finales disminuye. Según extraemos del diagrama de densidad, los heridos graves muestran una acumulación de valores más regular, pero hacia final de año, decrece notablemente. Esta tendencia la podemos observar en los diagramas boxplot donde el rango intercuartílico RIC es mucho más elevado en los accidentes graves.

5.2.2. Variable dia

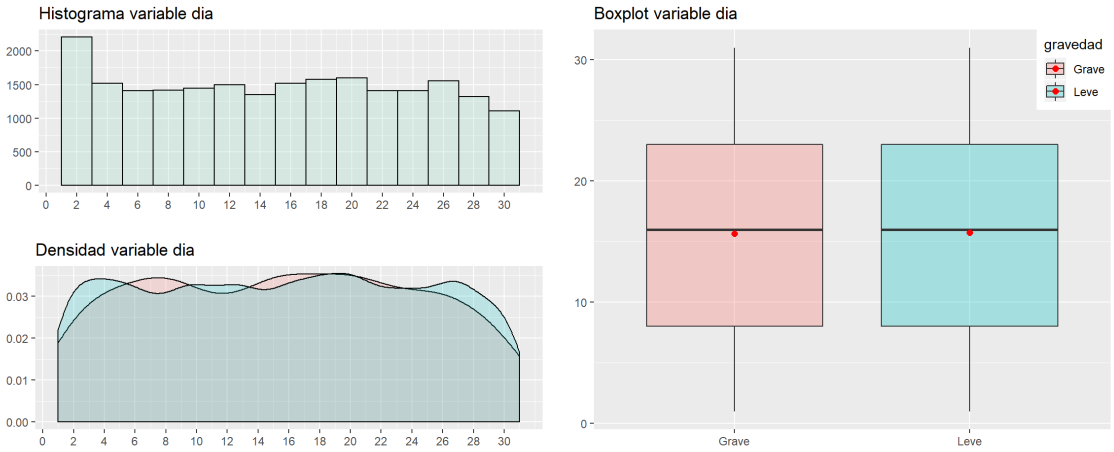

Figura 5-16. Histograma, densidad y boxplot de cada día de la semana.

Los primeros 2-3 días de cada mes hay una acumulación superior de heridos como se observa en el histograma. Del gráfico de densidad se puede extraer que los accidentes leves son más constantes durante todo el mes, mientras que los graves se acumulan más en los días intermedios (del día 6 al 22 aproximadamente).

5.2.3. Variable hora

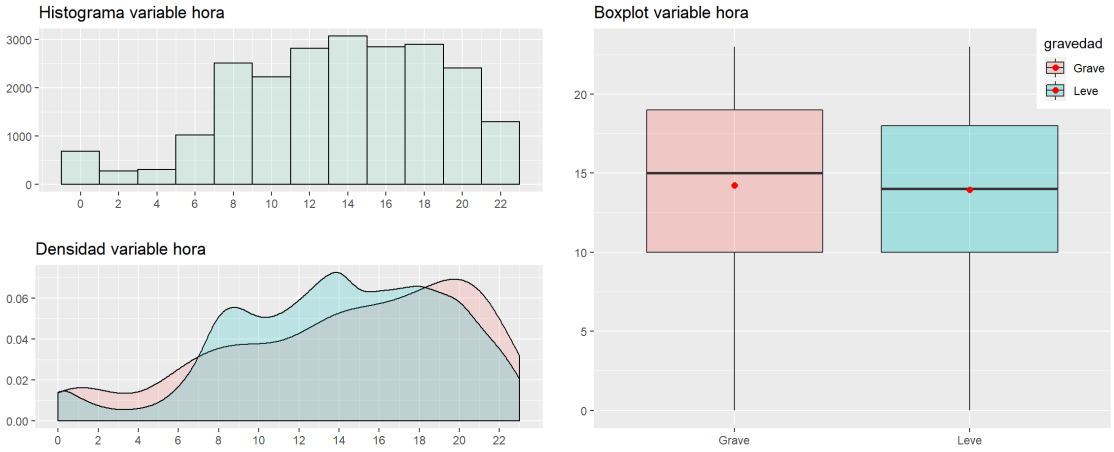

Figura 5-17. Histograma, densidad y boxplot de cada hora del día.

Los accidentes se acumulan en horas diurnas o de alta actividad en la ciudad. El rango horario iría de las 8:00 a las 20:00. Durante estas 12 horas, se acumulan la mayoría de heridos. A partir de las 20:00, el número de heridos disminuye; y entre las 1:00 y las 5:00 los registros son insignificantes. Esta tendencia se confirma en los diagramas de boxplot, donde el rango intercuartílico se desplaza hacia arriba.

Observando le gráfico de densidades, observamos un comportamiento curioso en ambas distribuciones. Mientras que la función de densidad de los heridos leves sigue una forma muy similar a la del histograma, los heridos graves siguen una pendiente positiva desde las 4:00 hasta las 20:00, donde llega al máximo de su valor. Es decir, los heridos graves se acumulan durante la noche, mientras que los leves se acumulan durante el día, alcanzando el valor máximo sobre las 14:00.

5.2.4. Variable tmed

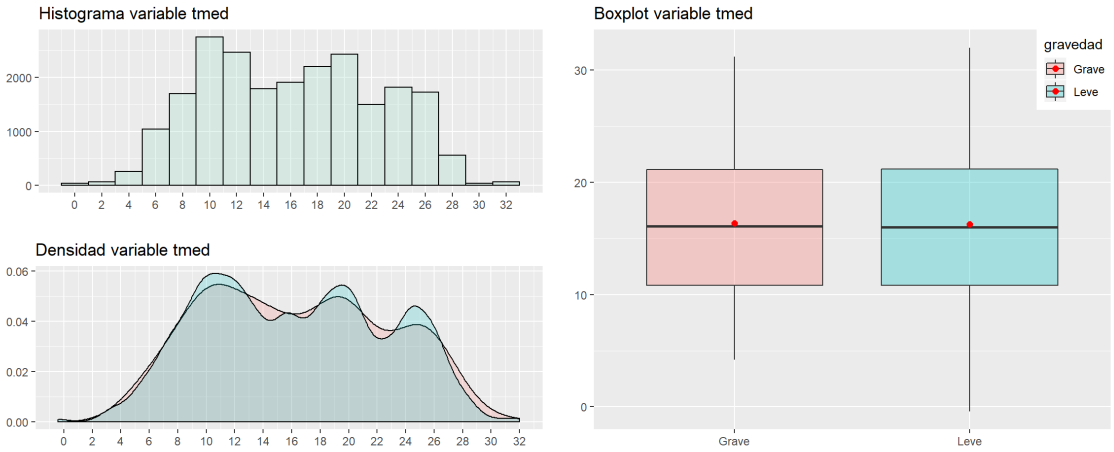

Figura 5-18. Histograma, densidad y boxplot de la temperatura media.

De los gráficos podemos concluir que la temperatura media no influencia sobre la gravedad de los accidentes ya que las distribuciones tanto de los heridos graves como de los leves es prácticamente igual.

Se pueden observar tres picos en las distribuciones, alcanzando niveles máximos a los 10ºC, 20ºC y 24ºC. Esto describe la tendencia de temperaturas durante el año en la ciudad: en épocas más frías, la temperatura media es de 10ºC, durante el verano es de 24ºC y la época de primavera se puede rondar los 20ºC.

6. Preparación de los datos

La preparación de los datos es un paso simple previo al modelado donde se realiza la partición de los datos en entrenamiento y test. El estudio descriptivo de los datos ha revelado que la variable dependiente `gravedad` tiene un claro desbalanceo de clases lo cual obliga a estudiar técnicas que permitan reducir su impacto negativo en la creación del modelo. En este capítulo, aparte de obtener la partición de los datos, se estudiará de qué manera puede influir negativamente el desbalanceo de clases y qué técnicas existen para lidiar con el problema.

En este capítulo se explicarán partes muy concretas del código creado. Si se desea comprobar el código entero, se puede encontrar en el ***Anexo: Códigos*** *en la sección* ***D.Código 4: Preparación de los datos.***

6.1. Conjunto de entrenamiento y test

El data frame *data* obtenido del procesado de los datos, contiene variables de geolocalización útiles para la evaluación descriptiva de los datos, pero que no son relevantes para el modelado. Por eso se eliminan.

En total hay 22356 observaciones, 466 de las cuales pertenecen a heridos graves. Dado que las observaciones se encuentran ordenadas por fecha; es necesario reordenarlas aleatoriamente mediante la función `sample()`. Además, para obtener la partición de datos en entrenamiento y test, se utilizará la función `sample.split()` la cual, a parte de volver a mezclar las muestras, indicándole la variable referencia (`gravedad`) y el ratio de partición (0.7), genera dos particiones donde la proporción de heridos graves y leves sea igual en ambas.

Finalmente se obtienen dos nuevos conjuntos de datos: train (enrenamiento) para entrenar el modelo, y test para evaluarlo. La proporción de heridos graves y leves en cada uno de ellos es la siguiente:

	train	***test***
Grave	326	140
Leve	15323	6567

Tabla 6-1. Heridos graves y leves en cada conjunto de datos train y test.

Comprobamos en la tabla anterior que la proporción en de heridos graves en ambos conjuntos de datos es prácticamente la misma: aproximadamente 2.13%.

6.2. Desbalanceo de clases

El desbalanceo de clases es un fenómeno habitual en conjuntos de datos donde algunas observaciones suceden con muy poca frecuencia. Esto es justo lo que sucede en el data frame que se ha obtenido y por consiguiente, lo encontramos en los conjuntos de test y entrenamiento. Como es normal, los accidentes tienen un mayor número de heridos leves que grave por lo que las observaciones de ambas clases están muy descompensadas.

Este fenómeno provoca una pérdida de rendimiento de los modelos generados. Los principales problemas se deben a **[13]**:

- Existencia de subclases poco representadas (*small disjuncts*): La clase minoritaria, al tener poca representación, puede ser confundida como ruido y, por lo tanto, el modelo descarta estas observaciones durante su creación.
- Confusión o ruido (n*oisy data*): Este problema está relacionado con el anterior, y es que los valores atípicos reales, pueden ser confundidos con la clase minoritaria.
- Falta de densidad en los datos de entrenamiento (l*ack of density*): La clase minoritaria, al no tener una densidad mínima, los algoritmos no pueden crear una generalización; y, por lo tanto, no se puede encontrar un patrón.
- Solape entre clases (c*lass separability problem*): Cuando aparece un solape entre clases en las zonas *fronterizas* provoca que se haga una representación similar entre clases lo que hace imposible la diferenciación entre ambas.
- Separación del conjunto de datos (*dataset shift*): Este problema aparece al separar los datos en conjuntos de test y de entrenamiento ya que la proporción de ambas clases puede ser diferente en los dos nuevos conjuntos de datos. Este problema queda descartado en nuestro caso ya que la partición de los datos se ha hecho mediante la función `sample.split()` la cual mantiene la proporción de la variable seleccionada.

Una vez descartado el problema de *dataset shift*, es necesario tratar el número de observaciones de cada clase mediante técnicas de remuestreo o resampling. En este Trabajo se aplicarán dos técnicas simples como son Upsampling y Downsampling y dos técnicas más complejas como SMOTE y ROSE. Todas las técnicas de remuestreo se deben aplicar sobre el conjunto de entrenamiento ya que de lo contrario estaríamos falseando el conjunto de test, obteniendo evaluaciones erróneas.

6.2.1. Técnica Upsampling

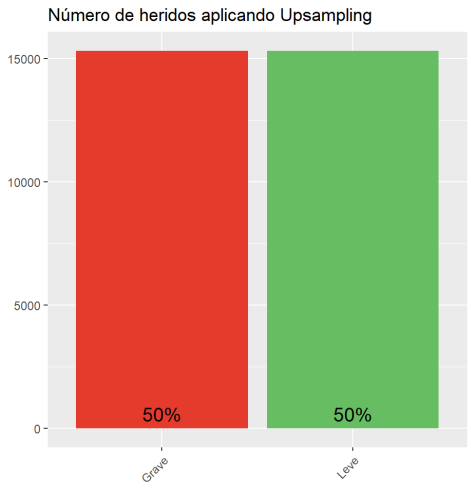

Figura 6-1. Proporción de clases Grave/Leve aplicando Upsampling.

Se repiten aleatoriamente observaciones de la clase minoritaria hasta conseguir el mismo número de observaciones que la clase mayoritaria.

Se ha empleado la función `upSampling()` de la librería `{caret}` **[14]**. El resultado es un nuevo conjunto de datos con el mismo número de observaciones para ambas clases de la variable gravedad: 15323. Así pues, el total de observaciones del nuevo conjunto de datos `train` es de 30646.

6.2.2. Técnica Downsampling

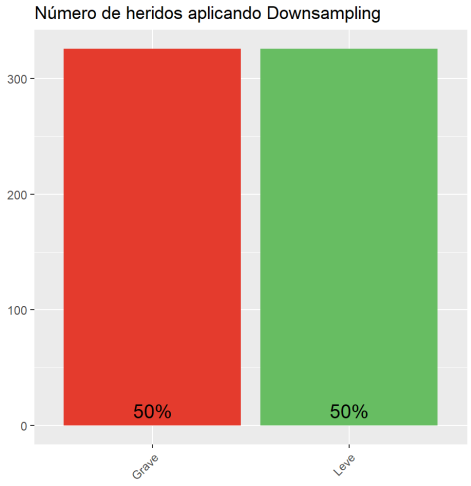

Figura 6-2. Proporción de clases Grave/Leve aplicando Downsampling.

