ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INFORMÁTICOS

UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DE FIN DE MÁSTER
MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICAL

PREDICCIÓN DE SEPARACIONES EN AERONAVES MEDIANTE REDES BAYESIANAS

AUTORA: Ana Álvarez Suárez

TUTOR: Antonio Jiménez Martín
Juan A. Fernández del Pozo

JULIO, 2017

Dedicado a mi padre,
informático en la sombra.
Y, por supuesto,
a Nietzsche.

Agradecimientos

A todos aquellos que han soportado el caos producido y mi iracundia hacia los ordenadores en casa, especialmente mi madre e Illán, mi particular psicólogo gratuito.

A todos aquellos que me han ayudado en CRIDA, respondiendo incontables dudas de novato sobre navegación aérea y las peores aún dudas (y lloros) sobre bases de datos. Nunca podré invitar a suficientes cafés en compensación. Gracias especialmente a mi paciente tutor de empresa Miguel, a los *descansadores* y al equipo de Desarrollo en general.

Y, por supuesto, a los tutores de este trabajo, Antonio y Juan Antonio, preocupados y serenos hasta el último día.

Resumen

A través del análisis de la inmensa cantidad de datos con los que nos encontramos hoy día en los sistemas de navegación aérea podemos tratar de mejorar la seguridad en la aviación. Uno de los principales incidentes de seguridad del cual existen datos es la pérdida de separación, cuyas causas subyacentes no han recibido aún extensivos estudios desde el mundo académico. Una *pérdida de separación* se define como un conflicto entre un par de aviones en los que la distancia mínima legal definida entre ambos ha sido violada.

En minería de datos predictiva, uno de los posibles enfoques de aprendizaje supervisado es el enfoque probabilístico. Una *red bayesiana* es el tipo más importante de modelo gráfico probabilístico, entre cuyas ventajas encontramos el hecho de que permite, idealmente, un mayor entendimiento del problema al mostrar relaciones causales.

En el presente trabajo se ha logrado la obtención de una red bayesiana que represente las relaciones entre las posibles circunstancias consideradas en la producción de una pérdida de separación dada una aproximación, y las prediga con un buen porcentaje de aciertos, un 77 %, y buenas características como clasificador (área bajo la curva y calibración). La red bayesiana final se obtuvo tras la consideración y comparación con las diferentes opciones de aprendizaje que incluía el software utilizado, GeNIe.

Todo ello gracias a a la obtención y preprocesamiento de un amplio histórico de datos (32 variables y casi 4000 instancias) a través del cruce de diversas fuentes de información aeronáuticas, del uso y modificación de algoritmos de comparación de trayectorias a partir de los existentes en la herramienta PERSEO de la empresa de investigación y desarrollo en navegación áerea CRIDA A.I.E , en colaboración con la cual se ha llevado a cabo este proyecto.

Aunque este trabajo suponga únicamente una primera aproximación al problema, altamente simplificado, gracias a la interpretación de la red bayesiana se han identificado algunos de los factores que más afectan a la producción de una pérdida se separación (y cómo lo hacen): la sectorización, el momento del año, la temperatura isobárica, la altura a la que se produjo y las velocidades en vertical en el momento de la aproximación. El objetivo último del estudio de las pérdidas de separación es la construcción de un sistema de alerta para ayudar a los controladores aéreos.

Abstract

Aviation safety can be improved by the analysis of the immense amount of data available in air navigation systems nowadays. One of the main incidents for which lots of data can be found is the loss of separation, whose underlying causes have not yet received extensive studies from the academic world.

A loss of separation is defined as a conflict between a pair of aircrafts in which the minimum specified separation between them has been violated in controlled airspace. In the field of predictive data mining, one of the possible approaches in supervised learning is the probabilistic approach. Bayesian networks are the most important type of probabilistic graphical models, whose main advantage is the fact that it ideally allows a greater understanding of the problem by showing causal relations.

In the present work, we have derived a Bayesian network that represents the relationships between the possible circumstances considered in the occurrence of a loss of separation given an approximation, and predicts them with a good percentage of hits, 77 per cent, and good characteristics as classifier - area under the curve and calibration. The final Bayesian network was obtained after comparison of the different learning options that included the software used, GeNIe.

For this, a preprocessing task was performed accounting for considerable dataset – 32 variables and almost 4000 instances – through different sources of aeronautical information and the use and modification of algorithms of trajectory comparison from those existing in the PERSEO tool of the R&D air navigation company CRIDA AIE, in cooperation with which this project has been carried out.

Although this work is only a first approach to the problem, highly simplified, thanks to the interpretation of the Bayesian network we have identified some of the factors that most affect the production of a loss of separation: sector configuration, time of the year, isobaric temperature, flight level at which it happened, and vertical velocities at that time. The main goal of this study of separation losses is the construction of an alert system to assist air traffic controllers.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

La aviación, que la Real Academia Española define como "la locomoción aérea por medio de aparatos más pesados que el aire", ha supuesto un punto de inflexión en la vida moderna, gracias a los beneficios sociales y económicos que supone el rápido transporte de viajeros y mercancías. Sin embargo, todavía se enfrenta a muchos retos y perspectivas de mejora, especialmente teniendo en cuenta la creciente demanda. Según la agencia *Eurocontrol*, en 2017 sólo en Europa hubo 10.2 millones de vuelos [18], y las previsiones dicen que la demanda puede llegar a duplicarse en diez años.

Uno de los grandes proyectos de mejora planteados en la actualidad, que cuenta con un amplio respaldo internacional, es el *Single European Sky ATM Research (SESAR) Joint Undertaking*, que busca optimizar el control del tráfico aéreo (*Air Traffic Management* en inglés, ATM) en Europa, y tiene los siguientes objetivos: aumentar el tráfico reduciendo demoras, reducir las emisiones contaminantes, reducir el coste para los usuarios, y mejorar la seguridad en un factor 10 [62].

La seguridad se define como el estado en el cual la posibilidad de daño a personas o a propiedad es reducida y mantenida bajo un nivel aceptable tras identificar y controlar los riesgos [48], y supone un tema fundamental para la aviación.

La aviación es un entorno con multitud de peligros inherentes. En términos de accidentes aéreos, las estadísticas mundiales de la EASA (*European Aviation Safety Agency*) [16] muestran que el número de accidentes no ha cambiado demasiado

en los últimos años, nunca superando los 5 accidentes fatales anuales y en torno a los 20 no-fatales. De cada 10 millones de vuelos, menos de 2 sufren un accidente fatal, y menos de 30 accidentes no fatales, véase la Figura 1.1, obtenida de [16]. Sin embargo, el número de incidentes de seguridad es mucho mayor y excede los 10000 anuales. Las causas de incidentes de seguridad más comunes son: los choques con animales y pájaros, las fuertes turbulencias, los errores en el planteamiento de la aproximación y las pérdidas de separación.

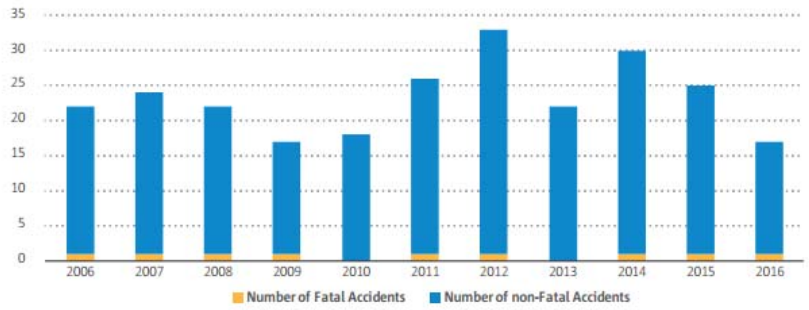

Figura 1.1: Evolución de los accidentes para el transporte aéreo comercial europeo de 2006 a 2016 según la EASA.

Una de las mejores maneras de incrementar la seguridad es a través del análisis de la inmensa cantidad de datos con los que nos encontramos hoy día en los sistemas ATM. Los datos de seguridad que se pueden considerar son, además de los datos de accidentes e incidentes, alertas de los sistemas de seguridad de los aviones y las ya mencionadas pérdidas de separación, menos estudiadas que los anteriores datos.

Una *pérdida de separación* se define como un conflicto entre un par de aviones en los que la distancia mínima legal definida entre ambos ha sido violada [55], y tratar de establecer algunas variables relacionadas con su existencia, encontrando así posibles precursores, será el tema central este trabajo.

Para poder procesar y analizar tales cantidades de dato se necesita del apoyo de procedimientos automáticos. El aprendizaje automático (en inglés, *machine learning*) es una rama de la Inteligencia Artificial que proporciona herramientas para ello. La aplicación más habitual del aprendizaje automático es el aprendizaje supervisado [3], que pretende establecer una correspondencia entre una serie de atributos descriptivos o entradas y unas salidas deseadas del sistema. Los algorit-

mos supervisados más comunes son a su vez los de clasificación [3], donde la salida deseada es una clase que el sistema trata de etiquetar, teniendo una base de conocimiento con etiquetados previos. En el aprendizaje automático actual es bastante común el uso de modelos probabilísticos, como los modelos gráficos probabilísticos (*PGM* por sus siglas en ingles) [3], siendo las *redes bayesianas* el tipo de modelo gráfico más destacado.

Aunque históricamente las áreas de aplicación más comunes del aprendizaje automático son la medicina y la economía, la gran cantidad de datos producida por el sector aeronáutico, particularmente por todos aquellos asociados a los vuelos que transitan día a día nuestros cielos, al igual que el creciente interés en su estudio, ya sea por razones de seguridad humana o por la búsqueda de la reducción costes económicos y temporales, hacen del mismo un campo de aplicación interesante y emergente [43].

1.1. Conceptos previos

1.1.1. La navegación aérea

La navegación aérea es el proceso por el cual se guía una aeronave en vuelo desde una posición inicial hasta un destino a través de una ruta determinada, cumpliendo con unos ciertos requerimientos de eficiencia y seguridad [57]. Es una acción que realiza cada avión de forma independiente, guiado por fuentes externas de información y el equipo técnico adecuado.

Además de los objetivos mencionados en la definición (llegada eficiente y segura al destino), existen otros como:

- Evitar la pérdida de ruta.
- Evitar colisiones con otras aeronaves u obstáculos.
- Minimizar la influencia de las condiciones meteorológicas adversas.

Paralelamente al desarrollo de la aviación a principios del siglo XX se desarrollaron las primeras técnicas rudimentarias de navegación, basadas en la observación

del terreno, mapas y compases. Evolucionaron gracias a la introducción de la *navegación por estima*, más conocida por su expresión en inglés *dead reckoning*, lo cual consiste en la predicción de la posición futura de un avión basándose en su posición y velocidades actuales.

Se comenzaron a diseñar los vuelos en base a ciertos puntos de guía y a usar aparatos como anemómetros. Para mejorar las referencias de los pilotos se comenzaron a utilizar técnicas de navegación astronómica, con la problemática de que tenía que existir la figura del navegador humano que realizara los complicados cálculos en vuelo. Era una *navegación autónoma*.

Para evitar su problemática asociada, surgieron con los años las primeras estructuras terrestres de apoyo que hicieron que la navegación dejara de ser autónoma, como las balizas de luz, las comunicaciones por radio, los sistemas ILS de aproximación a aeropuertos, etcétera.

Cuando ser capaz de llegar al destino eficientemente dejó de ser un reto de tal magnitud y con el aumento de tráfico aéreo, especialmente en torno a aeropuertos, surgió la idea de *circulación* aérea, una navegación que tuviera en cuenta al resto de aeronaves. Era necesario que los aviones siguieran ciertas normas en sus rutas, y que alguien desde torres en los aeropuertos asignara una secuencia de aterrizajes y despegues: estos fueron los primeros controladores aéreos.

En la actualidad, un sistema de navegación aérea debe proporcionar:

- Información estratégica previa al vuelo: limitaciones operativas, predicciones meteorológicas, limitaciones en las ayudas de navegación y limitaciones de ruta.
- Soporte táctico a los pilotos: para evitar conflictos con otros aviones, malas condiciones meteorológicas, optimizar despegues y aterrizajes.
- Infraestructuras radioeléctricas que ayuden a la navegación.

Todo ello en un marco de colaboración internacional jurídica. A este sistema se le denomina CNS-ATM (*Communications, Navigation and Surveillance - Air Traffic Management*). Cada país tiene su propio proveedor ANSP (*Air Navigation Service Provider*) para aviación civil, como AENA en España. En España, además, la aviación militar tiene sus propios controladores aéreos militares, parte del Ejército del Aire.

Para lograr sus fines, la aviación civil y la militar tienen que utilizar espacio aéreo. Aunque en España se utilice el enfoque de separación de control aéreo militar y civil, debe existir un adecuado grado de coordinación civil – militar

Previamente al vuelo, una compañía debe emitir un plan de vuelo, que indica la ruta planteada, procedimientos de despegue y aterrizaje, alternativas, el cálculo de combustible, y otros.