Se eliminan aleatoriamente observaciones de la clase mayoritaria hasta conseguir el mismo número de observaciones que la clase minoritaria.

Se ha empleado la función `downSampling()` de la librería `{caret}`. El resultado es un nuevo conjunto de datos con el mismo número de observaciones para ambas clases de la variable gravedad: 326. Así pues, el total de observaciones del nuevo conjunto de datos `train` es de 652.

6.2.3. Técnica SMOTE

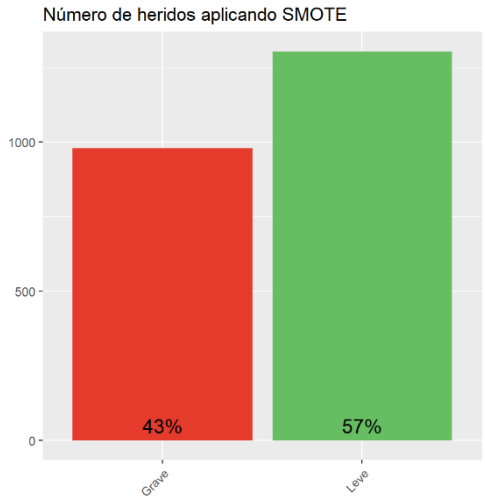

Figura 6-3. Proporción de clases Grave/Leve aplicando SMOTE.

Es una mejora de la técnica Upsampling en la que la clase minoritaria es sobre-muestreada creando ejemplos sintéticos en lugar de sobre-muestrear con sustituciones que ayuda al clasificador a crear regiones de decisión más grandes y menos específicas **[15]**.

Se ha empleado la función `SMOTE()` de la librería `{DMwR}` **[16]**. El resultado es un nuevo conjunto de datos con el mismo número de observaciones para ambas clases de la variable gravedad: 652. Así pues, el total de observaciones del nuevo conjunto de datos `train` es de 1304.

6.2.4. Técnica ROSE

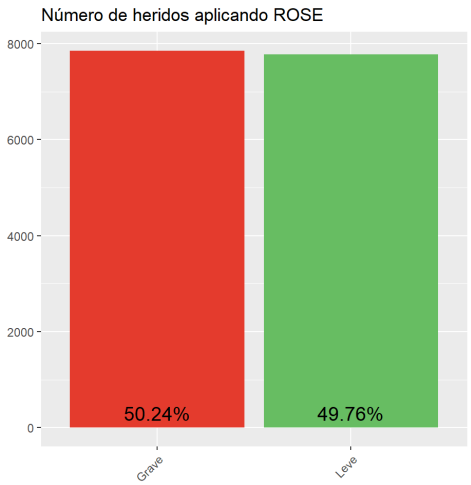

Figura 6-4. Proporción de clases Grave/Leve aplicando ROSE.

Se trata de otra mejora de la técnica Upsampling, generando nuevos ejemplos artificiales de la clase minoritaria siguiendo un método suavizado de Bootstrap **[17]**.

Se ha empleado la función `ROSE()` de la librería `{ROSE}` **[18]**. El resultado es un nuevo conjunto de datos con un número similar de observaciones para ambas clases de la variable gravedad; concretamente 7858 para la clase `Grave` y 7783 para la clase `Leve`. Así pues, el total de observaciones del nuevo conjunto de datos `train` es de 15641.

De esta manera, se obtienen un total de cinco conjuntos de entrenamiento: El conjunto original con clases desbalanceadas y cuatro conjuntos diferentes aplicando cada técnica. Utilizando los cinco conjuntos para entrenar los dos modelos de regresión logística y random forest, se acaba consiguiendo los diez productos mostrados en la tabla **Tabla 1-2** del primer capítulo.

7. Regresión logística

En este capítulo se procede a crear cinco modelos de regresión logística utilizando los conjuntos de entrenamiento obtenidos en el capítulo **6**. Antes de ejecutar el código en R se hará una breve explicación de cómo funciona el algoritmo de regresión logística.

En este capítulo se explicarán partes muy concretas del código creado. Si se desea comprobar el código entero, se puede encontrar en el ***Anexo: Códigos*** *en la sección* ***E.Código 5: Modelado 1 - regresión logística.***

7.1. Modelo teórico

El proceso de regresión es un método estadístico que permite encontrar la relación entre variables independientes y una dependiente. Aplicado en el ámbito de Machine Learning, la regresión permite predecir una variable dependiente a partir de las variables independientes.

El modelo más simple es el de regresión lineal el cual, aproxima la relación entre la variable dependiente y las variables independientes a una recta. A cada una de las variables independientes se les asigna un peso concreto. Este tipo de modelos es muy útil cuando las variables son numéricas, pero cuando nos encontramos ante un problema como el actual donde tenemos variables categóricas, es imposible realizar una aproximación a partir de una recta. Para solucionarlo se hace uso de la regresión logística la cual es especialmente útil en problemas donde la variable resultado únicamente puede adquirir dos valores. La regresión logística convierte la variable respuesta mediante un operador logístico.

$$\ln(^{p}/_{q}) = \alpha_0 + \alpha_1 X_1 + \cdots + \alpha_n X_n$$

Donde a_0, a_1 y a_n son constantes, X_1 y X_n son las variables independientes y $^{p}/_{q}$ son las odd; es decir, la probabilidad de que suceda un evento de la variable dependiente dividido por la probabilidad que no suceda. De nuevo, como en la regresión lineal, cada variable independiente obtiene un peso específico (a_n), pero en este caso en vez de predecirse valores, se predicen probabilidades.

7.2. Aplicación en R

R nos permite modelar regresiones logísticas gracias a la función `glm()` (*Fitting Generalized Linear Models*) **[19]** de la librería `{stats}` **[20]** . La estructura de la función es la siguiente:

```
glm(formula, family = gaussian, data, weights, subset,
    na.action, start = NULL, etastart, mustart, offset,
    control = list(...), model = TRUE, method = "glm.fit",
    x = FALSE, y = TRUE, singular.ok = TRUE, contrasts = NULL, ...
```

Para crear los modelos se tendrán en cuenta los tres siguientes parámetros: `formula`, `data` y `family`.

La fórmula empleada será del estilo `gravedad ~.` ya que la variable dependiente es gravedad y se utilizarán todas las variables independientes.

El campo `data` corresponde a los datos que se utilizan para generar el modelo. Recordemos que se trabajarán con los datos de entreno originales, y con los mismos datos modificados mediante cuatro técnicas de desbalanceo.

El campo `family` es `binomial(link = "logit")`. La función `glm()` no retorna estrictamente un regresión logística; si no que corresponde a un modelo lineal generalizado como su nombre indica: *Fitting Generalized Linear Models*. Para que la regresión sea logística, debe añadirse el valor `link = "logit"` en la estructura de la fórmula. Por otro lado, al tener una variable dependiente que solo puede tener dos valores (`Grave` o `Leve`), el parámetro `family` tiene que ser `binomial`.

Así pues, los cinco modelos son los siguientes:

```
# - Regresión logística con clases desbalanceadas.
lr <- glm(gravedad ~., data = train,
          family = binomial(link = "logit"))

# - Regresión logística aplicando Upsampling sobre train
lr.up <- glm(gravedad ~., data = train.up,
          family = binomial(link = "logit"))

# - Regresión logística aplicando Downsampling sobre train
lr.dn <- glm(gravedad ~., data = train.dn,
          family = binomial(link = "logit"))

# - Regresión logística aplicando SMOTE sobre train
lr.sm <- glm(gravedad ~., data = train.sm,
          family = binomial(link = "logit"))

# - Regresión logística aplicando ROSE sobre train
lr.rs <- glm(gravedad ~., data = train.rs,
          family = binomial(link = "logit"))
```

8. Random forest

En este capítulo se procede a crear cinco modelos random forest utilizando los conjuntos de entrenamiento obtenidos en el capítulo **6**. Antes de ejecutar el código en R se hará una breve explicación de cómo funciona el algoritmo de random forest.

En este capítulo se explicarán partes muy concretas del código creado. Si se desea comprobar el código entero, se puede encontrar en el ***Anexo: Códigos*** *en la sección* ***F.Código 6: Modelado 2 - random forest.***

8.1. Modelo teórico

Random forest **[21]** es un algoritmo de aprendizaje supervisado que mejora los árboles de decisión más simples. La diferencia con otros modelos es que se generan un gran número de árboles de decisión independientes a partir de subconjuntos de los datos de entrada aleatorios, pero de igual distribución. Para la construcción de cada árbol ***i*** se realizan los siguientes pasos:

1- Considerando un conjunto de datos de entrenamiento con ***N*** observaciones, se extrae un subconjunto aleatoriamente, ***n***, con reemplazo para poder construir el árbol de decisión ***i***.
2- En cada nodo se obtiene un subconjunto aleatorio ***m*** de la ***M*** variables de entrada. El conjunto de variables predictoras que genere una mejor partición del nodo serán las elegidas para aquel nodo. Este proceso se repite en todos los siguientes nodos del árbol ***i*** con diferentes subconjuntos ***m***.
3- Cada árbol ***i*** crece tanto como sea posible y no hay proceso se poda.

Finalmente, cada árbol clasifica las diferentes clases a predecir. Al final, la clase con mayor número de predicciones en todos los árboles será el resultado del algoritmo. Este proceso también es conocido como votar; por lo que el voto mayoritario, indicará la clase resultante.

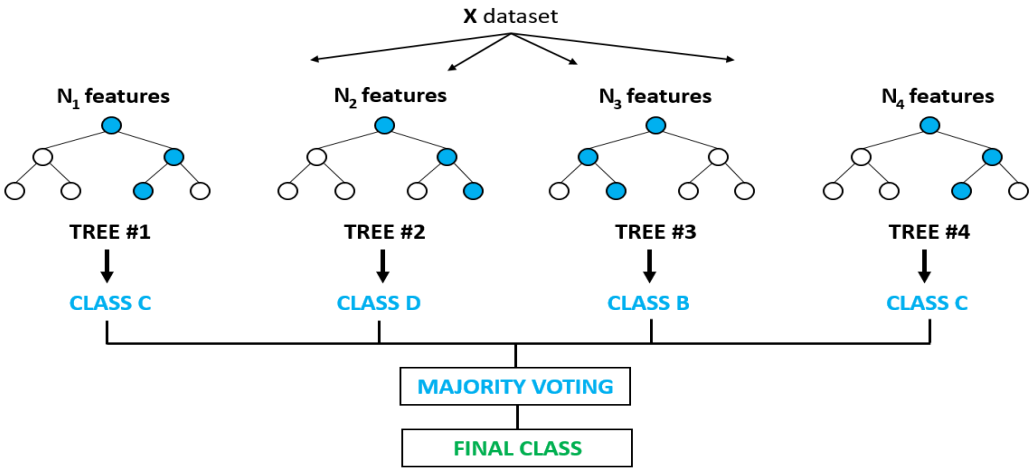

Figura 8-1. Funcionamiento random forest.
(fuente: https://www.globalsoftwaresupport.com/random-forest-classifier/)

En la Figura **8-1** se muestra de forma simple el algoritmo de random forest creado a partir de 4 árboles N_i. En cada nodo se eligen diferentes variables, por eso observamos que los caminos seguidos son distintos. Cada árbol predice una clase o voto, que finalmente, se utilizan para encontrar el resultado final.

Para obtener los subconjuntos de observaciones ***n***, se aplica un proceso llamado Bootstrap. Este proceso de muestreo con reemplazo (pueden aparecer observaciones repetidas en un mismo subconjunto) se realiza para cada uno de los árboles, y en cada caso, se reserva 1/3 del total de observaciones para test llamadas “out of bag” (OOB). Los OOB serán utilizados en cada árbol para calcular el error.

Figura 8-2. Proceso de Bootstrap.
(Fuente: Sebastian Raschka - https://sebastianraschka.com/blog/2016/model-evaluation-selection-part2.html)

8.2. Aplicación en R

La función `randomForest()` de la librería `{randomForest}` **[22]** será la utilizada en este capítulo para crear el modelo. La estructura de la función es la siguiente:

```
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             maxnodes = NULL,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest),corr.bias=FALSE,
             keep.inbag=FALSE, ...)
```

8.2.1. Elección de los hyper-parámetros

En la construcción del modelo, los dos hyper-parámetros más relevantes son la elección del número de árboles `ntree` y el número de variables aleatorias que serán tratadas en cada nodo `mtry`.

El mismo paquete `{randomForest}` incluye la función `tuneRF()` muy útil para encontrar los mejores hyper-parámetros a aplicar a la función `randomForest()`. A continuación, se muestra su estructura:

```
tuneRF(x, y, mtryStart, ntreeTry=50, stepFactor=2, improve=0.05,
       trace=TRUE, plot=TRUE, doBest=FALSE, ...)
```

Esta función realiza un proceso iterativo donde, partir de unos datos de entrenamiento de entrada y un número concreto de árboles, va calculando en todo momento los errores OOB utilizando valores aleatorios de `mtry`. El proceso se detiene en el momento que el error OOB relativo al calculado en la iteración anterior, es inferior al especificado en su hyper-parámetros `improve`.

Para elegir los mejores hyper-parámetros para `randomForest()` se decide crear una función `mejores.parametrosRF()`, la cual a partir de un conjunto de datos de entrenamiento, ejecuta la función `tuneRF()` tantas veces como árboles se quieran analizar. Concretamente se decide evaluar 9 valores de `ntree` diferentes: 200, 300, 400, 500, 600, 700, 800, 900 y 1000. En cada iteración se almacena el error OOB, el número de árboles y el número de variables utilizadas. Finalmente, la función retorna la fila que minimiza el error OOB, obteniendo los hyper-parámetros óptimos para aplicar en la función `randomForest()`. Adicionalmente, también se retorna un gráfico como el siguiente, donde se muestran todos los errores OOB por `ntree` y `mtry`.

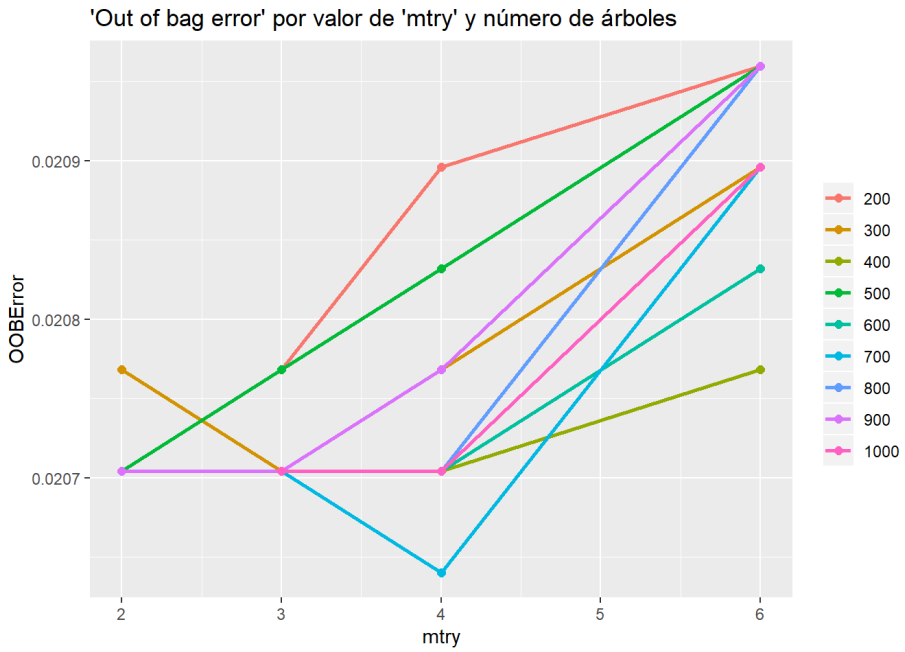

Figura 8-3. Errores OOB por cada ntree y mtry.
Gráfico generado por la función `mejores.parametrosRF()` para clases desbalanceadas.

El código entero de `mejores.parametrosRF()` también está incluido en el **Anexo: Códigos**, sección **F.Código 6: Modelado 2 - random forest.**

En la Figura **8-3**, que hace referencia a los datos de entrenamiento sin aplicar ninguna técnica de balanceo, se puede observar como la función genera 9 valores de `ntree` de 200 a 1000. Generando un modelo de 700 árboles y eligiendo 4 variables de forma aleatoria en cada nodo, se obtiene el menor error OOB. A nivel numérico, la función retorna los siguientes resultados:

```
##        mtry  OOBError tree
## 4.OOB5    4 0.0206403  700
```

La función `mejores.parametros()` se ha ejecutado también para los conjuntos de entrenamiento a los que se les ha aplicado las cuatro técnicas de balanceo de clases (Upsampling, Downsampling, SMOTE y ROSE). Para no añadir información redundante, se decide no adjuntar los gráficos obtenidos y mostrar los valores numéricos retornados:

```
             ##        mtry     OOBError tree
Upsampling   ## 4.OOB7    4 0.0005873523  900
Downsampling ## 4.OOB1    4    0.3159509  300
SMOTE        ## 3.OOB4    3    0.1722174  600
ROSE         ## 4.OOB5    4   0.04939613  700
```

De los resultados obtenidos, observamos que el número de variables elegidas en cada nodo tiende a 4. Esto coincide con la recomendación de Breiman **[21]** donde afirma que el valor de `mtry` debe ser aproximadamente $\sqrt[2]{número\_total\_predictores}$. Teniendo en cuenta que el conjunto de datos tiene 16 variables predictoras, un valor de `mtry` cercano a 4 debe minimizar el error OOB.

Antes de aplicar cada modelo, se llama a la función `mejores.paramterosRF()`. Los resultados retornados se incluyen automáticamente en los hyper-parametros `mtry` y `ntree` de la función `randomRF()` tal y como se muestra a continuación.

```
# - Random Forest con clases desbalanceadas.
mejores.rf <- mejores.parametrosRF(train)

rf <- randomForest(gravedad~., data = train,
        ntree = mejores.rf$tree[1],
        mtry = mejores.rf$mtry)

# - Random Forest aplicando Upsampling sobre train
mejores.rf.up <- mejores.parametrosRF(train.up)
rf.up <- randomForest(gravedad~., data = train.up,
        ntree = mejores.rf.up$tree[1],
        mtry = mejores.rf.up$mtry[1])

# - Random Forest aplicando Downsampling sobre train
mejores.rf.dn <- mejores.parametrosRF(train.dn)
rf.dn <- randomForest(gravedad~., data = train.dn,
        ntree = mejores.rf.dn$tree[1],
```

```
        mtry = mejores.rf.dn$mtry[1])

# - Random Forest aplicando SMOTE sobre train
mejores.rf.sm <- mejores.parametrosRF(train.sm)
rf.sm <- randomForest(gravedad~., data = train.sm,
        ntree = mejores.rf.sm$tree[1],
        mtry = mejores.rf.sm$mtry[1])

# - Random Forest aplicando ROSE sobre train
mejores.rf.rs <- mejores.parametrosRF(train.rs)
rf.rs <- randomForest(gravedad~., data = train.rs,
        ntree = mejores.rf.rs$tree[1],
        mtry = mejores.rf.rs$mtry[1])
```

9. Métricas

Este capítulo previo al estudio de los modelos resultantes pretende introducir qué métricas son las más adecuadas para evaluar modelos donde las clases de su variable dependiente están desbalanceadas y como interpretarlas correctamente.

Dado que los diez modelos generados deben ser evaluados a partir de las misma métricas, se ha creado una función común `evaluacion()` que permitirá obtener los resultados para cada uno de los modelos creados; ya sean modelos de regresión logística o modelos random forest. Se explicará brevemente los puntos más importantes de esta función.

9.1. Métricas para evaluar clases desbalanceadas.

En el estudio de los modelos predictivos se considera la exactitud o *accuracy* como un parámetro básico para determinar la calidad del modelo. Basarse únicamente en este parámetro cuando se está trabajando con clases desbalanceadas, puede conducir a extraer conclusiones erróneas. Comprobémoslo con un simple ejemplo aprovechando la gravedad de las víctimas de los accidentes:

Figura 9-1. Ejemplo de matriz de confusión con clases desbalanceadas.

La figura de la izquierda muestra el resultado de una matriz de confusión cualquiera, típica en problemas donde las clases están desbalanceadas. El eje horizontal corresponde a las predicciones mientras que el vertical corresponde a los valores de referencia o test. Si calculamos la precisión a partir de estos datos, obtenemos el siguiente resultado:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.9805$$

Una exactitud del 98% es realmente buena, pero eso quiere decir que el modelo es bueno? Fijémonos que de 20 casos reales donde hay heridos graves (TP + FN), el modelo detecta únicamente 5 y, además, predice 113 graves (TP + FP). Detectando únicamente el 25% de clases positivas, no podemos decir que el modelo sea bueno. Así pues, en casos donde las clases se encuentran desbalanceadas, se recomienda combinar diferentes medidas para evaluar los modelos **[23]**. A parte de la exactitud, en una matriz de confusión, los valores de falsos y verdaderos positivos y negativos, permiten extraer otras métricas como las explicadas a continuación:

Exhaustividad o *sensitivity*; también conocida como recall o true positive rate (TPR): Describe la capacidad del modelo para clasificar la clase positiva.

$$TPR = \frac{TP}{TP + FN}$$

Especificidad o *specificity* también conocida como selectividad o true negative rate (TNR): Describe la capacidad del modelo para clasificar la clase negativa.

$$TNR = \frac{TN}{TN + FP}$$

Precisión o *positive predictive value* (PPV): Describe la capacidad del modelo para predecir la clase positiva. Observamos que es diferente a la exhaustividad, ya que mientras esta evaluaba cuántas observaciones son clasificadas por el modelo como positivas, la precisión se centra en el total de clases positivas predichas.

$$PPV = \frac{TP}{TP + FP}$$

F1 score: Esta métrica relaciona las medidas de precisión y exhaustividad; asumiendo que interesa por igual ambas medidas. Esto quiere decir, que es igual de importante que el modelo detecte las clases positivas como que las clasifique correctamente.

$$F1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}$$

A parte de las cinco métricas anteriores, también se analizará la curva ROC **[24]** ya que es una forma interesante de evaluar visualmente la exhaustividad y especificidad. También se retornará el valor AUC (área bajo la curva) como métrica de estudio adicional. Establecer qué valor debe tener para considerar que el modelo es bueno dependerá de la naturaleza de la predicción. Por ejemplo, en el ámbito de la medicina, los diagnósticos deben ser lo más precisos posible; por eso se acostumbra a utilizar un valor de referencia muy elevado, superior al 90%. Teniendo en cuenta que la finalidad del trabajo actual es detectar la gravedad de los heridos, es importante que el valor AUC también se elevado. Se considerará un buen modelo a partir del 80%.

9.2. Método de evaluación.

En el apartado anterior se han definido las cinco métricas que se emplearán para la evaluación de los modelos: exactitud, exhaustividad, especificidad, precisión y F1 score; así como curvas ROC.

El siguiente paso es plantear un método para combinar todas estas métricas y poder evaluar los modelos creados. Para ello, debemos tener presente el objetivo principal de este proyecto: La predicción de heridos graves y leves en accidentes de tráfico. De esta frase ya podemos extraer que la capacidad de predicción las clases (precisión) tiene una gran importancia.

Además, hay que tener presente que la finalidad de los modelos creados es aportar una herramienta más a los servicios de emergencia para saber con qué tipos de heridos se encontrarán cuando acudan a un accidente. Este aspecto es muy importante ya dependiendo la gravedad de los heridos se desplegará un dispositivo u otro. De esta manera, acudir a un accidente con un dispositivo donde se predicen heridos leves cuando en realidad son graves, es un error muy importante y que debe ser minimizado. En cambio, en el caso contrario, si se predice un accidente con heridos graves cuando en realidad son leves, el error es menor, ya que no hay vidas en juego, pero sí que hay un desperdicio de recursos. Por esta razón, se debe valorar la capacidad de clasificación de las clases; dando máxima importancia a la clase positiva `Grave` (exhaustividad).

Las curvas ROC son un buen elemento visual para comprobar la relación entre clasificación de clases positivas y negativas (exhaustividad vs. especificidad).

Para acabar, se pueden aunar estos dos conceptos de clasificación y predicción mediante el valor retornado por el F1 score.

Así pues, las evaluaciones se centrarán en la capacidad de predicción y la capacidad de clasificación, dándole máxima importancia a la exhaustividad de los modelos.

9.3. Funcion evaluacion()

En total se van a obtener diez modelos (cinco para regresión logística y 5 para random forest). Obtener las métricas descritas anteriormente, de forma separada por cada modelo, generaría un código absurdamente largo, por lo que se plantea crear una función llamada `evaluacion()` que calcula todas estas medidas tanto para los modelos de regresion logística como para los random forest. Evaluar dos modelos donde los tipos de resultados obtenidos son diferentes, tiene algunos aspectos importantes a considerar que serán explicados en este apartado.

9.3.1. Predicciones

Para evaluar un modelo, se deben encontrar las predicciones a partir de un conjunto de test y del propio modelo. La función encargada de ello en R es `predict()` la cual, en el caso de modelos random forest, retorna las predicciones con las clases `Grave`/`Leve` ya definidas o sus probabilidades, mientras que con modelos de regresión logística únicamente retorna la probabilidad que una de las clases suceda. De las dos opciones, es mucho más preciso trabajar con probabilidades, pero desgraciadamente, comporta un problema para la obtención de la matriz de confusión, ya que esta se genera a partir de las clases predichas. Así pues, al trabajar con probabilidades, se debe hacer un paso previo y obtener manualmente las clases. Generalmente, se utiliza un punto de corte del 50% donde valores de probabilidad inferior a 0.5 pertenecen a una clase y valores igual o superior a 0.5 pertenecen a la otra clase. Este valor no deja de ser un valor estándar elegido al azar, así pues, se plantean otras formas más precisas que permitan encontrar un punto de corte óptimo.

Dado que la finalidad de este proceso es diferenciar ambas clases, se ha decidido trabajar con métricas de exhaustividad y especificidad; de manera que, el punto de corte será aquel que maximice la exhaustividad y la especificidad. La función `roc()` permite extraer ambos parámetros por separado, así como el punto de corte asociado a cada par. Para encontrar el punto de corte, por cada observación obtenida de la función `roc()`, se suman los valores de exhaustividad y especificidad, obteniendo una columna con la suma y añadiendo una nueva con todos los valores de threshold. El valor máximo de la columna suma retornará el valor de corte de la columna thresholds. El siguiente código muestra el proceso, y la definición de las clases a partir del punto de corte.

```
# p.prob es la predicción en probabilidades.
analisis <- roc(response = test$gravedad, predictor = p.prob,
                quiet = "TRUE")
c <- cbind(analisis$thresholds,
           analisis$sensitivities + analisis$specificities)
```

```
corte_optimo <- subset(c, c[,2] == max(c[, 2]))[,1]
p.clase <- as.factor(ifelse(p.prob > corte_optimo, "Leve", "Grave"))
```

9.3.2. Métricas obtenidas

Con las predicciones obtenidas (clases y probabilidades) se pueden obtener todos los parámetros, gráficos y matrices comentadas en el apartado anterior.