1.1.2. El espacio aéreo

Una ruta es una descripción del camino a seguir por un avión entre aeropuertos, y utiliza para ello varias aerovías y puntos de referencia (más conocidos por su expresión en inglés, *waypoints*). Un *waypoint* es un punto del espacio con su latitud y longitud al que se le ha dado un nombre de cinco letras. Una aerovía es semejante a una carretera en el aire, con un ancho de 10 millas náuticas y una altura de 1000 pies, que conecta dos *waypoints* y puede contener otros intermedios.

El espacio aéreo, para facilitar su control y vigilancia, se divide en diferentes regiones y volúmenes definidos.

La primera división con la que nos encontramos es la de FIR/UIR (*Flying Information Region/Upper Information Region*), amplias regiones que son FIR hasta el nivel de vuelo FL245 y UIR a partir de ahí. Los niveles de vuelo son referencias a la posición vertical de un vuelo y se miden en cientos de pies. En España, el espacio aéreo está englobado en tres FIR/UIR: Barcelona, Madrid y Canarias.

Estos a su vez poseen más subdivisiones, como muestra la Figura 1.2. Estas subdivisiones son:

- Espacio aéreo de ruta: volúmenes que contienen las aerovías que conectan entre sí los espacios aéreos de aeropuertos, los TMA (*Terminal Maneuvering Areas*), sobre los que se ejecuta el control de área en los llamados ACC (*Area Control Centers*). En España existen cinco ACCs: Barcelona y Mallorca para el amplio FIR/UIR de Barcelona, Madrid y Sevilla para el FIR/UIR Madrid y Canarias para el FIR/UIR del mismo nombre.
- TMAs: volúmenes situados sobre los aeropuertos que se encargan de los procedimientos de despegue y aterrizaje.

- CTR (*Control zone*): volúmenes interiores a los TMA que requieren un permiso específico de acceso. Los TMA y los CTR siguen las instrucciones de los controladores de aproximación desde un APP (*Approximation offices*)
- ATZ (*Aerodrome Traffic Zone*): los ATZ son la unidad más interior, con un aérea específica en torno a su aeropuerto, y que son controlados desde las torres de control de los aeropuertos.

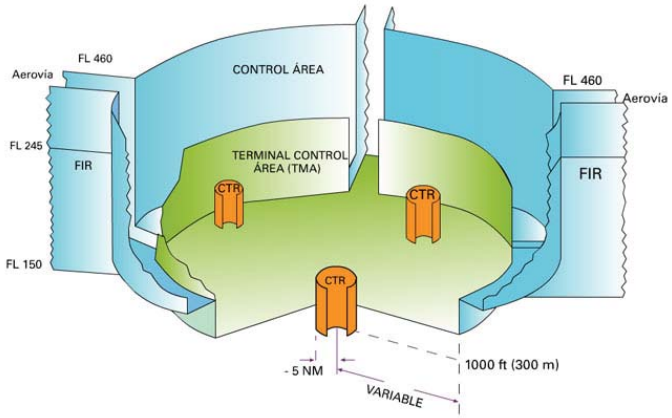

Figura 1.2: Tipos de espacios aéreos controlados. Fuente: AENA.

Además, a tiempo real, se cuenta con sistemas de vigilancia (radares y en el propio avión) que permiten a los controladores conocer la posición de los aviones bajo su responsabilidad en cada momento y pueden mandarle instrucciones. Los controladores intercederían, por ejemplo, de conflicto o aproximación entre varios aviones. También el propio avión cuenta con sistemas de alerta TCAS (*Traffic alert and Collision Avoidance System*) para evitar pérdidas de separación, a los que el piloto debe responder siempre en primera instancia.

Los sectores del espacio de ruta no son constantes en el tiempo, sino que cambian en función de la previsión de tráfico. Sobre unas unidades mínimas denominadas volúmenes, se pueden definir, para un sector grande, distintas configuraciones, es decir, diferentes divisiones. Esas divisiones son también sectores, y cada sector está asignado a una pareja de controladores que lo dirigen. En función de la previsión de tráfico se elige un número de subdivisiones adecuado para facilitar la tarea de los controladores.

1.1.3. El trabajo de controlador

Cada uno de los sectores de control mencionados están gestionados desde una posición UCS (acrónimo de *Unidad de Control de Sector*) [17], equipada con dos pantallas para datos radar, dos pantallas de información meteorológica, dos pantallas táctiles de comunicaciones voz y dos pantallas táctiles de ayuda. En cada posición UCS se encuentran dos controladores, uno de tipo *ejecutivo* y el otro de tipo *planificador*.

Cuando un vuelo va a salir de un sector de control, sus controladores asociados pasan la responsabilidad del mismo a al siguiente pareja de controladores, y así hasta que el vuelo llega a su destino. Por ejemplo, en un vuelo Madrid-Barcelona, el piloto se comunica con unos 16 controladores, 8 en Madrid, entre controladores de torre y el Centro de Control de Madrid-Torrejón y otros ocho en Barcelona.

Los controladores de ruta o de ACC, son la mayoría de los controladores existentes. Los límites entre aproximación y ruta, se establecen entre los centros de control involucrados, mediante las llamadas *cartas de acuerdo*. En líneas generales, el controlador de ruta o de área, controla los tráficos establecidos a un nivel de vuelo, y el controlador de aproximación, los tráficos en como en descenso para aterrizar en el aeropuerto de destino.

Existen gran cantidad de órdenes y mensajes del controlador de ruta al vuelo, pero se pueden resumir en: cambios de nivel, decir un vector de posición o velocidad, indicar un tránsito directo en lugar de por la aerovía a un *waypoint* determinado o desviar hacia otros *waypoint*, informar de meteorología adversa u otras situaciones de interés y transferirle a otro controlador.

Como hemos mencionado, por seguridad, un controlador no puede manejar simultáneamente una gran cantidad de vuelos, y de ahí que existan subdivisiones en los sectores de control que den lugar a nuevos sectores de control en función del tráfico esperado. Para medir el esfuerzo que realiza el controlador en una situación determinada existe el concepto de *carga de trabajo* del controlador, que es una medida de complejidad. Podemos encontrar varias métricas de carga de trabajo de un controlador, pero la mayor parte tienen en cuenta lo siguiente: la concentración de aviones por unidad de tiempo y de volumen de espacio aéreo, el número de coordinaciones con controladores colaterales y de decisiones inmediatas necesarias, el número de comunicaciones por radio necesarias, la fatiga mental y estrés del controlador y la complejidad del espacio aéreo en el que se encuentra.

Para distinguir la complejidad de diferentes espacio aéreos las autoridades publican la capacidad declarada de cada sector, esto es, una medida del número máximo de operaciones posible en un aeropuerto o volumen de espacio aéreo con seguridad.

1.1.4. La pérdida de separación

Una pérdida de separación entre aeronaves ocurre cuando unos mínimos específicos de separación en espacio aéreo controlado son violados [55]. Dichos mínimos están estandarizados y en España se siguen los propuestos por la ICAO (*International Civil Aviation Organization*). Los límites cambian en función del tipo de espacio aéreo en el que se encuentran, y pueden llegar a alcanzar gran complejidad: por ejemplo, en el aeropuerto de Madrid varían en función del tipo de estela de los vuelos implicados. En ruta son 5 NM la distancia horizontal y 1000 pies la vertical.

Cuando existe una confirmación oficial por parte de pilotos o controladores de la existencia de una pérdida de separación se utiliza el término ICAO de AIRPROX (*Aircraft Proximity*). En este trabajo no disponemos de listas oficiales de incidentes AIRPROX sino de un algoritmo que utiliza las trazas o listas de puntos 4D para obtener situaciones en las que se han superado dichos mínimos.

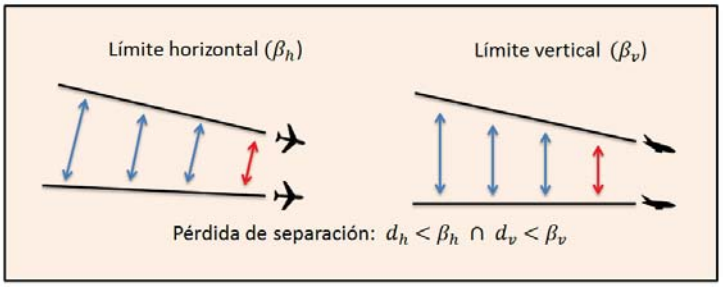

Figura 1.3: Definición gráfica de pérdida de separación

La pérdida de separación puede ocurrir en un plano vertical, horizontal o en ambos. Consideraremos aquí sólo aquellas pérdidas de separación que ocurran en ambos planos ya que en caso contrario no existe riesgo real. Las pérdidas de separación respecto al terreno o a espacio aéreo restringido no serán tampoco consideradas.

Los posibles efectos de una pérdida de separación dependen de la gravedad de la misma.

- Pueden resultar en encuentros de estela turbulenta, con los riesgos asociados a turbulencias.
- Puede llegar a ocurrir una colisión, o a producirse lesiones como resultado de violentas maniobras para evitarla.

Para evitar que ocurran este tipo de incidentes, existen multitud de defensas más allá del control aéreo, como los ACAS *(Airborne Collision Avoidance System)*, sistemas de detección de otras aeronaves que no requiere asistencia desde tierra, una de cuyas implementaciones es el TCAS (*Traffic Alert and Collision Avoidance System*), asistidos por procedimientos especializados en respuesta; y diversos sistemas de protección desde tierra, como el STCA (*Short-Term Conflict Alert*)

1.2. Estado del arte

1.2.1. Análisis de pérdidas de separación

Las pérdidas de separación, como todo incidente de seguridad, son objeto de amplio estudio. Desde hace más de treinta años, la ICAO considera obligatorio que siempre que ocurra uno de estos incidentes, éste sea reportado, bajo el nombre de AIRPROX [21]. Incluso aunque no sea considerado un incidente grave, la existencia del mismo, junto con su categoría y sus circunstancias, son comunicados y reportados si es posible durante el mismo vuelo en el que se produjo. Los procedimientos y regulaciones se dejan a disposición de las autoridades de cada país, los cuales llevan a cabo una investigación individualizada de los mismos.

Sin embargo, dichos datos no están a disposición pública, y las investigaciones realizadas buscan esclarecer causas particulares y no buscar un modelo del problema. Entre sus posibles **causas** se encuentran órdenes de controlador, como un cambio de nivel erróneo, decisiones de piloto y fallos en la electrónica asociada [55]. Algunos países como Inglaterra [44] y Suiza [20] tienen comités destinados únicamente al análisis de estos datos, pero sus resultados, son, nuevamente, información clasificada.

Desde una perspectiva académica, la investigación sobre las pérdidas de separación se ha centrado en su carácter de conflicto de trayectorias, y por tanto, en la predicción y evitación de los mismos a partir de la geometría de las trazas de los vuelos [61], pero no en el estudio de sus causas subyacentes. Sí que existen algunos estudios estadísticos de baja repercusión como [39], que utiliza datos de Nueva Zelanda.

En general, en el campo de la navegación aérea así como en aeronáutica y muchos otros campos de la ingeniería, la mayor parte de la investigación es investigación realizada desde la industria, y en consecuencia no termina con una publicación al uso, sino que forma parte de documentación privada o, incluso, patentes. Es por eso que iniciativas públicas como la mencionada SESAR [62], de carácter internacional y que aboga por colaboración entre empresas, son de tal importancia.

Gracias al hecho de que este trabajo ha sido realizado en la empresa de I+D en navegación aérea *CRIDA* (Centro de Referencia de Investigación, Desarrollo e Innovación ATM, A.I.E.) [11], se ha podido contar con la colaboración de expertos participantes en el proyecto europeo *SafeClouds* [31], que engloba distintos proyectos de aplicación de técnicas de minería de datos a la seguridad en la navegación aérea, dentro del Programa Horizonte 2020, y que cuenta con la participación de más de 15 agencias, aerolíneas y empresas de investigación.

Así pues, entre los diferentes posibles factores que afectan a la aparición de una pérdida de separación, nos encontramos, según [39] y los aportes de *SafeClouds*, con:

- Localización espacial y temporal.
- Condiciones de la aeronave (modelo, peso).
- Información meteorológica.
- Velocidad y ángulo en el momento de la pérdida de separación.
- Separación horizontal, vertical y temporal inicial.
- Desviaciones respecto al plan de vuelo.
- Conocimiento de la situación y comunicación por parte de pilotos y controladores.

- Situación del tráfico aéreo.
- Existencia de alertas de TCAS y su tipo.
- Información personal de la tripulación aérea y los controladores (experiencia, edad).