La matriz de confusión se construye a partir de la predicción de las clases y un conjunto de test (referencia). La función `confusionMatrix()` de la librería `{caret}`, aparte de retornar la matriz, también devuelve parámetros que interesan como: exactitud, exhaustividad, especificidad, precisión. Con ellos, también se calcula el valor de F1.

A parte de devolver los valores de exhaustividad y especificidad, la función `roc()` también retorna el valor del área bajo la curva ROC. Con los dos primeros valores, se procede a crear el gráfico de la curva ROC.

La función `evaluación()` retorna una lista de tres posiciones:

1- Data frame con los parámetros: exactitud, exhaustividad, especificidad, precisión, F1y AUC.
2- Matriz de confusión.
3- Gráfico curva ROC.

10. Resultados

En este capítulo se procede a aplicar la función `evaluación()` con todos los modelos generados anteriormente en el capítulo **7** y capítulo **8**. Primero se evaluarán los modelos por tipologías; es decir, primero se evaluarán los modelos de regresión logística y después los random forest. Finalmente se comparan los resultados de ambos modelos para detectar si hay alguna mejora.

Todos los modelos se evaluarán a partir de la capacidad de predicción (precisión) y la capacidad de clasificación (exhaustividad y especificidad) tal y como se ha especificado en el capítulo anterior en el apartado **9.2. Método de evaluación.**

10.1. Resultados regresiones logísticas.

A continuación, se muestra la tabla resumen con los resultados de cada modelo:

```
##              Accuracy Sensitivity Specificity   PPV         F1   AUC
## Imbalanced      0.759       0.629       0.761 0.053 0.09805014 0.744
## Upsampling      0.658       0.736       0.656 0.044 0.08243297 0.752
## Downsampling    0.598       0.771       0.594 0.039 0.07417582 0.721
## SMOTE           0.719       0.614       0.721 0.045 0.08353570 0.716
## ROSE            0.711       0.671       0.712 0.047 0.08851224 0.744
```

Tabla 10-1. Métricas obtenidas para cada modelo de regresión logística.

Para cada modelo se muestran las matrices de confusión:

Imbalanced

```
##           Reference
## Prediction Grave Leve
##      Grave    88 1567
##      Leve     52 5000
```

Upsampling

```
##           Reference
## Prediction Grave Leve
##      Grave   103 2256
##      Leve     37 4311
```

Downsampling

```
#           Reference
## Prediction Grave Leve
##      Grave   108 2664
##      Leve     32 3903
```

SMOTE

```
##           Reference
## Prediction Grave Leve
##      Grave    86 1833
##      Leve     54 4734
```

ROSE

```
##           Reference
## Prediction Grave Leve
##      Grave    94 1890
##      Leve     46 4677
```

Tabla 10-2. Matrices de confusión para cada modelo de regresión logística.

10.1.1. Capacidad de predicción.

Observando los valores de precisión (PPV) de la Tabla **10-1**, comprobamos que, tanto para clases desbalanceadas como para las balanceadas, los valores son realmente negativos, rondando todos el 4% y 5%. Este hecho lo podemos comprobar en las matrices de confusión de la **Tabla 10-2**, donde se aprecia que de 140 casos reales

donde los accidentes son graves, lo modelos detecta muchos más; siendo el que menos el modelo con clases desbalanceadas con 1655 (88+1567).

10.1.2. Capacidad de clasificación.

La capacidad de clasificación de clases positivas (exhaustividad) se ve relativamente mejorada cuando se aplican técnicas de balanceo tal y como se puede comprobar en la tabla de resultados, llegando a obtener valores del 73 o 77%. En las matrices de confusión observamos cómo se pasa de clasificar correctamente 88 casos positivos a 103 o 108 aplicando técnicas de Upsampling o Downsampling. Este hecho que puede parecer positivo, realmente se produce en detrimento de la especificidad. Los modelos a los que se les ha aplicado técnicas de desbalanceo, han generado un enorme número de clases positivas tal y como hemos visto anteriormente (precisión muy baja). Este hecho da lugar a que haya más opciones de obtener verdaderos positivos, pero a su vez, se generan un gran número de falsos positivos (clases que realmente pertenecen a clases negativas y se clasifican como positivas), provocando unos valores de especificidad peores que trabajando con clases desbalanceadas.

Por otro lado, una forma de relacionar la capacidad de clasificación tanto de clases positivas como negativas es analizando las curvas ROC y obteniendo el área bajo su curva (AUC). Observamos que, en todos casos, su valor es muy similar ya que se encuentran entre el 71% y 75%. Se confirma pues que existe una relación entre la exhaustividad y la especificidad inversamente proporcional, ya que en cuanto mayor es una, menor es la otra. A continuación, se muestran las curvas ROC con sus valores AUC para cada uno de los cinco modelos de regresión logística creados.

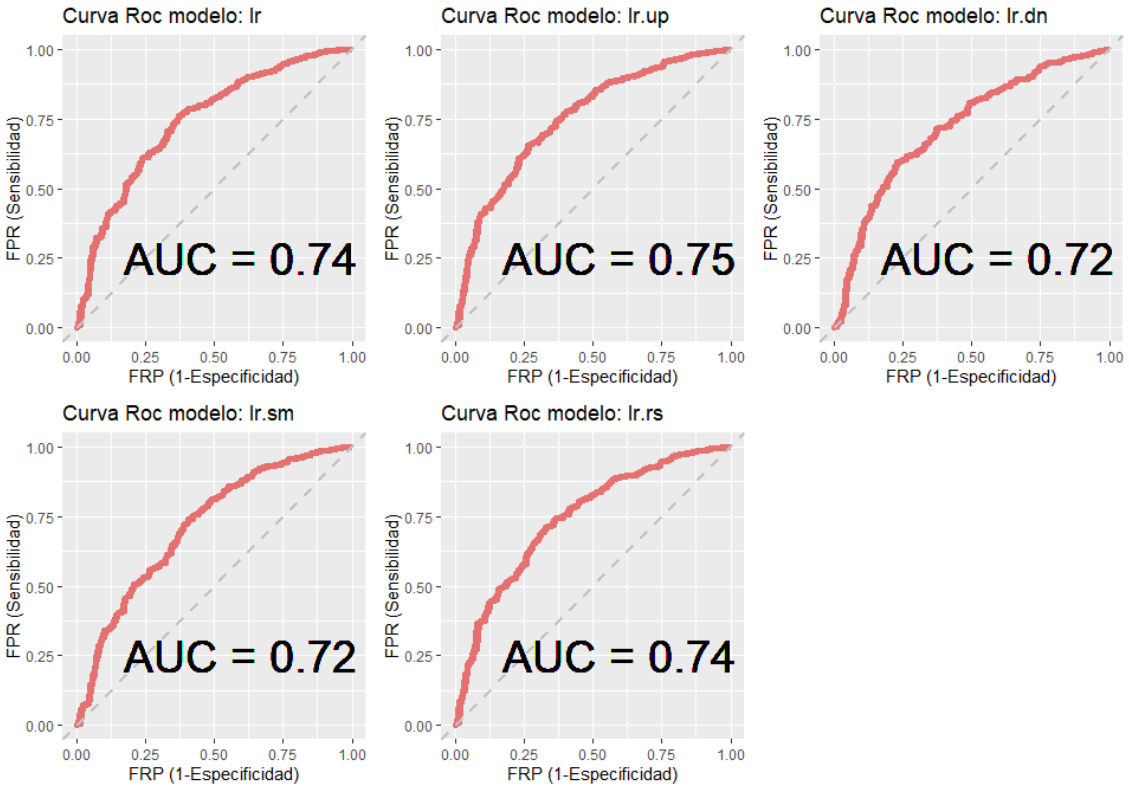

Figura 10-1. Curvas ROC de modelos de regresión logística.

Las curvas obtenidas son muy parecidas entre ellas, lo que imposibilita extraer ningún tipo de conclusión. En cuanto al área bajo la curva, los valores también son muy similares, pero distan ligeramente respecto el 0.8 definido en el capítulo **9**.

10.1.3. Capacidad de predecir y clasificar.

Finalmente, relacionamos la precisión y exhaustividad a partir del F1 score. Los resultados son muy negativos, llegando al 9.8% utilizando clases desbalanceadas. Hemos observado que el hecho de aplicar técnicas de balanceo de clases, provocan un aumento en cuanto a la exhaustividad, pero una disminución de la precisión. Esta relación, genera valores de F1 muy parecidos entre ellos, moviéndose entre el 7.4% y 8.8%.

10.2. Resultados random forest.

A continuación, se muestra la tabla resumen con los resultados de cada modelo:

	Accuracy	Sensitivity	Specificity	PPV	F1	AUC
Imbalanced	0.689	0.664	0.690	0.044	0.08193833	0.727
Upsampling	0.664	0.707	0.663	0.043	0.08071749	0.731
Downsampling	0.754	0.643	0.756	0.053	0.09825328	0.752
SMOTE	0.788	0.579	0.793	0.056	0.10246679	0.729
ROSE	0.664	0.664	0.664	0.040	0.07626076	0.724

Tabla 10-3. Métricas obtenidas para cada modelo random forest.

Para cada modelo se muestran las matrices de confusión:

Imbalanced

Prediction \ Reference	Grave	Leve
Grave	93	2037
Leve	47	4530

Upsampling

Prediction \ Reference	Grave	Leve
Grave	99	2214
Leve	41	4353

Downsampling

Prediction \ Reference	Grave	Leve
Grave	90	1602
Leve	50	4965

SMOTE

Prediction \ Reference	Grave	Leve
Grave	81	1360
Leve	59	5207

ROSE

Prediction \ Reference	Grave	Leve
Grave	93	2206
Leve	47	4361

Tabla 10-4. Matrices de confusión para cada modelo random forest.

10.2.1. Capacidad de predicción.

Como se puede comprobar en la **Tabla 10-3**, la capacidad de predicción de los modelos generados (PPV) es realmente baja, obteniendo valores muy similares tanto en clases desbalanceadas, como aplicando técnicas de balanceo. Observando las matrices de confusión de la **Tabla 10-4**, se pueden confirmar los valores obtenidos, ya que los modelos generan un número de observaciones positivas mucho más elevado respecto

las que realmente hay. Por ejemplo, en la técnica de Upsampling, de los 140 heridos graves que realmente hay en el conjunto de test, el modelo genera 2313 (99+2214) casos.

10.2.2. Capacidad de clasificación.

La capacidad de clasificación de los modelos generados (exhaustividad y especificidad) es bastante similar tanto en el modelo con clases desbalanceadas como en el resto de modelos. Este hecho es especialmente significativo, ya que, en teoría, al aplicar técnicas de balanceo, se debería obtener una mejora relativa en la exhaustividad y, por el contrario, un empeoramiento de la especificidad. En cuanto a la exhaustividad, como mucho, se ve mejorada un 4.3% si se aplica la técnica de Upsampling, mientras que la especificidad, lejos de empeorar, se ve mejorada hasta un 10.3% cuando aplicamos SMOTE.

La relación entre exhaustividad y especificidad la podemos visualizar mediante la curva ROC y su valor asociado del área bajo la curva (AUC). Debido a las pocas variaciones entre exhaustividad y especificidad, los valores de AUC son todos muy similares, rondando todos el 73% aproximadamente. A continuación, se muestran las curvas ROC para cada uno de los cinco modelos random forest creados.

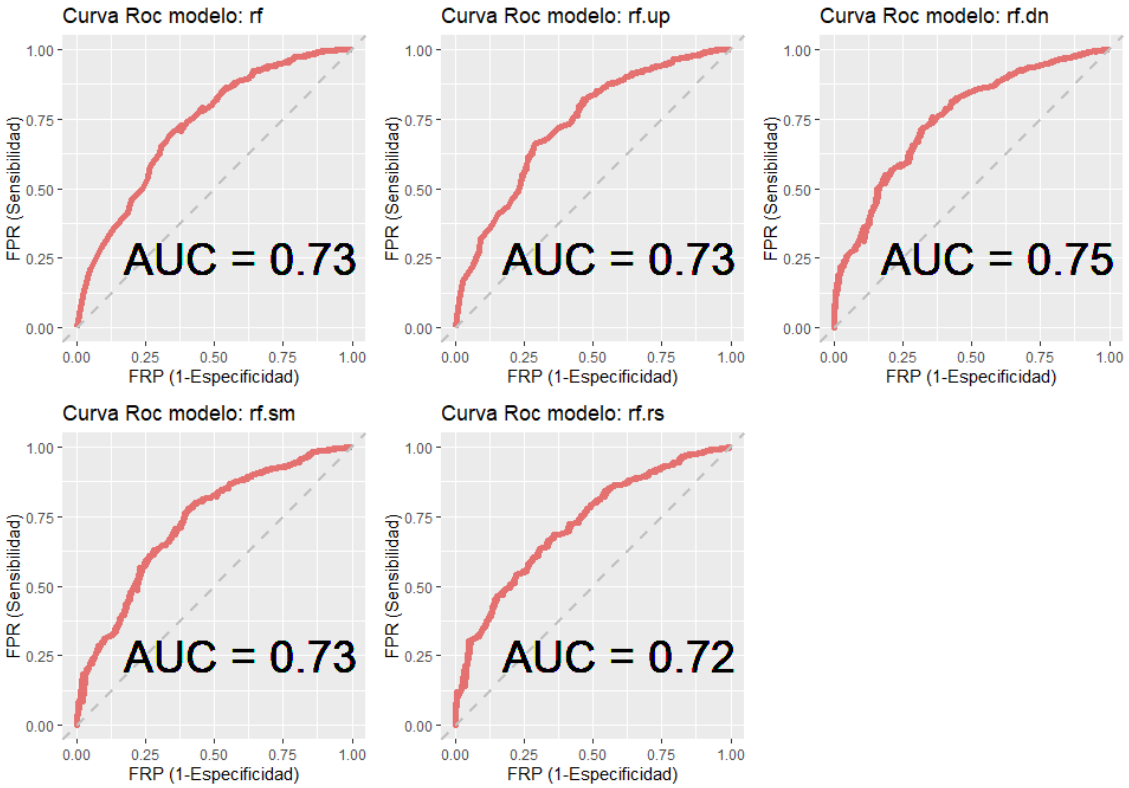

Figura 10-2. Curvas ROC de modelos random forest.