1.2.2. Redes bayesianas

El *aprendizaje automático* es un campo dentro de las ciencias de la computación que, apelando a la definición de Arthur Samuel en 1959, otorga a los ordenadores la capacidad de *aprender* sin estar programados explícitamente. Engloba la minería de datos predictiva, es decir, el estudio de algoritmos que aprenden y hacen predicciones para datos a partir de la creación de un modelo gracias a la existencia de datos de entrada [26]. Cada instancia de un conjunto de datos se representa según unos atributos, los cuales pueden ser continuos, categóricos o binarios. Si existen instancias que contienen la salida o predicción deseadas, se habla de aprendizaje de tipo supervisado, frente al no supervisado.

La selección de algoritmos de aprendizaje para un problema dado es una decisión compleja dada la gran variedad y diferentes ventajas y desventajas de los mismos. Podemos encontrar una sucinta comparación en [34] para los métodos supervisados. Uno de los posibles enfoques de aprendizaje supervisado es el enfoque estadístico o probabilístico cuyo máximo representante son las *redes bayesianas*.

Se puede decir que el término redes bayesianas fue acuñado por el informático y filósofo de origen israelí Judea Pearl en 1985 [45], aunque el concepto actual de red bayesiana y su campo de estudios asociado se afirmó con la definición y descripción de sus propiedades en textos posteriores de Pearl [46] y Neapolitan [41].

Algunas ventajas de las redes bayesianas sobre otros métodos de aprendizaje automático son [28]:

- Aceptan satisfactoriamente conjuntos de datos incompletos. Un modelo típico de clasificación o regresión, ante una variable no observada que tenga correlación con otras del modelo, va a producir una predicción poco precisa, mientras que la forma de tratar con correlación de las redes bayesianas evita ese error.

- Permiten, idealmente, un mayor entendimiento del problema al mostrar relaciones causa-efecto.
- Facilitan la inclusión de conocimiento experto en el modelo.
- Es un método robusto frente al sobreajuste u *overfitting*.
- Incluyen la posibilidad de tener en cuenta conocimiento previo experto del dominio en un problema dado en su estructura, ya que casi todos los algoritmos de aprendizaje permiten el añadido o prohibición obligada de arcos. Esta característica es considerada como la más interesante por [34].

Una *red bayesiana* es el tipo más importante de modelo gráfico probabilístico, existiendo otros como los *campos de Markov* [12]. Podemos encontrar una comparativa en [56]. Estos son menos conocidos que las redes bayesianas por su opacidad en contraste con la facilidad de comprensión asociada a las redes bayesianas [28], razón que los hace también menos interesantes para nuestro estudio.

De forma sucinta, una red bayesiana está compuesta por un grafo acíclico $G(V, E)$ (la estructura de la red bayesiana) donde cada nodo del conjunto V representa una variable aleatoria y sus arcos E representan dependencias probabilísticas entre variables y una distribución de probabilidades condicionadas (los llamados parámetros de la red) para cada nodo dado su conjunto de padres.

Un modelo probabilístico necesita de la distribución conjunta de sus variables para su descripción, mas la obtención de la misma como producto de probabilidades condicionadas a partir de la conocida regla de Bayes sería de gran complejidad si no fuera por el concepto de independencia condicional [12], mostrado en (1.1) para dos eventos E y F condicionalmente independientes (c.i).

$$E, F \ \ son \ \ c.i \leftrightarrow P(E|F, G) = P(E|G) \cap P(F|E, G) = P(F|G). \tag{1.1}$$

La idea fundamental pues detrás de una red bayesiana es el hecho de que cada nodo es condicionalmente independiente de sus no sucesores (aquellos que no pertenecen a su descendencia, utilizando la terminología familiar habitual) dados sus padres, es decir, que satisface la condición de Markov, lo cual reduce enormemente el número de productos en la mencionada cadena y, por tanto, los tamaños de las

tablas de probabilidad a *priori* de cada nodo de la red. Si la variable X_1 es condicionalmente independiente de todas las demás variables dada X_2 nos encontramos con:

$$P(X_1, X_2, ..., X_n) = P(X_1|X_2)P(X_2, X_3, ..., X_n). \tag{1.2}$$

Aplicaciones

Clásicamente las redes bayesianas han sido utilizadas sobre todo en los campos de diagnóstico médico, diagnóstico de fallos y en biología, además de en economía y detección de fraude [12]. Más recientemente, todavía dentro del campo de la biología, se ha utilizado en el estudio de interacciones entre genes [22], pero también se han abierto muchos más campos, como la ciberseguridad [65] y las redes sociales [36].

En el campo que trata este trabajo, el campo de la aeronáutica, existen también estudios que utilizan redes bayesianas para diferentes temas. Encontramos modelos de diagnóstico de fallos en motores de aeronaves [50], detección de vehículos en vigilancia aérea [6] o estudios de dinámicas en aeropuertos como el realizado sobre la propagación de las esperas en [66].

Algunos tipos específicos de redes bayesianas son más utilizados en la actualidad: las redes bayesianas dinámicas para clasificación o predicción en tiempo real [32], los métodos de clasificación clásicos como el ingenuo o el TAN continúan estando en vigor mientras aparecen métodos nuevos como los de multiclasificación (con un vector de clases) [2].

Algoritmos de aprendizaje de redes bayesianas

Además de por conocimiento experto, como ya hemos mencionado las redes bayesianas se pueden construir aprendiendo de un conjunto de datos. Existen algoritmos para aprender la estructura de la red, y también algoritmos para aprender los parámetros (tablas de probabilidades) dada la estructura. Muchos algoritmos de aprendizaje de estructura también estiman los parámetros en el proceso.

El método de estimación de parámetros más común es la *estimación de máxima verosimilitud* [12]. Para evitar la problemática existente si los datos son dispersos se

aplica una distribución previa a las variables, estando considerada como óptima la Dirichlet [24]. Existen otros métodos, enfocados a casos con variables no categóricas o con datos ausentes, que utilizan redes neuronales, muestreo de Gibbs... explicadas en [42].

Respecto a los algoritmos de aprendizaje de estructuras, existe una gran variedad de ellos, dado el carácter NP-duro del problema. Pueden agruparse en tres categorías:

- Métodos basados en *tests de independencia condicional*: realizan un estudio cualitativo de las relaciones de dependencia e independencia entre las variables existentes para posteriormente buscar una red que represente las más posibles. Un ejemplo de estos algoritmos es el PC [58], el cual comienza con el grafo completo, al que se le van quitando los arcos que no tengan ninguna relación de independencia condicional, y seguidamente los que tienen relaciones de primer orden y así.

- Métodos basados en una métrica o puntuación (*score*): buscan una estructura que maximice dicha métrica, que representa lo bien que se adecúa a los datos la estructura, usando un método de búsqueda posiblemente metaheurístico. La métrica y el método de búsqueda utilizados definen a este tipo de algoritmos. Algunos de los métodos de búsqueda más utilizados son *búsqueda voraz* [9] y *algoritmos genéticos* [64]. Algunas de las métricas clásicas más utilizadas son de tipo bayesiano: K2 (de 1992 [9] y BDeu [5], ambas casos particulares de la inabarcable BD (*Bayesian Dirichlet*). Más recientemente han parecido métricas basadas en *teoría de la información* (en la comprensión que se puede alcanzar para una estructura dada), como *Minimum Description Length* (MDL), que además busca la red más simple posible y su versión moderna NML (*Normalized Minimum Likelihood*) [54].

- Métodos híbridos de los dos anteriores, como el sistema BENEDICT [1].

Los algoritmos de aprendizaje de redes bayesianas tratan en general con datos de caracter discreto, si bien existen implementaciones para datos continuos. En el caso de conjuntos de datos que incluyan discretos y continuos, es necesario utilizar distribuciones para modelar los discretos o utilizar alguno de los métodos, con restricciones, diseñados para ambos tipos de datos [13].

Librerías

Gracias a lo extendido de su uso, existen multitud de implementaciones disponibles para diferentes tipos de usuarios.

- Una de las herramientas más conocidas es GENIE [15], una interfaz de usuario construida sobre la librería de modelado y aprendizaje SMILE para el lenguaje C++. Es un producto comercial que lleva en continua actualización desde 1999 y la interfaz gráfica que posee es versátil y de fácil manejo, con una buena visualización de las redes. Como puntos negativos nos encontramos con poca variedad de algoritmos de aprendizaje y escaso acceso a los parámetros internos de los mismos, así como difícil manipulación de los resultados.
- Otra herramienta gráfica, esta vez de código libre, es TETRAD [51]. Un gran proyecto de redes causales en actualización constante de construcción de redes causales que incluye entre sus métodos las redes bayesianas. Su uso es ligeramente más complejo y no incluye una gran variedad de algoritmos.
- Existe una amplia diversidad de herramientas gráficas disponibles para la creación de redes bayesianas. Algunas no están actualizadas, como la española Elvira [8], que no proporciona soporte para sistemas operativos modernos, mientras que otras son de más comerciales y limitan el tamaño del modelo en la versión para estudiantes, como Netica [29], pese a su gran variedad de características. También está disponible la potente herramienta *Hugin* [38], referente en la industria que cuenta sólo con versiones de pago.
- Las librerías de programación son siempre más adaptables que las interfaces gráficas. Existen muchas librerías para distintos lenguajes de programación. Una de las más utilizadas es BNLearn [52] para el lenguaje estadístico R, creada en 2009, la cual posee una extensa variedad algorítmica. Sin embargo, su herramienta de dibujo de redes no permite una correcta visualización para redes lo suficientemente grandes.

El problema de la causalidad

Aunque intuitivamente puede parecer que una red bayesiana, o al menos una *buena* red bayesiana, debería representar relaciones de causa-efecto y ser por tan-

to una red causal, la existencia de un arco dirigido entre dos nodos en una red bayesiana no implica necesariamente causalidad, sino simplemente algún tipo de correlación [53]. Prueba suficiente de ello es la existencia de clases de equivalencia [12], desde un punto de vista probabilístico, en las estructuras de la redes bayesianas, esto es, distintas disposiciones de arcos entre nodos son en la práctica una misma red y, por tanto, distintas interpretaciones causales posibles.

Utilizando la definición de causalidad de Pearl [47], para que una red bayesiana sea causal tiene que cumplir una serie de requisitos adicionales, que no serán descritos aquí, o estar, por supuesto, producida por adecuado conocimiento experto. Así pues, las redes bayesianas creadas por los algoritmos clásicos de aprendizaje descritos no son necesariamente causales, aunque recientemente se estén desarrollando técnicas para ello [27].

Capítulo 2

Objetivos

En este trabajo, se pretende llevar a cabo un estudio de los factores causales de una pérdida de separación más allá de su traza, o al menos de factores correlacionados que puedan *avisar* de la posible producción de una pérdida de separación, es decir, predecirla. Dicha búsqueda de precursores de una pérdida de separación se enmarca dentro del mencionado proyecto europeo *SafeClouds* del cual CRIDA A.I.E es participante, con el propósito último de construir un sistema capaz de alertar a los controladores cuando, además de la cercanía de dos vuelos en el espacio-tiempo, aparezcan otros factores contribuyentes, a tiempo real. Un resumen visual del problema se muestra en la Figura 2.1.

Para obtener un modelo de predicción de pérdidas de separación en función de factores asociados, se ha obtenido un conjunto de datos históricos de las diversas herramientas y bases de datos disponibles en CRIDA. En primer lugar, se obtienen, a través de análisis de trayectorias, casos de aproximaciones de vuelos que han llegado a incurrir en pérdida de separación y casos que no; para posteriormente añadir información asociada a los mismos obtenida de las bases de datos de CRIDA. Se utilizarán redes bayesianas para la construcción de un modelo probabilístico que permita la visualización de las relaciones entre las distintas variables planteadas y la clase (aproximación o pérdida de separación).

Este estudio se ha centrado en pérdidas de separación (y aproximaciones sin incidente final) ocurridas en ruta, ubicadas en el espacio aéreo español, y producidas en un intervalo de 6 meses. Se dispone de información directa de la pérdida de separación, de los vuelos, de su situación previa, de la aeronave, del tráfico del sector

y de la meteorología, pero no de datos de las alertas TCAS ni información directa sobre los pilotos y controladores implicados. Tampoco hay disponibles datos del espacio aéreo entre la Península y las Canarias, ya que pertenece a Marruecos.

Como restricciones añadidas, mencionar que al no ser datos de carácter público, en esta memoria no se podrá aportar información completa sobre su obtención ni mostrar instancias de los datos. En caso de que exista necesidad de mostrarlos, se mostrarán deslocalizados. Decir además que este trabajo es sólo una primera aproximación al problema, el cual posee gran complejidad no sólo en sus causas subyacentes sino también en su detección, descripción y solución.

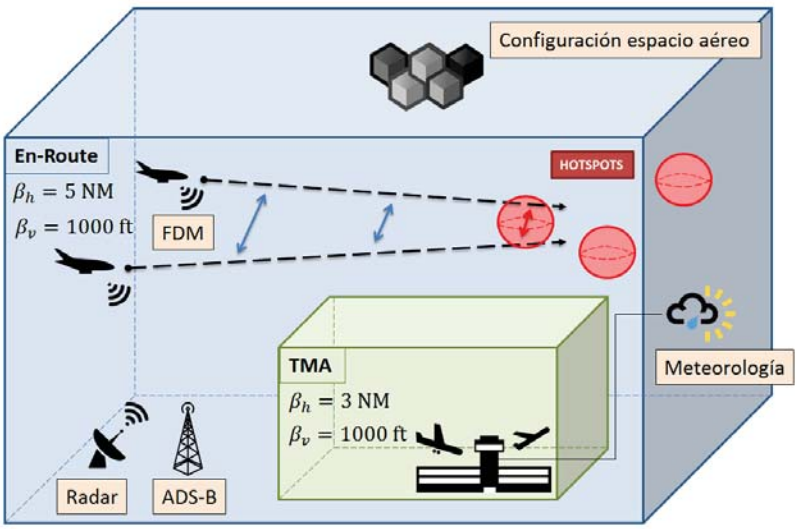

Figura 2.1: Resumen visual del entorno de una pérdida de separación y sus límites en España.

Con el fin de alcanzar el objetivo global planteado, proponemos la siguiente metodología en cuyos pasos se representan los distintos objetivos específicos que han de alcanzar para el correcto desarrollo del trabajo.

1. **Proceso de obtención de los datos.** En este trabajo, no se parte de una base de datos ya existente sino que una gran parte del esfuerzo será dedicada a su recolección. En el proceso tienen lugar las siguientes etapas:

   1.1. Estudio teórico de posibles factores de interés (Apartado 1.3.1).

1.2. Obtención de las pérdidas de separación utilizando el código disponible en la parte de Análisis de la herramienta PERSEO (descrita en el Capítulo 4) realizando ligeras modificaciones. El código compara trayectorias de los vuelos y devuelve identificadores de vuelos y posiciones espacio-temporales de la pérdida de separación. Previamente se utilizarán técnicas de suavizado de datos y detección de valores atípicos en las trazas.

1.3. Ampliación del código de PERSEO para que devuelva también aproximaciones que no llegan a ser pérdidas de separación, y obtención de las mismas en el mismo espacio temporal que las pérdidas de separación. Se consideran aproximaciones a aquellas que se encuentran entre la distancia de separación mínima y 1,5 veces esta distancia, y que posteriormente no llegan a ser una pérdida de separación. Como el número de éstas probablemente supere, como es lógico, al de pérdidas reales, se deberá proceder a su nivelado, tratando además de quitar aquellas aproximaciones para las que falten datos.

1.4. Enriquecimiento de ambos tipos de datos con otras variables procedentes de otras fuentes de datos y con algunas variables cuyo cálculo deberá ser programado (más información en el Capítulo 3), hasta la consecución de una base de datos inicial con 72 variables descriptivas de estas aproximaciones y más de 4000 instancias.

2. **Preprocesado de los datos.** La base de datos inicial obtenida podrá poseer demasiadas variables para su correcto tratamiento, además de poseer imperfecciones.

2.1. Estudio de datos ausentes (Apartado 3.2.1). Utilizando el lenguaje de programación R se buscarán, eliminarán o sustituirán aquellos datos que faltaban en la base de datos.

2.2. Estudio descriptivo de las variables (Apartado 2.4.2). De nuevo con el lenguaje R y sus funciones disponibles se realizará un pequeño estudio de estadística descriptiva sobre las variables.

2.3. Estudio de datos atípicos (Apartado 3.2.2). Utilizando R y sus funciones de diagrama de cajas y de creación de histograma se identificarán los valores atípicos, sus posibles causas y se decidirá su conservación.

2.4. Selección de variables (Apartado 3.2.3) para evitar el uso de variables irrelevantes además de reducir el número de variables a uno manejable

por un método de aprendizaje automático, utilizando R y sus funciones de matriz de correlaciones y el paquete *Boruta* para la selección.

2.5. Discretización de datos continuos (Apartado 3.2.4) y reducción del dominio de algunas variables categóricas para poder utilizarlas en el *software* GENIE.

3. **Aplicación de métodos de aprendizaje automático.**

3.1. Selección del algoritmo de aprendizaje automático. Como el objetivo del proyecto es el estudio analítico de las variables que afectan a las pérdidas de separación como clase supervisada, ya previamente a la obtención de los datos existe la intuición de que el mejor enfoque puede ser la utilización de redes bayesianas, ya que son muy visuales en comparación con la mayor parte de algoritmos de clasificación, manejan bien el sobreajuste y además devuelven una probabilidad sobre la que se podría basar un sistema de alerta. Una posible opción alternativa es la utilización de clasificadores bayesianos ingenuos o aumentados, mas no se considerarán dada la fuerte interconexión entre las variables a estudiar frente a las asunciones de independencia que precisan estos modelos. Más información en el capítulo 5.

3.2. Creación de redes bayesiana con GeNIe (herramienta descrita en el Apartado 4) con los algoritmos de aprendizaje disponibles (Apartado 5.1.1), si procede. La elección de dicha herramienta se producirá tras la comparación de las tres opciones comentadas en el estado del arte y la prueba de las mismas.

3.3. Validación de las redes creadas por el método de validación cruzada en el propio GeNIe y selección de la mejor para su posterior estudio (apartado 5.1.2) y estudio e interpretación de la red seleccionada (apartado 5.1.3).

Capítulo 3

Descripción y proceso de obtención de los datos iniciales

Para el estudio de los precursores de pérdidas de separación mencionados, se obtiene inicialmente un amplio conjunto de datos y atributos que se procederá a describir en este capítulo. Está formado por 4402 instancias y 65 variables, procedentes de distintas fuentes y que abarcan algunos de los distintos factores que podrían afectar al objetivo de este estudio.

La **variable clase** principal clasifica las instancias de conflictos en aquellas que han entrado en pérdida de separación y aquellas que no. Existe otra variable clase secundaria que además divide a las que sí han llegado a ser pérdidas de separación en severidad alta y severidad baja.

Debido a la amplitud de la bases de datos y sus diferentes orígenes, dividiremos las restantes variables en diferentes grupos para facilitar su explicación. Estos son: aquellos datos obtenidos directamente del cálculo de pérdidas de separación, datos sobre los vuelos que la produjeron, datos geométricos y físicos sobre la situación del avión en el momento de la separación, información meteorológica y datos sobre el sector y de tráfico del sector.

3.1. Descripción de las diferentes variables

3.1.1. Variables relacionadas con la pérdida de separación

Proceso de obtención

Con la herramienta PERSEO procesamos todas las trazas de los vuelos en España en seis meses, cuyas fechas exactas deben quedar deslocalizadas. Se realizó una modificación sobre el código de dicha herramienta específicamente para este trabajo para ser capaces de localizar también los que hemos denominado como aproximaciones, y en inglés, *previolation* frente a *violation*.

Esta herramienta devuelve una gran cantidad de variables sobre la pérdida de separación en sí, un total de 78 variables. De éstas, se retiraron inicialmente las variables que incluían únicamente metadatos de la base de datos, como pueden ser identificadores, fechas de actualizaciones, archivos de origen...

Como se había decidido previamente que una de las restricciones impuestas a nuestro problema sería tener en cuenta únicamente las separaciones en ruta, también se eliminaron las variables referentes a los límites utilizados (ya que en ruta siempre son los mismos), así como las que indicaban el tipo de pérdida de separación (RUTA) y las referentes a detalles aeroportuarios.

El resto de variables describen la posición 4D (latitud, longitud, y nivel de vuelo para los dos aviones considerados e instante temporal) de los siguientes eventos que definen una pérdida de separación:

- Inicio de la pérdida de separación o momento inicial en el que se violan los límites definidos.
- Instante en el que se produce la menor separación vertical.
- Instante en el que se produce la menor separación horizontal.
- Instante en el que se produce la menor separación diagonal.
- Fin de la pérdida de separación.

De entre todas ellas nos quedamos con algunos de los datos iniciales, finales y los de la separación anterior que fuera menor porcentualmente con su límite

establecido. Se eligieron éstas ya que lo que nos interesa son las causas de la pérdida de separación y no su desarrollo, así que lo relacionado con el fin de la misma, salvo el momento final para tener una idea de la duración, no se tuvo en cuenta.

Además, los datos respecto a las dimensiones en las cuales no se produjo la violación del límite, o era menos considerable que otra, no son de suficiente interés. Se redujo pues significativamente el número de variables, de 78 a 22. Este primer filtro de variables, que describiremos para cada tipo de variable, está basado en conocimiento experto y sólo es el primer paso en la obtención del conjunto final de variables.

- *startime*: fecha y hora de comienzo de la pérdida de separación. Es interesante tener en cuenta en qué momento del día y qué día se produjo la pérdida de separación y saber si ésta puede afectar.
- *start_sep_h*: separación o distancia horizontal (en millas náuticas) existente entre ambos aviones en el momento de inicio de la pérdida de separación.
- *start_sep_v*: separación o distancia vertical (en pies) existente entre ambos aviones en el momento de inicio de la pérdida de separación.
- *start_sep_3d*: separación o distancia diagonal (en millas náuticas) existente entre ambos aviones en el momento de inicio de la pérdida de separación.
- *start_lat_a*, *start_lat_b*: latitud (en grados) de la ubicación de uno de los aviones (los considerados vuelos A y B) en el momento de inicio de la pérdida de separación.
- *start_lng_a*, *start_lng_b*: longitud (en grados) de la ubicación de uno de los aviones (los considerados vuelos A y B) en el momento de inicio de la pérdida de separación.
- $start\_fl\_a$ $start\_fl\_b$: coordenada z, es decir, la altura en nivel de vuelo, medido en niveles de vuelo (cientos de pies), de la ubicación de uno de los aviones (los considerados vuelos A y B) en el momento de inicio de la pérdida de separación.
- *endtime*: fecha y hora de final de la pérdida de separación. Es interesante principalmente de cara a considerar la duración de la misma, el tiempo que los aviones se mantuvieron en situación de riesgo.

- *min_p_t*: fecha y hora del momento en que la pérdida de separación fue más severa. Posiblemente innecesaria.
- *min_p_sep_h*: separación o distancia horizontal (en millas náuticas) existente entre ambos aviones en el momento que la pérdida de separación fue más severa.
- *min_p_sep_v*: separación o distancia vertical (en pies) existente entre ambos aviones en el momento que la pérdida de separación fue más severa.
- *min_p_sep_3d*: separación o distancia diagonal (en millas náuticas) existente entre ambos aviones en el momento que la pérdida de separación fue más severa.
- *min_p_lat_a*, *min_p_lat_b*: latitud de la ubicación de uno de los aviones (los considerados vuelos A y B) en el momento de que la pérdida de separación fue más severa.
- *min_p_lng_a*, *min_p_lng_b*: longitud de la ubicación de uno de los aviones (los considerados vuelos A y B) en el momento que la pérdida de separación fue más severa.
- *min_p_fl_a* , *min_p_fl_b*: coordenada z, esto es, la altura en nivel de vuelo medida en niveles de vuelo (cientos de pies), de la ubicación de uno de los aviones (los considerados vuelos A y B) en el momento
- *min_p_severity*: severidad máxima de la pérdida de separación, considerada a partir del porcentaje de la distancia mínima en que se incurre. Valora del 1 al 10 los casos de pérdidas de separación y con 0 los casos que no llegan a serlo. Es una posible variable clase alternativa si es reducida a alta, media y baja.

3.1.2. Variables con información de los vuelos

En las bases de datos de PERSEO se almacenan además de las trazas multitud de atributos sobre un vuelo obtenidos de muy diversas fuentes. Algunas de las fuentes originales son IFS y GIPV, esto es, datos radar (desde tierra) y de planes de vuelo que provienen de PALESTRA, un sistema de ENAIRE que recolecta muy distintos datos de entornos aeroportuarios. Algunos de los mismos son:

- En la dimensión de vuelo nos encontramos con hasta 46 variables distintas. En un primer filtro para reducir el número de variables de nuestro modelo, se elimina todo lo relacionado con metadatos (identificadores, presencia de actualizaciones, archivo fuente...etcétera), así como momentos de eventos que no son de interés para nuestro estudio, como puesta de calzos y quitado de calzos, que ocurren en el aeropuerto y son dependientes de la hora de despegue y aterrizaje que nos interesan.

  Aunque nos quedaremos con el aeropuerto de origen y destino, no lo haremos ni con el número de la pista de aterriza (de nuevo, porque sólo consideramos RUTA) ni con el número de *stand* en el que se aparca el avión. Finalmente, sólo se han considerado datos (y variables) reales históricos y no aquellos que parten de los llamados planes de vuelo, puesto que no nos interesan las previsiones para realizar este tipo de análisis.