10.2.3. Capacidad de predicción y clasificación.

La métrica que relaciona la capacidad de predicción y la de clasificación, F1 score, retorna valores muy bajos (entre el 8% y 9%) como es de esperar debido a los valores de precisión tan bajos. Cabe destacar el F1 score aplicando técnicas SMOTE. Este obtiene un valor del 10% lo cual puede indicar que es el mejor modelo. Desgraciadamente, si nos fijamos en la exhaustividad (parámetro

más importante en la evaluación de estos modelos) es únicamente del 56%. Esto es debido a que la precisión es la más elevada de todas con un 5.6%. Así pues, observamos que el F1 score nos puede extraer conclusiones erróneas.

10.3. Comparativa de modelos.

Con la implementación de los modelos de regresión logística y random forest, se pretendía comprobar qué diferencias; y a su vez mejoras, podían aportar cada uno de ellos en el caso de estudio. Además, debido a la dificultad extra de trabajar con clases desbalanceadas, se ha querido mejorar los resultados aplicando diferentes técnicas de balanceo. Curiosamente, en ambos casos, y utilizando las cuatro técnicas, los resultados obtenidos son muy similares, siendo posiblemente los de regresión logística, los que ofrecen una mínima mejora. Pese a esta afirmación, cabe destacar la complejidad enorme de comparar los modelos debido a la aparente aleatoriedad de sus resultados. Cuando un modelo obtiene buenos resultados en una medida, hay otra que lo empeora; por ejemplo, una buena exhaustividad implica una mala especificidad en la mayoría de los casos.

Se puede comprobar que la precisión es realmente baja en todos los modelos, rondando valores entre el 4% y 5%, lo cual nos indica claramente que predice erróneamente las clases positivas. Comprobando detenidamente las matrices de confusión, se observa que los modelos predicen una cantidad mucho más elevada de heridos graves de los que en realidad hay. Este hecho confirma que las técnicas aplicadas no han sido capaces de solucionar el problema de solape entre clases (*class separability problema*) explicado en el apartado **6.2.Desbalanceo de clases** y, por lo tanto, se procede a evaluar otras medidas.

La exhaustividad mejora considerablemente respecto la precisión, obteniendo valores entre el 57.9% y el 77%. De todas maneras, altos valores de exhaustividad implican valores de especificidad malos, por lo que hay un compromiso entre la predicción de clases positivas y clases negativas; siendo imposible obtener buenos resultados en ambas clases. Comparemos la exhaustividad y especificidad para cada uno de los casos:

Modelos	**Detección heridos graves (exhaustividad)**	**Derección heridos leves (especificidad)**
RL: clases desbal.	62.9%	76.1%
RL: Upsampling	**73.6%**	**65.6%**
RL: Downsampling	**77.1%**	**59.4%**
RL: SMOTE	61.4%	72.1%
RL: ROSE	67.1%	71.2%
RF: clases desbal.	66.4%	69.0%
RF: Upsampling	**70.7%**	**66.3%**
RF: Downsampling	64.3%	75.6%
RF: SMOTE	57.9%	79.3%
RF: ROSE	66.4%	66.4%

Tabla 10-5. Valores de exhaustividad y especificidad de los modelos generados.

En la **Tabla 10-5** comprobamos la relación inversa entre exhaustividad y especificidad; y se aprecia la complejidad en la elección del mejor modelo debido a la similitud de sus

resultados. En verde se han marcado los modelos que tienen una exhaustividad superior al 70% ya que como se ha explicado, es primordial la correcta detección o clasificación de las clases positivas `Grave`. Curiosamente, las técnicas de balanceo que consiguen mejores resultados son las más simples: Upsampling y Downsampling; aunque también se debe tener presente que estas técnicas aumentan o disminuyen de forma aleatoria las observaciones, por lo que posiblemente, si se volviese a emplear esta técnica sobre el mismo conjunto de datos de entrenamiento, los resultados obtenidos variarían.

El modelo de regresión logística aplicando la técnica Downsampling, es el que ofrece la mayor exhaustividad (77.1%), pero a su vez es el modelo con peor especificidad. Así pues, de los dos modelos restantes resaltados en la Tabla **10-5**, se ha intentado buscar un cierto equilibrio entre la detección de heridos graves y leves, concluyendo que el que mejores resultados ofrece es el modelo de regresión logística aplicando la técnica de Upsampling sobre sus datos de entrenamiento.

11. Conclusiones

En este último capítulo del trabajo se procede a reflexionar tanto los resultados obtenidos como el desarrollo del proyecto, enfatizando esos puntos donde más dificultades se han encontrado.

Finalmente, una vez descrito el producto final, se realizará un análisis del porqué de la calidad obtenida, planteando posibles soluciones en caso de continuar el trabajo en un futuro.

11.1. Planificación y problemas encontrados.

Para la realización de este Trabajo se ha intentado seguir la planificación definida en los inicios del proyecto. Se intentó ajustar al máximo a los tiempos de cada PAC y teniendo en cuenta la duración total de cuatro meses aproximadamente.

El grosor del proyecto se centra en la PAC3 de diseño e implementación donde se decidió darle una duración de 2 meses, destinándole únicamente 8 días al proceso de obtención y unión de los datos. Precisamente, en esta temprana fase fue donde más problemas tuve; concretamente con la unión de los datos. Como se ha explicado en el capítulo **4**, el proceso de unificación de ha sido realmente complejo debido a la falta de información en los datos. Este aspecto ha supuesto que el tiempo para llevar a cabo esta función haya sido prácticamente de un mes en vez de los 8 días previstos inicialmente. Para poder recuperar todo el tiempo perdido y ajustarme a la entrega de la PAC3, se han destinado muchas más horas diarias de las previstas inicialmente, y a su vez el proceso de limpieza de datos se ha unido con el de unificación.

Por otro lado, el desbalanceo de clases ha supuesto un descubrimiento durante el análisis de los datos nada deseado. Este aspecto ha implicado que se debiera buscar más información sobre los efectos que provoca durante el modelado y qué técnicas existen para mitigar el problema. Afortunadamente, existe una amplia bibliografía sobre clases desbalanceadas y, pese a planificar esta fase con 7 días, sólo se han destinado pocos días más.

11.2. Productos obtenidos.

El producto que se pretendía obtener al finalizar este proyecto correspondía a un modelo que permitiese predecir la gravedad de los heridos en los accidentes de tráfico en la ciudad de Barcelona. Esta información sería de gran utilidad para los servicios de emergencia ya que el dispositivo a desplegar podría ser más preciso.

Para ello se ha hecho uso de un gran número de observaciones correspondientes a los accidentes de tráfico sucedidos en la ciudad en los años 2017 y 2018. Además, se ha añadido otra información de interés relacionada con las festividades de la ciudad y el estado meteorológico.

A medida que se ha ido avanzando, se ha comprobado como las clases de la variable dependiente estaban claramente desbalanceadas, teniendo que ser tratadas mediante diferentes técnicas de balanceo para poder obtener modelos más precisos. Este aspecto

ha comportado que se generasen un total de diez modelos, con lo que la elección de mejor modelo no se limitaba solo a la tipología de regresión logística o random forest; ahora también se debían considerar estos mismos modelos con diferentes técnicas de balanceo.

Para evaluarlos, se han planteado cinco métricas y la curva ROC, y se ha definido un procedimiento de evaluación en el que se consideraban los siguientes conceptos: capacidad de predicción (precisión) de los modelos y capacidad de clasificación (exhaustividad y especificidad) de los modelos. Se le ha dado una importancia mayor a la capacidad de clasificar clases positivas, ya que los heridos graves son los más sensibles y el error en su predicción debe ser mínimo.

Los resultados obtenidos en ambos modelos son muy similares, siendo los de regresión logística los que mejor pueden predecir los heridos graves (mejor exhaustividad). El primer aspecto que llama la atención es la gran cantidad de predicciones graves respecto las originales del conjunto de test. De los 140 heridos graves reales existentes, hay modelos que llegan a predecir hasta 2772, acertando únicamente 108. Esta baja precisión (alrededor del 4%) indica que los modelos, pese a aplicar técnicas de balanceo, no son capaces de separar correctamente las dos clases y se produce un solape (*class separability problema*). Si se extrapolasen estos datos a la realidad, se hubiesen detectado 2772 heridos graves cuando en realidad hay 140, desperdiciando recursos hasta en 2664 casos.
De todas maneras, la capacidad de clasificación de las clases positivas no es del todo mala, llegando a obtener valores del 77.1%. Dado que este es el parámetro más importante en la evaluación, se han seleccionado aquellos modelos con una exhaustividad superior al 70% y se ha buscado un equilibrio entre exhaustividad y especificidad. Finalmente se ha declinado por el modelo de regresión logística aplicando técnicas Upsampling en su conjunto de datos de entrenamiento:

```
##              Accuracy Sensitivity Specificity   PPV         F1   AUC
## Upsampling      0.658       0.736       0.656 0.044 0.08243297 0.752
```

Tabla 11-1. Resultados del modelo de regresión logística aplicando técnicas Upsampling.

Con los resultados obtenidos se puede extraer la siguiente conclusión: Lo modelos tienen una muy mala capacidad de predicción, pero en cambio tienen una decente capacidad de clasificación. Aplicando cualquiera de estos modelos, se estarían desperdiciando recursos ya que se predicen muchos más heridos graves de los que hay realmente (precisión muy baja), pero se estarían detectando los heridos graves con relativa eficacia (buena exhaustividad).

11.3. Mejoras sobre el producto y trabajo futuro.

La poca variación entre los diez modelos hace pensar que pude existir un problema con los datos trabajados. Descartado el factor de aleatoriedad en los accidentes de tráfico descrito por J.S.Baker **[1]** en 1970, se debe mejorar la obtención de variables que permitan predecir los accidentes. Para este proyecto se han empleado datos muy genéricos y se han introducido errores como los siguientes:

- Por falta de información, se han tratado los datos meteorológicos como medias diarias en vez de tratarlos por minutos o horas.

- Se han reducido los niveles de varias variables tipo factor, reduciendo enormemente la capacidad de discernir entre una clase u otra, pero mejorando los grados de libertad en los modelos creados.
- Al tener datos de entrada desagregados, se han tenido que ajustar variables como las causas de los accidentes o sus tipologías, siendo estas últimas donde se ha introducido mayor error al reducir varias tipologías a una. Además, debido a la nula capacidad de relacionar estas variables con cada uno de los heridos, se han tenido que eliminar varias observaciones.

Estos datos podrían mejorarse poniéndose en contacto con la Guardia Urbana de Barcelona ya que disponen de información más precisa de todos los accidentes. En la web Open Data, al tratarse de un portal web donde se muestra información de forma abierta, se publican datos no confidenciales, lo cual hace muy complicado ubicar a cada herido en su vehículo y tipología de accidente sufrida.

Además, como se ha explicado en el capítulo **1**, los proyectos actualmente en uso como Waycare o Crash Risk Map, trabajan con variables mucho más precisas, como son: la proximidad de vallas comerciales respecto el punto donde se ha producido el accidente de tráfico, el estado del tráfico, la posición del sol, etc. Para obtener este grado de precisión en las variables, es necesario un trabajo de campo que en este Trabajo no ha sido realizado; y, por lo tanto, un tiempo para la extracción de datos mucho más amplio.

Otro aspecto que puede explicar los resultados obtenidos es la dimensionalidad del problema. Recordemos que en el capítulo **5**, cuando se han estudiado los datos geolocalizados, se ha llegado a la conclusión que un estudio a nivel de calles, puede explicar mejor la gravedad de los accidentes que trabajando a nivel de distrito; como se ha hecho en este Trabajo.

Con todas estas reflexiones podemos vislumbrar cuáles podrían ser las vías que explotar en trabajos futuros.