- Respecto a la dimensión relacionada con el avión en sí, existen 8 variables en PERSEO. Son en general datos sobre el modelo de avión disponibles. Algunos ejemplos de información existente sobre un avión que no hemos tenido en cuenta son consumo y emisiones. Las variables relacionadas con el avión resultan de gran importancia para el asunto a tratar, ya que la maniobrabilidad (ángulo de giro, velocidad máxima, altura de crucero...) de cada modelo de avión es muy distinta, y puede inducir a distintos tratamientos y prioridades por parte del controlador aéreo.

  Además, nos da información sobre el tipo de avión que es, lo cual se refleja directamente sobre el tipo de controlador que se encarga de su seguimiento, ya que puede haber también aviones militares (que tienen sus propios centros de control) y de aeródromos privados. La falta de comunicación entre centros de control aéreo civiles y militares es considerada una de las principales fuentes de conflictos.

Para una pérdida de separación, tenemos esas 54 variables por duplicado, de las cuales, tras el filtrado descrito, usaremos las siguientes 10 por duplicado:

- *callsign_a*, *callsign_b*: código utilizado en un vuelo para su identificación en las comunicaciones, habitualmente formado por el código de su compañía más 4 caracteres que le llevan a convertirse en un identificador único para un día. A menudo, el mismo callsign en distintos días implican la misma ruta,

así que se puede considerar como un identificador inexacto de ruta. Además, es también habitual que los mismos pilotos realicen las mismas rutas.

Como se comentó en la introducción, una de las posibles causas de una pérdida de separación es un error el piloto, y es por tanto posiblemente interesante mantener la variable. Si es un callsign militar, en lugar de por el prefijo de la compañía comienzan por unos prefijos militares establecidos como TUCAN y ALCON y van seguidos de dos números.

- *company_a*, *company_b*: a partir del callsign, podemos obtener la compañía que opera el vuelo (si es comercial) o sencillamente clasificarlo como militar. Las compañías que operan los vuelos son interesantes ya que, aunque idealmente no debería afectar en absoluto, se dice que existe la posibilidad de que exista diferencia de tratamiento por parte de los controladores. Por ejemplo, como es habitual que un avión de una compañía de bajo coste no lleve combustible de sobra, evitarían hacerles realizar desvíos o cambios que les lleven a aumentar el consumo. Esta variable contará pues con las compañías civiles (anonimizadas) o el valor M para los militares.
- *adep_id_a*, *adep_id_b*: código identificador del aeropuerto de salida del vuelo.
- *ades_id_a*, *ades_id_b*: código identificador del aeropuerto de destino del vuelo.
- *atot_a*, *atot_b*: día y hora de despegue del vuelo.
- *aldt_a*, *aldt_b*: día y hora de aterrizaje del vuelo. Entre ambas se obtendría la duración del mismo. No sabemos todavía si son variables significativas.
- $aircraft\_a$, $aircraft\_b$: modelo de avión. Como dijimos, la maniobrabilidad de cada modelo de avión y por tanto cómo puede actuar ante una pérdida de separación es determinante.
- *wake_a*, *wake_b*: tipo de estela que deja el avión. Dependiente del modelo de avión y sujeta a una posible eliminación, es conceptualmente de gran interés ya que de ellas depende la aparición de uno de los peligros que conlleva un pérdida de separación: las turbulencias de estela. Es un concepto complejo que tiene especial importancia en aterrizaje y despegue. Dentro del alcance de este trabajo, decir simplemente que los aviones de mayor envergadura dejan una estela más pesada causando turbulencias de distinta gravedad a aviones más ligeros, y que esto es algo que los controladores tienen en cuenta al posicionarlos.

- *description_a*, *description_b*: variables categóricas que indican el tipo de avión. Sujetas a posible eliminación, dependen del modelo de avión.
- *enginetype_a*, *enginetype_b*: variables categóricas que indican el tipo de motor del avión. Sujetas a posible eliminación, dependen del modelo.

3.1.3. Variables relativas al sector

Dada la posición del vuelo en el comienzo de la pérdida de separación, podemos saber a qué sector pertenece, y en qué configuración del sector. Como ya mencionamos, una de las posibles causas o factores contribuyentes a la existencia de una pérdida de separación es un error del controlador. Un sector en su mínima división está controlado directamente por una pareja de controladores.

Aunque no es posible (por protección de datos) hacer un seguimiento en particular de qué controladores estaban asignados en un momento dado para tener en cuenta su edad, sexo o experiencia de cara a valorar la relación entre estos factores y la existencia de pérdida de separación, sí qué podemos intentar obtener una medida del estrés al que estaban sometidos en el momento de la pérdida de separación con una métrica muy simplificada de carga de trabajo, ya que no es posible calcular alguna de las existentes [37].

La métrica que vamos a utilizar es una sencilla proporción entre el número de vuelos existentes en el sector en el instante de inicio de la separación (tráfico o *occupancy* del sector) dividido por la capacidad declarada del sector:

$$\mu(t)_{WL} = \frac{Occupancy(t)}{Capacity}. \tag{3.1}$$

El denominador es constante para cada sector y se encuentra declarado en publicaciones de la AIP (*Publicación de Información Aeronáutica*), mientras que el numerador ha tenido que ser obtenido a través de consultas geométricas a bases de datos que incluyen las trazas unto a punto de los vuelos y los polígonos de los sectores (como ejemplo del trabajo en base de datos, se incluye esta consulta en el Apéndice B).

- *sector*: el sector en el que ha ocurrido la pérdida de separación, considerando

el punto medio entre la localización de los dos aviones en el momento de mayor severidad de la pérdida de separación.

- *atc_unit*: es la unidad de control aéreo a la que corresponde ese sector. En España existen 6.
- *sectorization*: es el tipo de sectorización al que corresponde el sector, es decir, el número de divisiones y tipo de división en las que está dividido su sector padre. Es un indicador de las previsiones de tráfico que había, ya que la configuración de un sector se elige anteriormente en función de las previsiones de tráfico aéreo existentes.
- *occupancy*: es el número de aviones que había en el sector en el momento en el que comenzó la pérdida de separación, es decir, el número de aviones que estaba manejando el controlador.
- *occupancy/capacity*: es el número de aviones existentes partido de la capacidad declarada del sector, es decir, del número máximo de aviones que podría manejar en una hora la pareja de controladores asignada para un sector determinado. Al ser la *occupancy* dada una medida instantánea frente a una medida de aviones en una hora, el resultado no tiene significado porcentual, pero sí modula la unidad de medida de carga de trabajo de controlador que supone, ya que no es lo mismo que haya 20 aviones en un sector sencillo que en un sector complejo.

3.1.4. Variables obtenidas de las trazas de los vuelos

Estos datos son aquellos que hemos obtenido en función de cálculos matemáticos aplicados a los datos obtenidos de las bases de datos.

- vx_a, vx_b: la velocidad en el eje x del avión en nudos justo al inicio de la pérdida de separación, obtenida a partir de los puntos de la traza.
- vy_a, vy_b: la velocidad en el eje y del avión en nudos justo al inicio de la pérdida de separación, obtenida a partir de los puntos de la traza
- vz_a, vz_b: la velocidad en el eje z (altura) del avión, en pies por minuto, justo al inicio de la pérdida de separación, obtenida a partir de los puntos de la traza.

- *convergence*: es una variable binaria que indica la geometría de la pérdida de separación en función de la convergencia o divergencia del vector velocidades del avión. Podemos ver un ejemplo en la Figura 3.1. Si la trayectoria converge, nos encontramos con un 1, y en caso contrario, con un 0. Puede afectar a la pérdida de separación ya que es plausible que los controladores estén más atentos a las trayectorias convergentes, de mayor peligro.

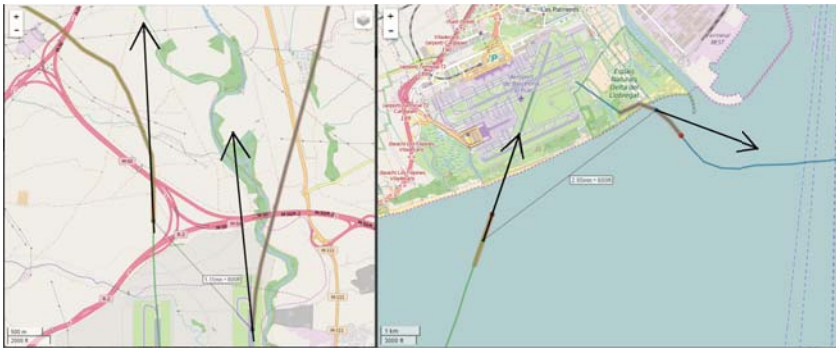

Figura 3.1: A la izquierda, ejemplo de trayectorias convergentes, a la derecha, trayectorias divergentes.

- *attitude*: la fase de vuelo en la que se encuentra el avión, es decir, si está en despegue (CLIMB), crucero (CRUISE) o aterrizaje (DESCEND), que son las tres posibles fases en ruta. Se calculan a partir de los instantes temporales de *top of climb* y *top of descent* que a su vez han sido obtenidos con un algoritmo que busca cambios bruscos en la traza. Aunque aparentemente es algo dependiente directamente de la velocidad en el eje vertical, la principal dificultad reside en el hecho de que los vuelos pueden realizar cambios de nivel en medio de la fase de crucero, es decir, ascender o descender sin aterrizar o despegar.

3.1.5. Variables meteorológicas

Las condiciones meteorológicas son determinantes en la existencia de cualquier complicación en un vuelo, y lo mismo ocurre con las pérdidas de separación. Especialmente las tormentas, los fuertes vientos y la posibilidad de turbulencias, pueden dar lugar a cambios de rumbo problemáticos.

Las fuentes de datos meteorológicos más utilizadas en el campo de la aeronáutica en España son METAR y SIGMET, pero la primera sólo devuelve información meteorológica en aeropuertos (y para este estudio sólo consideramos ruta) y la segunda son partes meteorológicos, predicciones para el plan de vuelo y no medidas.

La única fuente de datos disponible que devuelva información meteorológica real en un punto cualquiera del espacio es la propia del modelo GFS (Global Forecast System) creado por la agencia estadounidense NOAA. Este modelo incluye docenas de variables terrestres y atmosféricas, desde relacionadas con temperaturas, vientos y precipitaciones hasta concentración de ozono o humedad del suelo.

Este modelo envuelve la tierra en una malla con 0.5 grados de latitud y longitud de separación y devuelve medidas para cada intersección en la malla, dando lugar así a una resolución de 18 millas en las medidas. Existen también medidas en función de la capa isobárica, es decir, de la altura. Es un modelo ampliamente utilizado que no sólo devuelve medidas sino también predicciones, todo ello cada 3 horas, mientras que las medidas reales son cada 6 horas. Un ejemplo de consulta base de datos se encuentra, de nuevo, en el Apéndice B.

Para este problema, no se han considerado todas las variables disponibles en GFS. Reducidas inicialmente a 30 en la propia base de datos consultada, se escogieron posteriormente sólo aquellas relacionadas con fenómenos meteorológicos que pudieran afectar a la altura en la que se encuentra el avión. Estas son:

- *Absolute_vorticity_isobaric*: relacionada con tormentas y turbulencias.
- *Cloud_mixing_ratio_isobaric*: cantidad de agua presente en las nubes de la capa isobárica en la que se encuentra el avión.
- *Convective_available_potential_energy_surface*: uno de los índices de inestabilidad más populares, la CAPE (*Convective Available Potencial Energy*, en inglés) o energía potencial disponible convectiva. Se mide en Julios por kilogramos (J/kg).
- *Per_cent_frozen_precipitation_surface*: porcentaje de precipitación congelada, relativo a la posibilidad de granizo.
- *Pressure_maximum_wind*: presión en el máximo nivel de viento, medida en Pascales.

- *Relative_humidity_isobaric*: porcentaje de humedad en una capa isobárica dada.
- *Temperature_isobaric*: temperatura en una capa isobárica dada, medida en Kelvins.
- $Wind\_speed\_gust\_surface$: rachas de aire en la tierra medidas en metros por segundo.
- $u - component\_of\_wind\_isobaric$: componente este del viento en una capa isobárica dada en m/s.
- $v - component\_of\_wind\_isobaric$: componente norte del viento en una capa isobárica dada en m/s.

3.2. Preprocesamiento de los datos

3.2.1. Estudio de datos ausentes

Los Cuadros 3.1 y 3.2 muestran el número de datos ausentes encontrados en cada variable del conjunto de datos. Las variables relacionadas directamente con la pérdida de separación, al ser los resultados de un algoritmo ejecutado exclusivamente con el propósito de su obtención a partir de datos limpios, carecen de valores ausentes.

En los datos de información relativa a los vuelos provienen de fuentes ya existentes nos encontramos con que carecen de información para 50 vuelos. Esto ocurre porque no aparece información sobre los mismos en las bases de datos, muy probablemente porque son vuelos de poco interés, avionetas privadas o militares, con errores en los datos, como vemos, por ejemplo, en la traza de uno de ellos que muestra la Figura 3.2.

Precisamente por lo poco fiable de la calidad de la traza asociada a estos vuelos de bajo perfil, se ha decidido eliminar los registros completos asociados, ya que podría no existir una pérdida de separación real.

Dos registros han sido considerados excepciones y sus valores han sido encontrados manualmente (son todo variables de tipo categórico). Las razones por las

Figura 3.2: Ejemplo traza de vuelo con datos ausentes

cuales han sido exentos de eliminación son las siguientes: pertenecen a la clase con menos instancias, esto es, lo casos que no han llegado a ser pérdidas de separación; el vuelo con el que han tenido el conflicto es un vuelo comercial y, por último, no presentan errores en su traza.

Los casos de *occupancy* (2 casos) y velocidades (35+39 casos) desconocidas, es decir, variables numéricas obtenidas de las trazas, serán resueltos utilizando como valor de los mismos la mediana de los demás, ya que son registros que no por ello pierden interés, y dada la cantidad no se puede hacer una búsqueda manual. Se ha elegido la mediana y no la media porque la media no produce valores plausibles. Por ejemplo, para velocidades en el eje z, el avión lleva o una velocidad superior a 1000 pies por minuto o 0, los 11 pies por minuto que indica la media son un valor extraño.

Finalmente, nos encontramos con 127 registros en los que por una razón desconocida no existe información atmosférica en la base de datos para esas ubicaciones e instantes determinados (todos se encuentran en los mismos dos días). De nuevo, al ser una cantidad tan importante de datos, y tan cercanos temporalmente, en lugar de eliminarlos se decide sustituir los valores desconocidos por la mediana de las variables.

Cuadro 3.1: Datos ausentes en el conjunto de datos inicial (1)

Variable	Número de NA
id	0
flight_a	0
flight_b	0
startime	0
start_sep_h	0
start_sep_v	0
start_sep_3d	0
start_lat_a	0
start_lng_a	0
start_fl_a	0
start_lat_b	0
start_lng_b	0
start_fl_b	0
endtime	0
min_p_t	0
min_p_sep_h	0
min_p_sep_v	0
min_p_sep_3d	0
min_p_lat_a	0
min_p_lng_a	0
min_p_fl_a	0
min_p_lat_b	0
min_p_lng_b	0
min_p_fl_b	0
min_p_severity	0
sector	0
atc_unit	0
sectorization	0
convergence	0
callsign1	8
company1	8
adep_id	8
ades_id	8
wake	8
aircraft	8
enginetype	8

Cuadro 3.2: Datos ausentes en el conjunto de datos inicial (2)

Variable	Número de NA
callsign2	42
company2	43
adep_id2	42
ades_id2	42
aircraft2	42
wake2	45
enginetype2	45
occupancy	2
Capacity	0
occu_by_capacity	2
vx_a	35
vy_a	35
vz_a	36
attitude_A	0
vx_b	39
vy_b	39
vz_b	39
attitude_B	0
Absolute_vorticity	127
Cloud_mixing_ratio	127
Cloud_water	127
CAPE	127
Geopotential_height	127
Ice_cover	127
Land_cover	127
Per_cent_frozen_precip	127
Precipitable_water	127
Pressure_maximum_wind	127
Pressure_reduced_to_MSL	127
Relative_humidity	127
Temperature_isobaric	127
Wind_speed_gust_surface	127
u_component_of_wind	127
v_component_of_wind	127
CLASS	0
CLASS2	0

3.2.2. Estudio de datos atípicos

Para cada una de las variables del conjunto de datos, se realizó un estudio de sus valores utilizando diagramas de cajas. Los diagramas de cajas, también conocidos como *diagramas de cajas y bigotes*, representan la distribución de los valores de una variable utilizando los cuatro cuartiles en los que se divide: los valores entre

Q1 y Q3 están en las cajas cuya separación es la mediana, mientras que los bigotes encuadran los datos que están dentro de 1.5 veces el rango intercuartílico desde la mediana. Los valores exteriores se consideran valores atípicos o *outliers*.

Muchos métodos estadísticos, y por tanto la mayor parte de los algoritmos de aprendizaje automático son sensibles ante la aparición de este tipo de valores. Existen multitud de maneras de tratar con los valores atípicos en los conjuntos de datos [49] y con su error asociado, pero en este caso se ha decido simplemente mantenerlos o eliminarlos en función de un estudio individualizado documentado.

Para las variables en las que hay valores atípicos, se realizó también un histograma para investigar si era debido a una característica propia de la distribución, como por ejemplo, carácter multimodal, si eran valores atípicos, o si eran realmente anomalías.

Dentro de las propias anomalías, se ha decidido conservar aquellas que tienen sentido dada la naturaleza de los datos, como puede ocurrir con los valores muy extremos de velocidades cuando los vuelos no estaban en fase de crucero, y eliminarlos (junto con su fila correspondiente) cuando no se ha encontrado una explicación plausible de su existencia y es probable que se deban a errores de medida o en la recolección de datos.

Por otro lado, si suponían más de un 3 % del total de instancias, se han decidido mantener siempre para evitar una gran pérdida de datos.

Este estudio de valores atípicos se realizó utilizando el lenguaje de programación *R* y sus implementaciones predeterminadas de metodología estadística con la interfaz *RStudio*. Las gráficas y los resultados, para la variables originalmente continuas, se encuentran en el Apéndice A. Posibles valores atípicos de las variables categóricas se identificaron con la función *count* de R y conocimiento experto.

3.2.3. Selección de variables

Sobre el conjunto de datos anterior, previamente solucionados los datos ausentes y retirados los datos atípicos no deseados, se quiere reducir el número de variables implicadas, ya que el tan alto número del conjunto de datos descriptivos completo supone un problema a la hora de ejecutar algoritmos de aprendizaje automático así cómo a la hora de interpretar los resultados.

Matriz de correlaciones

El primer paso en la selección de variables consistió en calcular con R la matriz de correlaciones (de dimensiones 66x66), y, de entre ellas, señalar aquellas que superaran un coeficiente de correlación de 0.8. Los pares afectados fueron:

- Todas las variables relacionadas con el punto de mayor cercanía (las que comienzan por *min_p*) con respecto a su equivalente en el comienzo de la separación (*start*), y lo mismo con las del fin de la separación (comienzan por *end*). Se decidió mantener las del momento inicial puesto que se pretende idealmente capacidad de previsión, y crear una variable duración.
- Todas las variables de distancias diagonales con respecto a las distancias en x e y. Las diagonales fueron eliminadas.
- La capacidad del sector (*Capacity*) respecto a la ocupación por capacidad (*occu_by_capacity*), en las que claramente una es dependiente de la otra. Se decidió eliminar la capacidad por su inferior nivel de significado.
- Las posiciones para el avión A respecto a las del avión B. En lugar de mantener una de las dos, se decidió mantener el punto intermedio.
- Las estelas de los aviones (*wake1* y 2) respecto al modelo de avión. Los modelos de avión fueron mantenidos por su mayor importancia, calculada en el siguiente paso.

Aplicación del algoritmo *Boruta*

En el segundo paso, se aplica un algoritmo de selección de variables o *feature subset selection*. El algoritmo elegido es *Boruta* [35], implementado en el paquete de *R* del mismo nombre. Se ha escogido este algoritmo porque sigue un enfoque *all-relevant* en oposición al *minimal-optimal* habitual, es decir, porque en lugar de buscar el subconjunto de variables *mínimo* que devuelve unos resultados óptimos, busca todas las variables que pueden resultar relevantes.

Como en este trabajo se pretende buscar precursores de las pérdidas de separación, queremos mantener todas aquellas que sean mínimamente interesantes. Así pues, Boruta es un algoritmo tipo *wrapper* en lugar de uno tipo filtro, que

utiliza el clasificador de bosque aleatorio (*random forest*) como caja negra para devolver rangos para cada variable tras muchas iteraciones del mismo en diferentes conjuntos.

De los resultados obtenidos se extrajeron las siguientes conclusiones:

- Casi todas las variables que quedaban en el algoritmo, salvo *convergence*, resultaron relevantes.
- Existían variables con excesiva importancia: las separaciones iniciales, las variables de comienzo y fin y la duración de la separación. Todas estas variables tienen una relación directa y trivial con el hecho de que la aproximación sea o no pérdida de separación, ya que son descriptivas del hecho. Teniendo en cuenta que se buscan precursores que no tengan en cuenta las trayectorias, fueron eliminadas. Se trató de aunar las variables temporales en una de duración, pero ésta mantenía el mismo problema. Véase la Figura 3.3. En verde, las variables consideradas relevantes.

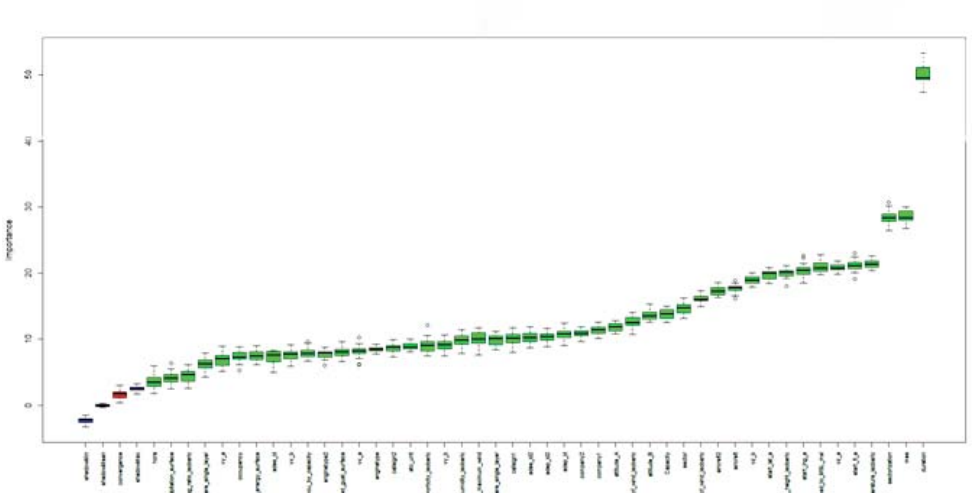

Figura 3.3: Resultado de aplicar el algoritmo Boruta al conjunto de datos con la variable *duration*.

- Después de eso, las más relevantes fueron (véase la Figura 3.4): la sectorización, las velocidades verticales, la temperatura y la presión reducida al nivel del mar. Además, las *wake* aparecieron como variables de importancia dudosa (en amarillo), y también fueron eliminadas.

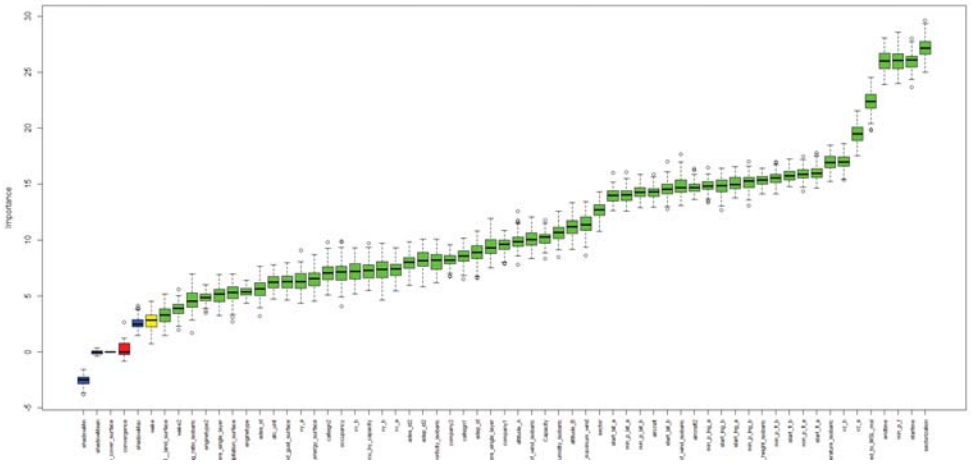

Figura 3.4: Resultado de aplicar el algoritmo Boruta al conjunto de datos reducido.