1- Evaluar otros modelos. Debido a que los resultados en los diez modelos son muy parecidos, pueden crearse otro tipo de modelos, también empleados en otros trabajos explicados en el capítulo **2**, como son redes neuronales o redes bayesianas.
   También, se debería realizar un trabajo de recerca para saber qué modelos son los más adecuados para trabajar con clases desbalanceadas.
2- Trabajar a nivel de calles. Una vez visto que trabajar a nivel de calles puede aportar mejores resultados sobre la gravedad que trabajando a nivel de distrito, se podrían explotar dos vías:
   - Trabajar con los datos de una sola calle o vía tal y como hace Luis Cruz Bellas en su Trabajo final de Máster **[4]** donde evalúa la autovía M-30 por quilómetros. En nuestro caso se podría estudiar cualquiera de las dos rondas, o la calle con más número de heridos.
   - Trabajar con subconjuntos de calles. Se dividirían los datos de entrada en subconjuntos de calles. Dado que hay 1401 calles en el conjunto de datos, se podría empezar evaluando subconjuntos de las 10 calles con mayor número de heridos.

3- Mejorar la calidad de los datos. Con los datos de la web Open Data de Barcelona, se debería poner en contacto con el ayuntamiento o Guardia Urbana para acceder a información más precisa que permita unificar correctamente los datos y no tener que aplicar procesos de reducción como los utilizados en el Trabajo actual.

   Por otro lado, se podrían extraer valores meteorológicos por horas o minutos, en vez de trabar con medias diarias como se ha hecho.

4- Obtención de variables más precisas. Este punto requiere un trabajo previo muy minucioso para intentar extraer factores, objetos, etc. que puedan aumentar las probabilidades de accidentes. Por ejemplo: vallas publicitarias, señales viarias, densidades de tráfico, etc.

Bibliografía

[1] Baker, J.S. (1970). *Manual de Investigación de Accidentes de Tráfico*.

[2] Úbeda González, David (2017). *Predicción de la severidad de accidentes de tráfico en la Red de Carreteras de España y Reino Unido mediante modelos estadísticos basados en Random Forest y Regresión Logística*. Tesis Doctoral, Universidad Miguel Hernández

[3] Montesinos Muñoz, Joaquín (2018). *Estudio sobre patrones de accidentes en la ciudad de València*. Trabajo final de Máster, Universidad Politécnica de Valencia, España.

[4] Cruz Bellas, Luis. (2017). *Modelos predictivos de accidentes de tráfico en Madrid*. Trabajo final de Máster, Universidad Internacional de la Rioja, España.

[5] Arenas, B., Aparicio, F., González, C. y Gómez, A. (20009) *The influence of heavy goods vehicle traffic on accidents on different types of Spanish interurban roads*. Accident Analysis and Prevention, 41 (1). 15-24 ISSN 0001-4575.

[6] Cintas del Río, R. y Brita-Paja Segoviano, J.L. (2015). *Metodología de Minería de datos para el estudio de tablas de siniestralidad.* Trabajo final de Máster, Universidad Complutense de Madrid, España.

[7] Waycare (2018). *Waycare and Nevada Transportation Agencies Partner to Dynamically Identify Roads at High Risk for Accidents, Resulting in 17% Reduction in Crashes Along I-15 in Las Vegas*. https://waycaretech.com/publication/waycare-reduced-primary-crashes-by-17-percent/

[8] ISP. (2016). *Daily Crash Prediction*. https://www.in.gov/isp/ispCrashApp/main.html

[9] Márquez Daniel, Carlos. (15/02/2019). *El 'Minority Report' de los accidentes*. elPeriódico. https://www.elperiodico.com/es/barcelona/20190215/algoritmo-prediccion-accidentes-trafico-barcelona-7305973

[10] ITEUVE. *Clasificación de vehículos*. https://www.iteuve.net/clasificacion-vehiculos

[11] Meteolobios. (2013). *Lluvia*. http://www.meteolobios.es/lluvia.htm

[12] SMN (Servicio Meteorológico Nacional). *¿Cómo clasificamos la intensidad del viento?* https://www.smn.gob.ar/noticias/%C2%BFc%C3%B3mo-clasificamos-la-intensidad-del-viento?

[13] López, V., Fernández, A., García, S., Palade, V. y Herrera, F. (2013). *An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*. Information Sciences 250 (2013) 113-141.

[14] Khun, M. (2019). *Classification and Regression Training. Package 'caret'*. https://cran.r-project.org/web/packages/caret/caret.pdf

[15] Chawla, N., Bowyer K., Hall, L. y Kegelmeyer W. (2002). *SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research* 16 (2002) 321–357.

[16] Torgo, L. (2015). *Functions and data for "Data Mining with R". Package 'DMwR'*. https://cran.r-project.org/web/packages/DMwR/DMwR.pdf

[17] Menardi, G. y Torelli N. (2010). *Training and assessing classification rules with unbalanced data*. Working Paper Series, N. 2, 2010.

[18] Lunardon, N., Menardi, G. y Torelli N. (2015). *ROSE: Random Over-Sampling Examples. Package 'ROSE'*. **https://cran.r-project.org/web/packages/ROSE/ROSE.pdf**

[19] R-core R-core@R-project.org. *Fitting Generalized Linear Models*. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html

[20] Documentation for package 'stats' version 4.0.0. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html

[21] Breiman, Leo (2001). *Machine Learning - Random Forests*. 45 (1): 5–32. doi:10.1023/A:1010933404324

[22] Breiman, Leo y Cutler, Adele (2018). *Breiman and Cutler's Random Forests for Classification and Regression*, version 4.6-14.
https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[23] Powers, David M W (2011). *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*. Journal of Machine Learning Technologies. 2 (1): 37–63.

[24] Fawcett, T. (2006). *An Introduction to ROC Analysis*. Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010

Anexo: Códigos

A. Código 1: Unificación de datos.

```
# Librerías necesarias
library(jsonlite)

## 1.1. CARGA DE DATOS Y SELECCIÓN DE VARIABLES ######################
#####################################################################################
# Carga de datos
persona_17 <- read.csv("../data/2017_accidents_persones_gu_bcn_.csv",
                       fileEncoding="UTF-8", header = TRUE)
tipo_17 <- read.csv("../data/2017_accidents_tipus_gu_bcn_.csv",
                       fileEncoding="UTF-8", header = TRUE)
causa_17<- read.csv("../data/2017_accidents_causes_gu_bcn_.csv",
                       fileEncoding="UTF-8", header = TRUE)
general_17 <- read.csv("../data/2017_accidents_gu_bcn.csv",
                       fileEncoding="UTF-8", header = TRUE)
persona_18 <- read.csv("../data/2018_accidents_persones_gu_bcn_.csv",
                       fileEncoding="UTF-8", header = TRUE)
tipo_18 <- read.csv("../data/2018_accidents_tipus_gu_bcn_.csv",
                       fileEncoding="UTF-8", header = TRUE)
causa_18<- read.csv("../data/2018_accidents_causes_gu_bcn_.csv",
                       fileEncoding="UTF-8", header = TRUE)
general_18 <- read.csv("../data/2018_accidents_gu_bcn.csv",
                       fileEncoding="UTF-8", header = TRUE)

meteo <- fromJSON("../data/meteo_2017_2018_bcn")

cal <- read.csv("../data/calendario_festivos_17_18.csv",
                header = TRUE, sep=";")

# Selección de variables
persona_17 <- persona_17[, c(1, 3, 10, 12, 13, 15, 17, 18, 19, 20, 21, 22, 24,
                             28, 27)]
persona_18 <- persona_18[, c(1, 3, 10, 12, 13, 15, 17, 18, 19, 20, 21, 22, 24,
                             28, 27)]
tipo_17 <-tipo_17[,c(1,18)]
tipo_18 <-tipo_18[,c(1,18)]
causa_17 <-causa_17[,c(1,18)]
causa_18 <-causa_18[,c(1,18)]
general_17 <- general_17[,c(1,23)]
general_18 <- general_18[,c(1,23)]

meteo <- meteo[, c(1,6,7,13)]

## 1.2. UNIÓN DATASETS: AÑO 2017 + 2018 ###################################
#####################################################################################
# Igualar nombre variables
var_per <- c("id", "distrito", "dia_nombre", "año", "mes", "dia",
             "hora", "causa_peaton", "vehiculo", "sexo", "edad",
             "tipo_persona", "gravedad","Lat", "Long")
var_tipo <- c("id", "tipo")
var_causa <- c("id", "causa")
var_gen <- c("id", "n_vehic")
```

```
colnames(persona_17) <- var_per
colnames(persona_18) <- var_per
colnames(tipo_17) <- var_tipo
colnames(tipo_18) <- var_tipo
colnames(causa_17) <- var_causa
colnames(causa_18) <- var_causa
colnames(general_17) <- var_gen
colnames(general_18) <- var_gen

# Unificación de tipologías de variables
persona_17$Lat <- as.numeric(as.character(persona_17$Lat))
persona_17$Long <- as.numeric(as.character(persona_17$Long))

# Unión xxxx_17 <-> xxxx_18
persona <- rbind(persona_17, persona_18)
tipo <- rbind(tipo_17, tipo_18)
causa <- rbind(causa_17, causa_18)
general <- rbind(general_17, general_18)

# Se eliminan duplicados
persona <- unique(persona)
tipo <- unique(tipo)
causa <- unique(causa)
general <- unique(general)

# Elimina vehiculos de movilidad reducida
persona <- persona[!(persona$vehiculo == "Veh. mobilitat personal sense motor"
                   | persona$vehiculo == "Veh. mobilitat personal amb motor")
           ,]

## 1.3. ADPTACIÓN Y CREACIÓN DE VARIABLES RELACIONALES ENTRA DF ##############
################################################################################
# Elimina espacios variable "id"
persona$id <- factor(as.character(
                     lapply(persona$id,function(x){gsub(" ","",x)})))
causa$id <- factor(as.character(
                     lapply(causa$id,function(x){gsub(" ","",x)})))
tipo$id <- factor(as.character(
                     lapply(tipo$id,function(x){gsub(" ","",x)})))
general$id <- factor(as.character(
                     lapply(general$id,function(x){gsub(" ","",x)})))

# Creación campo fecha
persona$mes <- formatC(persona$mes ,width=2,flag="0")
persona$dia <- formatC(persona$dia ,width=2,flag="0")
persona$fecha <- paste(persona$año, "-", persona$mes, "-", persona$dia,
sep="")

## 1.4. UNIFICACIÓN DE TIPOLOGÍAS ##############################################
################################################################################
data <- persona

data$tipo <- NA
ids = unique(data$id)

tipologias <- c("Abast", "Encalç", "Abast multiple", "Col.lisió lateral",
                "Col.lisió fronto-lateral", "Xoc contra element estàtic",
                "Col.lisió frontal", "Sortida de via amb bolcada",
```

```
                "Sortida de via amb xoc o col.lisió",
                "Xoc amb animal a la calçada", "Resta sortides de via",
                "Bolcada (més de dues rodes)", "Atropellament",
                "Caiguda (dues rodes)", "Caiguda interior vehicle",
                "Altres", "Desconegut")

for (x in 1: length(ids)){

  subp <- data[as.character(data$id) == ids[x],]
  subt <- tipo[as.character(tipo$id) == ids[x],]
  subg <- general[as.character(general$id) == ids[x],]

  heridos <- nrow(subp)
  tipos <- nrow(subt)
  vehiculos <- subg$n_vehic

  rol<-table(subp$tipo_persona)
  # Sólo una tipología -> todos comparten tipología
  if(tipos == 1){
    data[data$id == ids[x],]$tipo <- as.character(subt$tipo)
  } else{
  # Sólo un herido o sólo un conductor, se elige la tipología más restrictiva
    if(heridos == 1 || (rol[1] == 1 && vehiculos == 1)){
      i <- na.omit(match(subt$tipo, tipologias))
      tipo_reducida <- subt$tipo[which(i == min(i))]
      data[data$id == ids[x],]$tipo <- as.character(tipo_reducida)
    }
  }
}
# Si la víctima es un peatón, únicamente ha podido ser atropellado
data[data$tipo_persona == "Vianant",]$tipo <- "Atropellament"

## 1.5. UNIFICACIÓN DE CAUSAS ##################################################
###############################################################################
# Elimina doble causas: Exceso velocidad + Alcoholemia = Alcoholemia
ids = unique(causa$id)

for(x in 1:length(ids)){
  subc <- causa[as.character(causa$id) == ids[x],]
  if( nrow(subc) == 2){
    causa <- causa[!(causa$id == ids[x] & causa$causa == "Excés de velocitat o
                                                          inadequada"),]
  }
}

# Unión
data <- merge(data, causa, by = "id")

## 1.6. UNIFICACIÓN DE FESTIVOS Y METEOROLOGÍA ##################################
###############################################################################
data <- merge(data, cal, by ="fecha", all.x = TRUE)
data <- merge(data, meteo, by = "fecha", all.x = TRUE)

## 1.7. ESCRITURA ##############################################################
###############################################################################
write.csv(data, file = "data_unificada.csv")
```