Selección manual

Tras las operaciones anteriores, el conjunto de datos seguía teniendo una cantidad de atributos demasiado grande para que fuera tratable e interpretable, y para que algoritmos de aprendizaje automático devolvieran buenos resultados. Así pues, se decidió continuar cambiando y eliminando algunas variables más que no fueran excesivamente significativas, utilizando las importancias de cada variable devueltas por Boruta y la opinión experta:

- Para poder tratar con la fecha de comienzo, ésta fue dividida en las variables categóricas del mes y la hora.
- Las variables que tenían componentes en x e y o en u y v (velocidades y viento) se unieron dando lugar a su variable módulo correspondiente, tanto por reducción de variables como por significación. Asimismo, las variables de latitudes y longitudes de la posición de cada avión se resumen en una para latitud y otra para longitud utilizando el punto medio de ambos.
- Además, se eliminaron, por conocimiento experto y por su baja posición en el gráfico de importancias, las siguientes variables: *ice_cover_surface*, *land_cover_surface*, *wind_speed_gust_surface*, *cloud_mixing_ratio_isobaric*, *frozen_precipitation*. Las tres primeras variables hablan de la superficie terrestre y no

afectan directamente a un vuelo, *cloud_mixing* tiene el mismo significado que la variable mantenida *cloud_water*, y *frozen_precipitation* tiene valor 0 en más del 80 % de los casos.

3.2.4. Discretización

Dada la dificultad del tratamiento de conjuntos de datos híbridos (con variables continuas y categóricas), y la imposibilidad directa de su realización con la herramienta GENIE, se procede a la discretización de las variables continuas. La discretización es un proceso matemático mediante el cual los valores continuos se incluyen en *bins* o depósitos para que haya un número limitado de estados posibles, es decir, se convierten en variables categóricas discretas. Existen multitud de métodos de discretización y la bondad de sus resultados depende del método elegido y de la naturaleza de los datos. Una comparación de los mismos se encuentra en [23].

La herramienta GENIE incorpora funcionalidades de tratamiento de datos, entre ellas discretización. Aunque sólo implementa los tres métodos más sencillos, se decidió su uso por su simplicidad y conexión con los algoritmos de aprendizaje. Los métodos que incluye son: *igual anchura* (crea depósitos que son intervalos con la misma distancia), *igual frecuencia* (crea depósitos con el mismo número de elementos), y *clústers* (que agrupa los puntos en clústers).

Para conseguir que las diferencias entre los valores significativos de las variables se tuvieran en cuenta sin utilizar una gran cantidad de depósitos, se decidió utilizar el *método de igual frecuencia*, con modificaciones manuales para que los depósitos tuvieran sentido o agrupar valores de interés, como pueden ser los 0. El máximo número de depósitos con el que se consiguió que funcionaran los algoritmos de aprendizaje fue 4.

Reducción del dominio

El tamaño de las tablas de probabilidades condicionales a *priori* crece exponencialmente con el número de estados de los nodos en los que se definen. Por ello, el gran número de estados que contenían las variables *company_a, company_b, aircraft_a, aircraft_b* y *sectorization*, algunas de ellas por encima de 100 estados,

necesitaba ser reducido. Se decidió eliminar algunas filas que contenían estados de los que sólo existía un único caso y, además, reagrupar semánticamente los estados en un número inferior de ellos.

Por ejemplo, los modelos de avión se redujeron a familias de modelos, como los estados A318, A319, A320 y A321, que fueron reducidos al estado A320s. Las familias de modelos de avión poseen características muy similares salvo por detalles como la disposición de los asientos, lo cual se consideró suficientemente significativo en su semántica como para utilizarlo en los estados, eliminando así más de 100 estados en cada una de las dos variables.

3.3. Descripción del conjunto final de datos

El conjunto final de datos, tras la eliminación de instancias con datos ausentes y atípicos, consta de 3845 filas, 1813 aproximaciones sin pérdida de separación y 2032 pérdidas de separación. Tiene 32 atributos, de los cuales originalmente 16 eran continuos.

Entre esas 32 variables, nos encontramos con 9 variables relacionadas con el tiempo atmosférico, 2 variables relacionadas con el tráfico en el sector, 7 variables obtenidas del estudio de las trayectorias de los vuelos, 7 variables descriptivas de la pérdida de separación por ubicación y tiempo y 6 variables descriptivas de las aeronaves.

En el Apéndice C se puede encontrar también un resumen estadístico de las variables. El Cuadro 3.3 muestra las columnas o atributos del conjunto final de datos junto con su número de estados tras la realización completa del preprocesamiento de datos.

Cuadro 3.3: Número de estados de las variables en el conjunto final de datos.

Variables	**Número de valores posibles**
mes	6
hora	24
start_lat	(cont) [25.32, 44.92] → 4
start_lng	(cont) [-17.672, 4.66] → 4
start_fl	(cont) [250, 41675] → 4
atc_unit	6
sectorization	55
convergence	2
company_a	40
aircraft_a	40
enginetype_a	3
company_b	43
aircraft_b	33
enginetype_b	4
occupancy	(cont) [0, 26]
occu_by_capacity	(cont) [0, 0.722] → 4
v_horizontal_a	(cont) [40.7, 777.2] → 4
vz_a	(cont) [-5438, 5319] → 4
attitude_a	3
v_horizontal_b	(cont) [1.33 , 1042.4] → 4
vz_b	(cont) [-5225, 9119]
attitude_b	3
Absolute_vorticity_isobaric	(cont) [-0.00163, 0.00461] → 4
Cloud_water_entire_atmosphere	(cont) [0, 2.49] → 4
Geopotential_height_isobaric	(cont) [134.3, 13812.9] → 4
Precipitable_water_entire_atmosphere	(cont) [1.7, 34.7] → 4
Pressure_maximum_wind	(cont) [10733, 49970] → 4
Pressure_reduced_to_MSL_msl	(cont) [99181, 103989] → 4
Relative_humidity_isobaric	(cont) [0.00,100.00]→ 4
Temperature_isobaric	(cont) [205.2, 292.7] → 4
Total_wind_isobaric	(cont) [0.3265, 74.2114] → 4
CLASS	2

Capítulo 4

Herramientas utilizadas

4.1. PERSEO

El proyecto PERSEO (Plataforma de análisis de Efectos de Red de Sectorización En Operación) [14] es una aplicación *web* que muestra información operativa, históricos, análisis estadísticos cruzados y predicciones, de diversas naturalezas relacionadas con navegación aérea. Fue desarrollado para ENAIRE, la división de navegación aérea de Aena, por parte de CRIDA A.I.E., empresa que también se encarga de su mejora continua y mantenimiento, y es de uso únicamente interno. Se pretendía además que tuviera un carácter accesible para usuarios de diferentes niveles de conocimiento, y es accesible desde cualquier dispositivo en perfecto estado de actualización gracias a su carácter *web*.

Entre otras utilidades, proporciona estudios sobre el funcionamiento de aeropuertos, vuelos y planes de vuelo, carga de controlador, configuraciones existentes y previstas, el *índice de rendimiento de configuraciones operativas* (IRCO), informes de torre, cálculos de combustibles y emisiones, y pérdidas de separación. Todo ello con diferentes personalizaciones y filtros, con información desde hace un lustro hasta en tiempo real.

La existencia de la aplicación se remonta a casi diez años atrás, pero la versión actual utiliza el *framework* código abierto Vaadin [25] para aplicaciones web en el lenguaje de programación Java.

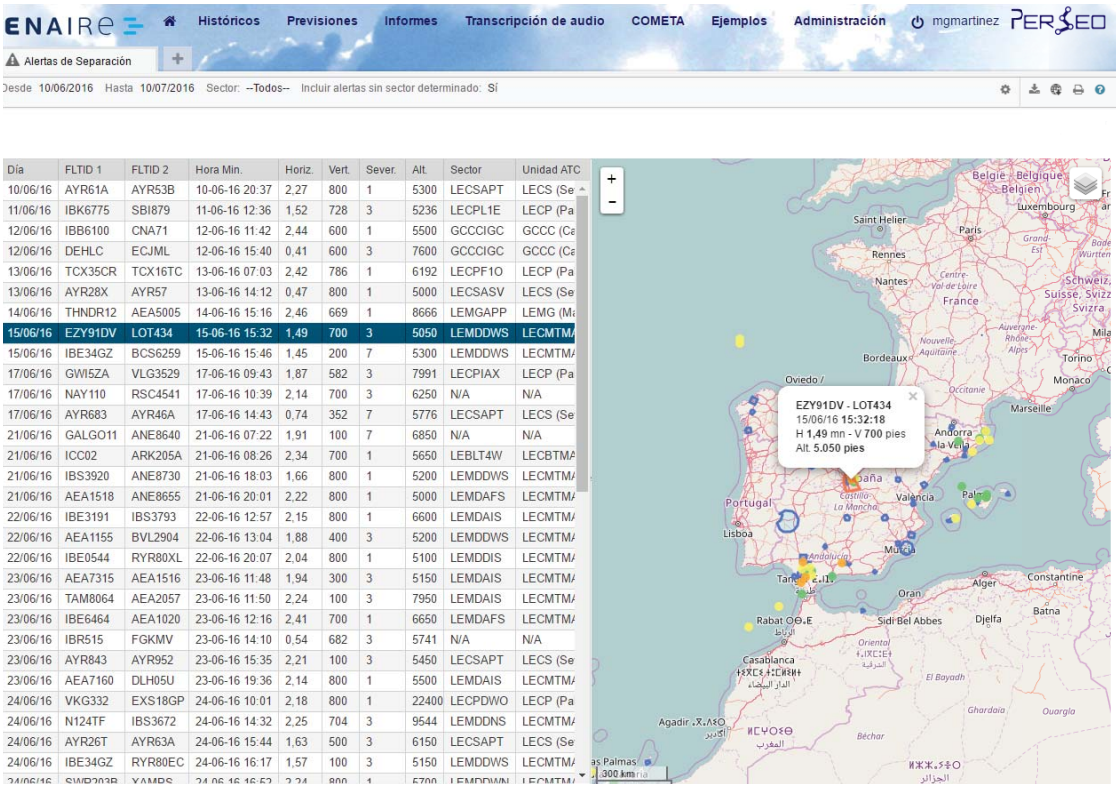

Figura 4.1: Captura de pantalla del uso del apartado de pérdidas de separación en la aplicación de PERSEO.

Está respaldado por una base de datos de gran tamaño, que unifica docenas de fuentes de información diferentes. Algunas de estas son: EUROCONTROL (archivos diarios con información de vuelos europeos), *Aeronautical Information Regulation And Control* AIRAC (sectores, procedimientos y capacidades, mensualmente), IFS (trazas radar sin suavizar con puntos cada 5 segundos), IPV y GIPV (planes de vuelo españoles recogidos por la plataforma PALESTRA), INSIGNIA (información sobre aeropuertos y el espacio aéreo español desde SACTA), METAR (meteorología en aeropuertos).

En cuanto a su funcionamiento interno (el cual muestra la Figura 4.2, cuenta con tres partes diferenciadas pero interconectadas:

- *PERSEO Analysis*: en este proyecto se encuentran los algoritmos de análisis que calculan métricas y estadísticas que se almacenan en la base de datos

propia de PERSEO (de tipo MySQL). Entre ellos, un analizador de separaciones, que compara a pares todas las trazas de vuelos en una ventana espacio temporal. Es la parte que ha sido utilizada y ampliada para la realización de este trabajo. Como son procesos muy lentos (el procesamiento de los 6 meses que han sido estudiados para este trabajo tardó una semana), no pueden ser realizados a petición del usuario de la aplicación, sino que las bases de datos se van actualizando diariamente de forma *offline*, si existen datos.

- *PERSEO Core*: este proyecto actúa de intermediario entre los otros siguiendo un patrón clásico de *software* y en él se almacenan los modelos y los procesos de acceso a las bases de datos, para lo cual se utilizan los *frameworks* JPA e Hibernate.

- *PERSEO Web*: es la aplicación *web* en sí, accede a las bases de datos para obtener los datos procesador por análisis y se los muestra a los usuarios en su aplicación.

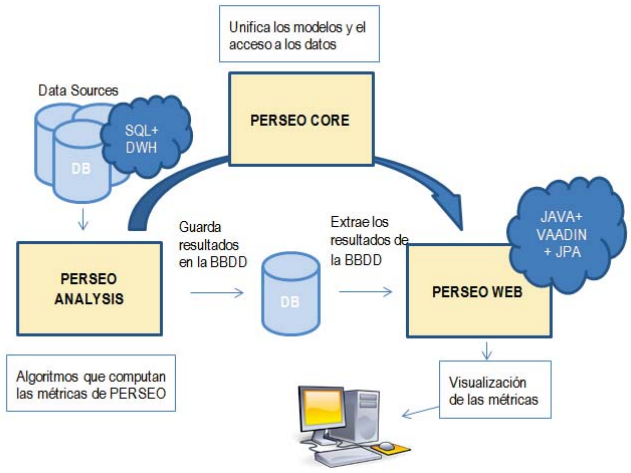

Figura 4.2: Funcionamiento interno de la aplicación web de análisis en navegación aérea PERSEO.

4.2. *Talend Open Studio*

Talend Open Studio for Data Integration [59] es una herramienta de código abierto desarrollada en 2006 por la empresa americana Talend que incrementa la eficacia en la integración de datos en un práctico entorno de trabajo dedicado. Está diseñada para combinar, convertir y actualizar datos de diferentes localizaciones, principalmente orientada al ámbito de los negocios. Las principales funciones de la aplicación son la sincronización de bases de datos, los intercambios de datos por lotes entre los sistemas de la infraestructura, la migración de datos, y la transformación y carga de datos complejos.