B. Código 2: Procesado de los datos.

```
# Librerías necesarias
library(dplyr)
library(mgsub)
library(ggplot2)

# Datos código anterior
data <- read.csv("data_unificada.csv", row.names = 1)


## 2.1. CONVERSIÓN DE VARIABLES ###############################################
##############################################################################

# - Variables meteorológicas: factor -> numeric o integer
data$tmed <- gsub(',', '.', data$tmed)
data$prec <- gsub(',', '.', data$prec)
data$velmedia <- gsub(',', '.', data$velmedia)
# Ip = 0 mm
data$prec <- gsub('Ip', '0', data$prec)

data$tmed <- as.numeric(data$tmed)
data$prec <- as.numeric(data$prec)
data$velmedia <- as.numeric(data$velmedia)

# - Variable edad: factor -> integer
data$edad <- as.integer(data$edad)

# - Arreglo valores mal convertidos de las variables Lon y Lat
for (i in 1:length(data$Lat)){
  if(data$Lat[i]>43 || is.na(data$Lat[i])){data$Lat[i] <- data$Lat[i]/1000 }
  if(data$Long[i]>3 || is.na(data$Long[i])){data$Long[i] <- data$Long[i]/1000
}
}

## 2.2. REDUCCIÓN DE NIVELES ##################################################
##############################################################################

# - Variable vehículo
Tipo1 <- c("Bicicleta", "Ciclomotor", "Cuadriciclo <75cc", "Motocicleta",
           "Quadricicle > 75 cc", "Quadricicle < 75 cc")
Tipo2 <- c("Taxi", "Todo terreno", "Tot terreny", "Turisme", "Turismo")
Tipo3 <- c("Camió rígid <= 3,5 tones", "Camión <= 3,5 Tm", "Furgoneta",
           "Microbús <= 17", "Microbus <= 17", "Autocaravana")
Tipo4 <- c("Autobús articulado", "Autobús articulat", "Autobús", "Autocar",
           "Maquinària d'obres i serveis", "Tractocamión", "Tractor camió",
           "Camión > 3,5 Tm", "Camió rígid > 3,5 tones")
Desc <- c("Tranvía o tren", "Tren o tramvia", "Otros vehíc. a motor",
          "Desconegut", "Altres vehicles amb motor",
          "Altres vehicles sense motor")

data <- mutate(data,
               vehiculo = ifelse(vehiculo %in% Tipo1, "Tipo1",
                          ifelse(vehiculo %in% Tipo2, "Tipo2",
                          ifelse(vehiculo %in% Tipo3, "Tipo3",
                          ifelse(vehiculo %in% Tipo4, "Tipo4", NA)))))
data$vehiculo <- factor(data$vehiculo, ordered = FALSE)
```

```
# - Variable tipo
Alcance <- c("Abast", "Abast multiple", "Encalç")
Salida <- c("Sortida de via amb bolcada",
            "Sortida de via amb xoc o col.lisió")
Choque <- c("Xoc amb animal a la calçada", "Xoc contra element estàtic")
Atropello <- c("Atropellament")
Vuelco <- c("Bolcada (més de dues rodes)")
Caida <- c("Caiguda (dues rodes)")
Caida_interior <- c("Caiguda interior vehicle")
Colision <- c("Col.lisió frontal", "Col.lisió fronto-lateral",
              "Col.lisió lateral")
Desc <- c("Desconegut", "0", "Altres")

data <- mutate(data,
               tipo = ifelse(tipo %in% Alcance, "Alcance",
                      ifelse(tipo %in% Salida, "Salida",
                      ifelse(tipo %in% Choque, "Choque",
                      ifelse(tipo %in% Atropello, "Atropello",
                      ifelse(tipo %in% Vuelco, "Vuelco",
                      ifelse(tipo %in% Caida, "Caída",
                      ifelse(tipo %in% Caida_interior, "Caída interior",
                      ifelse(tipo %in% Colision, "Colisión", NA)))))))))
data$tipo <- factor(data$tipo, ordered = FALSE)

# - Variable causa
Drogas <- c("Alcoholèmia", "Drogues o medicaments")
Estado <- c("Estat de la senyalització", "Calçada en mal estat")
Velocidad <- c("Excés de velocitat o inadequada")
Meteo <-c("Factors meteorològics")
Objetos <- c("Objectes o animals a la calçada")

data <- mutate(data,
               causa = ifelse(causa %in% Drogas, "Drogas",
                       ifelse(causa %in% Estado, "Mal estado elementos
                                                  viarios",
                       ifelse(causa %in% Velocidad, "Velocidad",
                       ifelse(causa %in% Meteo, "Meteorología",
                       ifelse(causa %in% Objetos, "Obstáculos", "Sin
causa"))))))
data$causa <- factor(data$causa, ordered = FALSE)


## 2.3. BINARIZACIÓN DE VARIABLES CATEGÓRICAS ##################################
###############################################################################
# - Variable causa_peaton
data$causa_peaton <- ifelse(data$causa_peaton == "No és causa del  vianant",
"No", "Si" )
data$causa_peaton <- factor(data$causa_peaton, ordered = FALSE)

# - Variable sexo
data$sexo <- ifelse(data$sexo == "Desconegut", NA, ifelse(data$sexo == "Home",
"Hombre", "Mujer"))
data$sexo <- factor(data$sexo, ordered = FALSE)

# - Variable gravedad
grave <- c("Ferit greu", "Ferit greu: hospitalització superior a 24h", "Mort",
           "Mort (dins 24h posteriors accident)")
data <- mutate(data,
               gravedad = ifelse(gravedad %in% grave, "Grave", "Leve"))
```

```
data$gravedad <- factor(data$gravedad, ordered = FALSE)


## 2.4. CREACIÓN DE NUEVAS VARIABLES ######################################
############################################################################
# - Variable lluvia
# Boxplot comprobación outliers
bp.prec <- ggplot(data = data, aes(x = factor(0), y = prec)) +
  geom_boxplot(fill = "#1F3552",alpha = 0.7) +
  stat_summary(fun.y = mean, geom = 'point', shape = 19, color = "red", cex =
               2) +
  scale_y_continuous(breaks = seq(0, 120, 10)) +
  scale_x_discrete(breaks = NULL) +
  xlab(label = "") +
  ylab(label = "Precipitaciones media (mm)")
bp.prec

data$lluvia[data$prec == 0] <- "Sin lluvia"
data$lluvia[data$prec > 0 & data$prec<= 2] <- "Débil"
data$lluvia[data$prec > 2 & data$prec<= 15] <- "Moderada"
data$lluvia[data$prec > 15 & data$prec<= 30] <- "Fuerte"
data$lluvia[data$prec > 30 & data$prec<= 60] <- "Muy fuerte"
data$lluvia[data$prec > 60] <- "Torrencial"

data$lluvia <- factor(data$lluvia, ordered = FALSE)

# - Variable viento

data$velmedia <- data$velmedia * 3.6

data$viento[data$velmedia >= 0 & data$velmedia < 6] <- "Calma"
data$viento[data$velmedia >= 6 & data$velmedia < 20] <- "Leve"
data$viento[data$velmedia >= 20 & data$velmedia < 29] <- "Moderado"
data$viento[data$velmedia >= 29 & data$velmedia < 39] <- "Regular"

data$viento <- factor(data$viento, ordered = FALSE)


## 2.5. TRATAMIENTO VALORES NA's ##########################################
############################################################################

# - Los NA en la variable festivo son laborables
data$festividad <- as.character(data$festividad)
data[is.na(data$festividad),]$festividad <- "Laboral"
data$festividad <- factor(data$festividad, ordered = FALSE)

# - Eliminación de TODOS NA's excepto Lat y Lon
data <- data[!is.na(data$sexo),]
data <- data[!is.na(data$vehiculo),]
data <- data[!is.na(data$tipo),]


## 2.6. TRADUCCIONES ######################################################
############################################################################

# - Variable dia_nombre
data$dia_nombre <- factor(mgsub(as.character(data$dia_nombre),
                                c("Dl", "Dm", "Dc", "Dj", "Dv", "Ds",
                                  "Dg"),
```

```
                                    c("Lunes", "Martes", "Miércoles", "Jueves",
                                      "Viernes", "Sábado", "Domingo")),
                              ordered = FALSE)


 # - Variable tipo_persona
data$tipo_persona <- factor(mgsub(as.character(data$tipo_persona),
                                    c("Conductor", "Passatger", "Vianant"),
                                    c("Conductor", "Pasajero", "Peatón")),
                              ordered = FALSE)


## 2.7. ELIMINACIÓN DE VARIABLES  ##########################################
###########################################################################
data <- data[,!(names(data) %in% c("id", "fecha", "año", "prec", "velmedia"))]


##  ESCRITURA  ############################################################
###########################################################################
data <- data[, c(1:10,12:19,11)]
write.csv(data, file = "data_clean.csv")
```

C. Código 3: Estudio de los datos.

```
library(ggplot2)
library(dplyr)
library(scales)
library(gridExtra)
data <- read.csv("data_clean.csv", row.names = 1)
# Fijo el orden de los días de la semana para una correcta representación

data$dia_nombre <- factor(data$dia_nombre, ordered = TRUE,
                          levels = c("Lunes", "Martes", "Miércoles", "Jueves",
                                     "Viernes", "Sábado", "Domingo"))
data$lluvia <- factor(data$lluvia, ordered = TRUE,
                      levels = c("Sin lluvia", "Débil", "Moderada", "Fuerte",
                                 "Muy fuerte", "Torrencial"))
## 3.1. EVALUACIÓN VARIABLE GRAVEDAD ######################################
###########################################################################
ggplot(data, aes(x = gravedad, fill = gravedad)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    theme(legend.position = "none",
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank())+
    scale_y_continuous(labels = percent) +
    scale_fill_manual(values=c("#E43B2D", "#67BD62")) +
    labs(y = "") +
    ggtitle("Gravedad del total de heridos")




## 3.2. EVALUACIÓN VARIABLES CATEGÓRICAS###################################
###########################################################################
```

```r
plot.cat <- function(variable){
  n.data <- data[,c(variable,"gravedad")]
  # General por variable
  var <- ggplot(n.data, aes(x = n.data[,1], fill = gravedad)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    theme(legend.position = c(0.75,1.07), legend.direction = "horizontal",
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank())+
    scale_y_continuous(labels = percent) +
    scale_fill_manual(values=c("#E43B2D", "#67BD62")) +
    labs(y = "") +
    ggtitle(variable)


# Separación de la variable en Grave y Leve
  dfg <- subset(n.data, n.data$gravedad == "Grave")
  dfl <- subset(n.data, n.data$gravedad == "Leve")

  bar.grave <- ggplot(dfg, aes(x = dfg[,1], fill = dfg[,1])) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    theme(legend.position = 'none',
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank()) +
    scale_y_continuous(labels = percent) +
    labs(y = "")+
    scale_fill_brewer(palette = "Set3")+
    ggtitle("Grave")

  bar.leve <- ggplot(dfl, aes(x = dfl[,1], fill = dfl[,1])) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    theme(legend.position = 'none',
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank()) +
    scale_y_continuous(labels = percent) +
    labs(x= "", y = "") +
    scale_fill_brewer(palette = "Set3")+
    ggtitle("Leve")


  m<-grid.arrange(bar.grave, bar.leve , ncol = 2)
  grid.arrange(var,m, ncol = 2)

}
plot.cat("distrito")
plot.cat("dia_nombre")
plot.cat("vehiculo")
plot.cat("tipo_persona")
plot.cat("tipo")
plot.cat("causa")
plot.cat("festividad")
plot.cat("lluvia")
plot.cat("viento")
plot.cat("causa_peaton")
plot.cat("sexo")


## 3.3. EVALUACIÓN VARIABLES NUMÉRICAS ######################################
################################################################################
plot.num <- function(variable){
```

```
  n.data <- data[,c(variable,"gravedad")]

  boxplot <- ggplot(n.data, aes(x = gravedad,
                                y = n.data[,1], fill = gravedad)) +
    geom_boxplot(alpha = 0.3) +
    theme(legend.position = c(0.93, 0.9))+
    stat_summary(fun.y = mean, geom = 'point', shape = 19,
                 color = "red",cex = 2) +
    labs(x = "Gravedad", y = "Edad") +
    labs(x= "", y = "") +
    ggtitle(paste("Boxplot variable", variable, sep=" "))


  densidad <- ggplot(n.data, aes(n.data[,1], fill = gravedad)) +
    geom_density(alpha = 0.2) +
    theme(legend.position = 'none')+
    labs(x= "", y = "") +
    scale_x_continuous(breaks = seq(0, max(n.data[,1]), 2))+

    ggtitle(paste("Densidad variable", variable, sep=" "))

  histograma <- ggplot(n.data, aes(x=n.data[,1])) +
    geom_histogram(binwidth=2,
                   color = "black", fill = "#A4DEC8", alpha = 0.3)+
    theme(legend.position = "none")+
    scale_x_continuous(breaks = seq(0, max(n.data[,1]), 2)) +
    labs(x= "", y = "") +
    ggtitle(paste("Histograma variable", variable, sep=" "))

  m<-grid.arrange(histograma, densidad , nrow = 2)
  grid.arrange(m, boxplot, ncol = 2)

}


plot.num("mes")
plot.num("dia")
plot.num("hora")
plot.num("tmed")
```

D. Código 4: Preparación de los datos.

```
library(caTools)
library(caret)
library(DMwR)
library(ROSE)
# Carga de los datos
data <- read.csv("data_clean.csv", row.names = 1)


## 4.1. LIMPIEZA PREVIA ###################################################
##########################################################################
# Se elimina:
    # Lat
    # Long
```

```
data <- data[,!(names(data) %in% c("Long", "Lat"))]
data <- data[sample(nrow(data)),]


## 4.2. SUBCONJUNTOS TRAIN & TEST ##############################################
################################################################################
indice <- sample.split(data$gravedad, SplitRatio = 0.70)
train <- subset(data, indice == TRUE)
test <- subset(data, indice == FALSE)


## 4.3. UPSAMPLING #############################################################
################################################################################
train.up <- upSample(x = train[, -ncol(train)], y = train$gravedad)
colnames(train.up)[ncol(train.up)] <- "gravedad"
summary(train.up$gravedad)
ggplot(train.up, aes(x = gravedad, fill = gravedad)) +
    geom_bar() +
    theme(legend.position = "none",
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank())+
    scale_y_continuous() +
    geom_text(mapping = aes(label =
                            scales::percent((..count..)/sum(..count..)),
                            y = (..count..)/sum(..count..) ),stat =
"count",vjust = -.5, size=5) +
    scale_fill_manual(values=c("#E43B2D", "#67BD62")) +
    labs(y = "") +
    ggtitle("Número de heridos aplicando Upsampling")

## 4.3. DOWNSAMPLING ###########################################################
################################################################################
train.dn <- downSample(x = train[, -ncol(train)], y = train$gravedad)
colnames(train.dn)[ncol(train.dn)] <- "gravedad"
summary(train.dn$gravedad)
ggplot(train.dn, aes(x = gravedad, fill = gravedad)) +
    geom_bar() +
    theme(legend.position = "none",
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank())+
    scale_y_continuous() +
    geom_text(mapping = aes(label =
                            scales::percent((..count..)/sum(..count..)),
                            y = (..count..)/sum(..count..) ),stat =
                                 "count",vjust = -.5, size=5) +
    scale_fill_manual(values=c("#E43B2D", "#67BD62")) +
    labs(y = "") +
    ggtitle("Número de heridos aplicando Downsampling")

## 4.4. SMOTE ##################################################################
################################################################################
train.sm <- SMOTE(gravedad ~., data = train, perc.over = 100, k =5)
summary(train.sm$gravedad)
ggplot(train.sm, aes(x = gravedad, fill = gravedad)) +
    geom_bar() +
    theme(legend.position = "none",
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank())+
    scale_y_continuous() +
```

```
    geom_text(mapping = aes(label =
              scales::percent((..count..)/sum(..count..)),
                              y = (..count..)/sum(..count..) ),stat =
                                   "count",vjust = -.5, size=5) +
    scale_fill_manual(values=c("#E43B2D", "#67BD62")) +
    labs(y = "") +
    ggtitle("Número de heridos aplicando SMOTE")

## 4.3. ROSE ##############################################################
##########################################################################
train.rs <- ROSE(gravedad ~ ., data  = train)$data
train.rs$gravedad <- relevel(train.rs$gravedad, "Grave")

summary(train.rs$gravedad)
ggplot(train.rs, aes(x = gravedad, fill = gravedad)) +
    geom_bar() +
    theme(legend.position = "none",
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title.x=element_blank())+
    scale_y_continuous() +
    geom_text(mapping = aes(label =
              scales::percent((..count..)/sum(..count..)),
                              y = (..count..)/sum(..count..) ),stat =
                            "count",vjust = -.5, size=5) +
    scale_fill_manual(values=c("#E43B2D", "#67BD62")) +
    labs(y = "") +
    ggtitle("Número de heridos aplicando ROSE")

## ESCRITURA ##############################################################
##########################################################################

write.csv(train, file = "train.csv")
write.csv(train.up, file = "train_up.csv")
write.csv(train.dn, file = "train_dn.csv")
write.csv(train.sm, file = "train_sm.csv")
write.csv(train.rs, file = "train_rs.csv")
write.csv(test, file = "test.csv")
```

E. Código 5: Modelado 1 - regresión logística.

```
library(gridExtra)

# Carga de datos
train <- read.csv("train.csv", row.names = 1)
test <- read.csv("test.csv", row.names = 1)

train.up <- read.csv("train_up.csv", row.names = 1)
train.dn <- read.csv("train_dn.csv", row.names = 1)
train.sm <- read.csv("train_sm.csv", row.names = 1)
train.rs <- read.csv("train_rs.csv", row.names = 1)

levels(train.dn$tipo) <- levels(test$tipo)
levels(train.dn$causa) <- levels(test$causa)


## MODELADOS RL ###########################################################
```

```
##############################################################################

# - Regresión logística con clases desbalanceadas.
lr <- glm(gravedad ~., data = train, family = binomial(link = "logit"))

# - Regresión logística aplicando Upsampling sobre train
lr.up <- glm(gravedad ~., data = train.up, family = binomial(link = "logit"))

# - Regresión logística aplicando Downsampling sobre train
lr.dn <- glm(gravedad ~., data = train.dn, family = binomial(link = "logit"))

# - Regresión logística aplicando SMOTE sobre train
lr.sm <- glm(gravedad ~., data = train.sm, family = binomial(link = "logit"))

# - Regresión logística aplicando ROSE sobre train
lr.rs <- glm(gravedad ~., data = train.rs, family = binomial(link = "logit"))
```

F. Código 6: Modelado 2 - random forest.

```
library(randomForest)
library(RColorBrewer)
library(gridExtra)
library(ggplot2)

# Carga de datos
train <- read.csv("train.csv", row.names = 1)
test <- read.csv("test.csv", row.names = 1)

train.up <- read.csv("train_up.csv", row.names = 1)
train.dn <- read.csv("train_dn.csv", row.names = 1)
train.sm <- read.csv("train_sm.csv", row.names = 1)
train.rs <- read.csv("train_rs.csv", row.names = 1)

levels(train.dn$tipo) <- levels(test$tipo)
levels(train.dn$causa) <- levels(test$causa)


## FUNCIÓN mejores.parametros() ##############################################
##############################################################################

mejores.parametrosRF <- function(train){
  trainX.rs <- train[, -ncol(train)]
  trainY.rs <- train$gravedad

  tune.all <- data.frame(matrix(vector(), 0, 3, dimnames=list(c(),
                  c("mtry", "OOBError","tree"))))
  # Se rellena el dataframe con los resultados por cada árbol
  for (tree in seq(200,1000,100)){
    tune <- tuneRF(trainX.rs, trainY.rs,
                   stepFactor = 1.5, improve = 1e-5, ntree = tree,
                   trace = FALSE, plot= FALSE)
    tune <- data.frame(tune)
    tune$tree <- tree

    tune.all <- rbind(tune.all,tune)
  }
```

```
  # Gráfico OOBError con todos los valores
  p <- ggplot(tune.all, aes(x = mtry, y = OOBError)) +
    geom_line(aes(colour = factor(tree)), size = 1) +
    geom_point(aes(colour = factor(tree)),size = 2) +
    theme(legend.title=element_blank()) +
    labs(title="'Out of bag error' por valor de 'mtry' y número de árboles")
  print(p)

  # Extracción de los mejores parámetros.
  mejor <- tune.all[tune.all$OOBError == min(tune.all$OOBError),]
  return(mejor)
}


## MODELOS RF ##################################################################
################################################################################

set.seed(1234)
# - Random Forest con clases desbalanceadas.
mejores.rf <- mejores.parametrosRF(train)
print(mejores.rf)
rf <- randomForest(gravedad~., data = train,
                   ntree = mejores.rf$tree[1],
                   mtry = mejores.rf$mtry)

# - Random Forest aplicando Upsampling sobre train
mejores.rf.up <- mejores.parametrosRF(train.up)
print(mejores.rf.up)
rf.up <- randomForest(gravedad~., data = train.up,
                      ntree = mejores.rf.up$tree[1],
                      mtry = mejores.rf.up$mtry[1])

# - Random Forest aplicando Downsampling sobre train
mejores.rf.dn <- mejores.parametrosRF(train.dn)
print(mejores.rf.dn)
rf.dn <- randomForest(gravedad~., data = train.dn,
                      ntree = mejores.rf.dn$tree[1],
                      mtry = mejores.rf.dn$mtry[1])

# - Random Forest aplicando SMOTE sobre train
mejores.rf.sm <- mejores.parametrosRF(train.sm)
print(mejores.rf.sm)
rf.sm <- randomForest(gravedad~., data = train.sm,
                      ntree = mejores.rf.sm$tree[1],
                      mtry = mejores.rf.sm$mtry[1])

# - Random Forest aplicando ROSE sobre train
mejores.rf.rs <- mejores.parametrosRF(train.rs)
print(mejores.rf.rs)
rf.rs <- randomForest(gravedad~., data = train.rs,
                      ntree = mejores.rf.rs$tree[1],
                      mtry = mejores.rf.rs$mtry[1])
```

G. Código 7: Evaluación.

```
library(pROC)
library(ggplot2)
library(caret)
library(gridExtra)

## FUNCIÓN evaluacion() ##################################################
#########################################################################

evaluacion <- function(modelo, test){

  # Extracción del tipo de modelo Reg.Log (lr) o RF (rf)
  nombre.m <- substitute(modelo)
  tipo <- substr(nombre.m,1,2)

  # Igualación de niveles
  variables <- names(test)
  for (x in 1:(length(variables)-1)){
    modelo$xlevels[[variables[x]]] <- union(modelo$xlevels[[variables[x]]],
                                            levels(test[[variables[x]]]))
  }


  # Extracción de probabilidades de predicciones
  if(tipo == "lr"){
    p.prob <- predict(modelo, test, type = 'response')
  }
  else{
    p.prob <- predict(modelo, test, type = "prob")[,2]
  }

  # Valor de corte para crear clases: corte_optimo
  analisis <- roc(response = test$gravedad,
                  predictor = p.prob, quiet = "TRUE")
  c <- cbind(analisis$thresholds,
             analisis$sensitivities + analisis$specificities)
  corte_optimo <- subset(c, c[,2] == max(c[, 2]))[,1]
  p.clase <- as.factor(ifelse(p.prob > corte_optimo, "Leve", "Grave"))

  # Generación curva ROC
  dROC <- data.frame(sn = analisis$sensitivities, sp = analisis$specificities)
  roc <- ggplot(dROC, aes(1 - sp, sn)) +
    geom_line(size = 2, alpha = 0.7, color = '#E13B3B')+
    labs(title= paste("Curva Roc modelo:", nombre.m),
         x = "FRP (1-Especificidad)", y = "FPR (Sensibilidad)") +
    geom_abline(size = 1, linetype = "dashed", slope=1,color = '#C2C2C2') +
    geom_text(x=0.6, y=0.25, label=paste("AUC =",round(analisis$auc,2)),
              size = 10)

  # Generación Matriz Confusión
  cm <- confusionMatrix(p.clase, test$gravedad, positive = "Grave")

  # Parámetros de retorno
  F1 <- 2*(cm$byClass[1]*cm$byClass[3])/(cm$byClass[1]+cm$byClass[3])
  parametros <- data.frame(Accuracy = round(cm$overall[1],3),
                           Sensitivity = round(cm$byClass[1],3),
                             Specificity = round(cm$byClass[2],3),
```

```
                              PPV = round(cm$byClass[3],3),
                              F1 = F1,
                              AUC = round(analisis$auc,3) )

  return(list(parametros, cm$table, roc))
}
## EVALUACIÓN REGRESIÓN LOGÍSTICA ##########################################
###########################################################################

# - Regresión logística con clases desbalanceadas.
resultado.lr <- evaluacion(lr, test)
# - Regresión logística aplicando Upsampling sobre train
resultado.lr.up <- evaluacion(lr.up, test)
# - Regresión logística aplicando Downsampling sobre train
resultado.lr.dn <- evaluacion(lr.dn, test)
# - Regresión logística aplicando SMOTE sobre train
resultado.lr.sm <- evaluacion(lr.sm, test)
# - Regresión logística aplicando ROSE sobre train
resultado.lr.rs <- evaluacion(lr.rs, test)

# - Resultados
tecnicas.par <- rbind(resultado.lr[[1]], resultado.lr.up[[1]],
resultado.lr.dn[[1]], resultado.lr.sm[[1]], resultado.lr.rs[[1]])
rownames(tecnicas.par)<- c("Imbalanced", "Upsampling", "Downsampling",
                           "SMOTE", "ROSE")
tecnicas.par

# - matrices confusión
resultado.lr.up[[2]]
resultado.lr.dn[[2]]
resultado.lr.sm[[2]]
resultado.lr.rs[[2]]

# - Curvas ROC técnicas ROC
grid.arrange(resultado.lr[[3]], resultado.lr.up[[3]], resultado.lr.dn[[3]],
             resultado.lr.sm[[3]], resultado.lr.rs[[3]],
             ncol = 3, nrow = 2)



## EVALUACIÓN RANDOM FOREST ################################################
###########################################################################
set.seed(4432)
# - Random Forest con clases desbalanceadas.
resultado.rf <- evaluacion(rf, test)
# - Random Forest aplicando Upsampling sobre train
resultado.rf.up <- evaluacion(rf.up, test)
# - Random Forest aplicando Downsampling sobre train
resultado.rf.dn <- evaluacion(rf.dn, test)
# - Random Forest aplicando SMOTE sobre train
resultado.rf.sm <- evaluacion(rf.sm, test)
# - Random Forest aplicando ROSE sobre train
resultado.rf.rs <- evaluacion(rf.rs, test)

# - Resultados
tecnicas.par <- rbind(resultado.rf[[1]], resultado.rf.up[[1]],
resultado.rf.dn[[1]], resultado.rf.sm[[1]], resultado.rf.rs[[1]])
rownames(tecnicas.par)<- c("Imbalanced" ,"Upsampling", "Downsampling",
                           "SMOTE", "ROSE")
```

```
tecnicas.par

# - matrices de confusión
resultado.rf[[2]]
resultado.rf.up[[2]]
resultado.rf.dn[[2]]
resultado.rf.sm[[2]]
resultado.rf.rs[[2]]

# - Curvas ROC técnicas ROC
grid.arrange(resultado.rf[[3]], resultado.rf.up[[3]], resultado.rf.dn[[3]],
             resultado.rf.sm[[3]], resultado.rf.rs[[3]], ncol = 3, nrow = 2)
```