Posee una cómoda interfaz, en la que cada programa es llamado un *job*, que se basa en programación por componentes, es decir, en ir uniendo diferentes componentes con funcionalidades diversas mediante sus flujos de entrada y flujos de salida para realizar una tarea más compleja. Internamente funciona como un generador de código Java. Es considerado un ETL, esto es, el acrónimo de *Extract, Transform* y *Load* (en español extraer, transformar y cargar) y hace referencia al proceso que permite obtener la información de una fuente de datos, procesarla, formatearla, limpiarla y cargarla en otra.

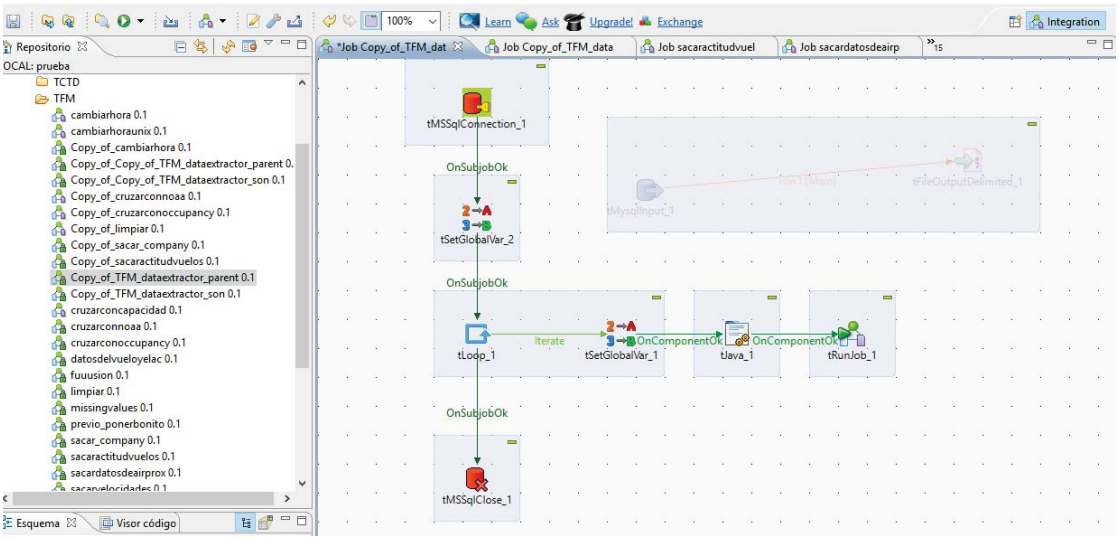

Figura 4.3: Captura de pantalla del proceso de trabajo en *Talend Open Studio*

Se ha utilizado esta herramienta para la extracción de información de las dife-

rentes bases de datos utilizadas, para las cuales a veces era necesario la realización de varias consultas a bases de datos que invocaban a otras, o la realización de grandes peticiones a bases de datos dividas y en bucle. Asimismo, también para cruzar los datos obtenidos (solucionando problemas como fechas incoherentes o en diferentes husos horarios, duplicados, trazas anómalas), limpiarlos y volverlos a cargar cuando era necesario. En la Figura 4.3 se muestra un ejemplo de uno de los *jobs* realizados para este trabajo, y a la izquierda el explorador con el total de los mismos.

4.3. Aplicaciones de bases de datos

Aunque a menudo para la extracción de datos se accedió a las bases de datos desde *Talend Open Studio*, también fueron utilizados los entornos propios de las mismas, especialmente para visualización y pruebas.

4.3.1. MySQL Workbench

MySQL Workbench [10] es una herramienta gráfica integrada para arquitectos de bases de datos en MySQL, desarrolladores y usuarios. Provee modelado de datos, desarrollo, migraciones, administración de usuarios y muchas otras características desde un entorno sencillo y fácil de montar.

En este trabajo se utilizó para el acceso a la base de datos interna de la aplicación PERSEO.

4.3.2. SQL Server Management Studio

SQL Server Management Studio (SSMS) [40] es la aplicación *software* utilizada desde 2005 para la configuración, manejo y administración de Microsoft SQL Server creada por el propio Microsoft. Incluye editores de código y herramientas gráficas. Posee más características y está mejor optimizada que su correspondiente en MySQL, aunque su funcionamiento y montaje es más complejo.

Podemos encontrar ejemplos de consulta en el Apéndice B.

4.4. RStudio

RStudio [30] es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R que contiene una consola con editor de sintaxis y otras herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

R [60] es un lenguaje y entorno de programación para análisis estadístico y gráfico, parte del sistema GNU y de tipo interpretado. Es un dialecto de código libre del lenguaje S, desarrollado por Robert Gentleman y Ross Ihaka en la Universidad de Auckland en 1993. Al estar orientado a la estadística, proporciona un amplio abanico de herramientas para ello. Entre otras características destacables de R, podemos nombrar su capacidad de generación de gráficos, de cálculo numérico y de minería de datos. Respecto a sus aplicaciones, es uno de los lenguajes más utilizados en investigación por la comunidad estadística, siendo además muy popular en el campo del análisis de datos, la investigación biomédica, la bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y gráficas.

4.5. GeNIe

GeNIe Modeler [15] es una interfaz gráfica de usuario (GUI) para el motor SMILE que permite aprendizaje y modelado interactivo de redes bayesianas, diseñada para Windows por BAYESFUSION, LLC. Actualmente de caracter comercial, lleva siendo utilizada y probada desde 1999 y cuenta con un alto grado de aceptación en la industria y en el ámbito investigador.

Entre sus características principales encontramos: dispone de un editor gráfico de redes, soporta nodos con diferentes modelos de independencia en las tablas de probabilidad condicionada, como *noisy-or* y *noisy max*, permite copiar, corta y pegar diferentes partes de redes entre diferentes redes, está integrado con muchos tipos de archivos de datos, provee también soporte para los costes de observación y diagnóstico.

Implementa además varios algoritmos de aprendizaje de redes bayesianas a partir de datos. Ninguno de ellos soporta una mezcla de variables continuas y discretas, así que si existe una variable discreta en el conjunto de datos, es necesario discretizar el resto. Cuando el algoritmo no devuelve los parámetros de

la estructura, GeNIe dispone de un algoritmo de aprendizaje de parámetros con *expectation–maximization* (EM). Los algoritmos de aprendizaje de la estructura disponibles son los siguientes:

- *Bayesian Search* es un método clásico de aprendizaje basado en *score*, similar al de [9], que utiliza como método de búsqueda *hill-climbing* con reinicios aleatorios, y como puntuación, BDeu.
- El algoritmo PC [58], explicado en el estado del arte, es un algoritmo de aprendizaje basado en tests de independencia condicional, y también uno de los métodos más tempranos y utilizados. Puede manejar conjuntos de datos continuos.
- El algoritmo *Essential Graph Search*, un método híbrido que realiza una búsqueda de grafos esenciales (grafos de redes bayesianas con arcos dirigidos y no dirigidos en los que los arcos dirigidos corresponden a los arcos cuya orientación no puede cambiarse sin cambiar las probabilidades codificadas) con PC y sobre ellos utiliza *Bayesian Search*.
- *Greedy Thick Thinning (GTT)* [7] es otro método de aprendizaje basado en *score* que busca en el espacio de clases de equivalencia, mejorando así la complejidad algorítmica del *Bayesian Search*. Permite el uso de las puntuaciones K2 y BDeu descritas en el estado del arte.
- GeNIe también permite el uso de algoritmos de aprendizaje ingenuos, como *Naive Bayes* y *Tree Augmented Naive Bayes* (TAN), ya descritos. El motor del programa utiliza un algoritmo similar al *Bayesian Search* para su aprendizaje.

Capítulo 5

Análisis y resultados

5.1. Análisis

Las redes bayesianas se consideraron un enfoque apropiado por su expresiva representación gráfica, adecuada para un estudio que busca explicaciones más allá de la predicción. Al ser la única herramienta sin coste capaz de mostrar el grafo de una red con tantas variables como con la que estamos tratando, pese a su menor flexibilidad y variedad algorítmica, se eligió la herramienta GeNIe para la aplicación de redes bayesianas.

Sobre el conjunto final de datos, se han aplicado desde GeNIe los siguientes métodos:

1. *Bayesian Search*, con los siguientes parámetros:

    - *Discrete threshold* (número de valores que tiene que tener una variable para ser considerada continua): El valor por defecto es 20. Se escogió 50, una cota superior del número de valores distintos en las variables discretas presentes.
    - *Max Parent Count* (número máximo de padres posibles para los nodos): Se mantuvo el valor por defecto, 8. En ningún caso superó algún nodo los 5 padres en la red resultante.
    - *Iterations* (número de reinicios del algoritmo): El valor por defecto es

20, se redujo a 5, número máximo que permitió que el programa no fallara.

- *Sample size* (tamaño de la muestra de instancias que toma el *score* BDeu): Se mantuvo el valor por defecto, 50.
- *Link Probability* (influencia la probabilidad de aparición de arcos en las redes iniciales): Se mantuvo el valor por defecto, 0.1, ya que era el mayor que permitía que el programa no fallara.
- *Prior Link Probability* (afecta a la distribución de probabilidad a priori, es decir, antes de los datos, del BDeu) Se mantuvo el valor por defecto, 0.001.
- *Max Time (s)* (tiempo máximo de ejecución del algoritmo, para evitar fallos) Se mantuve el valor por defecto, ilimitado.

2. *Greedy Thick Thinning* con *score* K2 y los siguientes parámetros:

   - *Discrete threshold*: 50.
   - *Max Parent Count*: 8. De nuevo, en ningún caso superó algún nodo los 5 padres en la red resultante.

3. *Greedy Thick Thinning* con *score* K2 y los parámetros anteriores utilizando conocimiento experto previo, con el cual se forzó la inclusión de arcos entre la clase y los perfiles de vuelo (*attitude_A*, *attitude_B*).

Los algoritmos *PC* y *Essential Graph Search* hacen fallar al programa con tiempo infinito, y limitando el tiempo al máximo posible (aproximadamente 200 segundos) devuelven una red casi completamente interconexionada.

5.1.1. Red bayesiana con *Bayesian Search*

Con sólo 30 arcos, esta red bayesiana indica que la variable sólo depende de la variable sectorización, e incluso ésta tiene también una muy pequeña red de nodos predecesores. Existen además variables sin aparente conexión con ninguna otra, como *convergence*, *hora* y la pareja de *occupancy* y *occu_by_capacity*. Produce además algunas relaciones entre variables que, desde un conocimiento experto, no tiene sentido que existan, como entre el modelo de avión y la unidad ATC, sin ser

además simétrica con el otro modelo de avión. La red se muestra en la Figura 5.1. En amarillo el nodo clase. A la derecha los parámetros elegidos y resultados de la ejecución.

5.1.2. Red bayesiana con *GTT*

La red bayesiana del problema obtenida con el algoritmo *Greedy Thick Thinning* parece, intuitivamente, modelar mejor el sistema que la anterior. Sus 31 nodos están unidos por 73 arcos. Hay un total de 322 estados, siendo la media de entradas 2.355 y de salidas 10.39. Nuestra variable clase se ve afectada por un total de 3 nodos padre (*sectorization*, *temperature_isobaric*, *mes*), y posee además 3 nodos hijos (*vz_a*, *vz_b*, *attitude_A*).

Como una de las cosas que se quería comprobar, por conocimiento experto, es que en ciertas actitudes de los aviones se producen más pérdidas de separación, se realizó otra red bayesiana que forzara a estos nodos a ser padres del nodo clase. Por lo demás muy parecida a la red previa, veremos en el siguiente apartado que su nivel de acierto resultó peor, por lo que nos centraremos en la red sin conocimiento previo.

Esta red bayesiana se muestra en la Figura 5.2. En amarillo el nodo clase, en verde los nodos de variables meteorológicas, en azul claro los descriptivos de la pérdida de separación, en azul oscuro los nodos relacionados con los vuelos y en naranja con el espacio aéreo.

5.2. Validación

Utilizaremos el método de validación cruzada o *cross-validation*, que divide en un número de particiones la muestra para luego utilizar cada uno de los posibles subconjuntos como datos de prueba. Implementado en el propio GeNIE, se eligió la utilización de la validación cruzada con 10 iteracones o *10-folds cross-validation* para estimar cómo de preciso es nuestro modelo de cara a predecir si una aproximación es o no una pérdida de separación, es decir, predecir la variable clase. También implementa el método *Leave one out*, validación cruzada con tantas particiones como instancias, que resultaba demasiado costoso dada la cantidad de instancias disponibles.

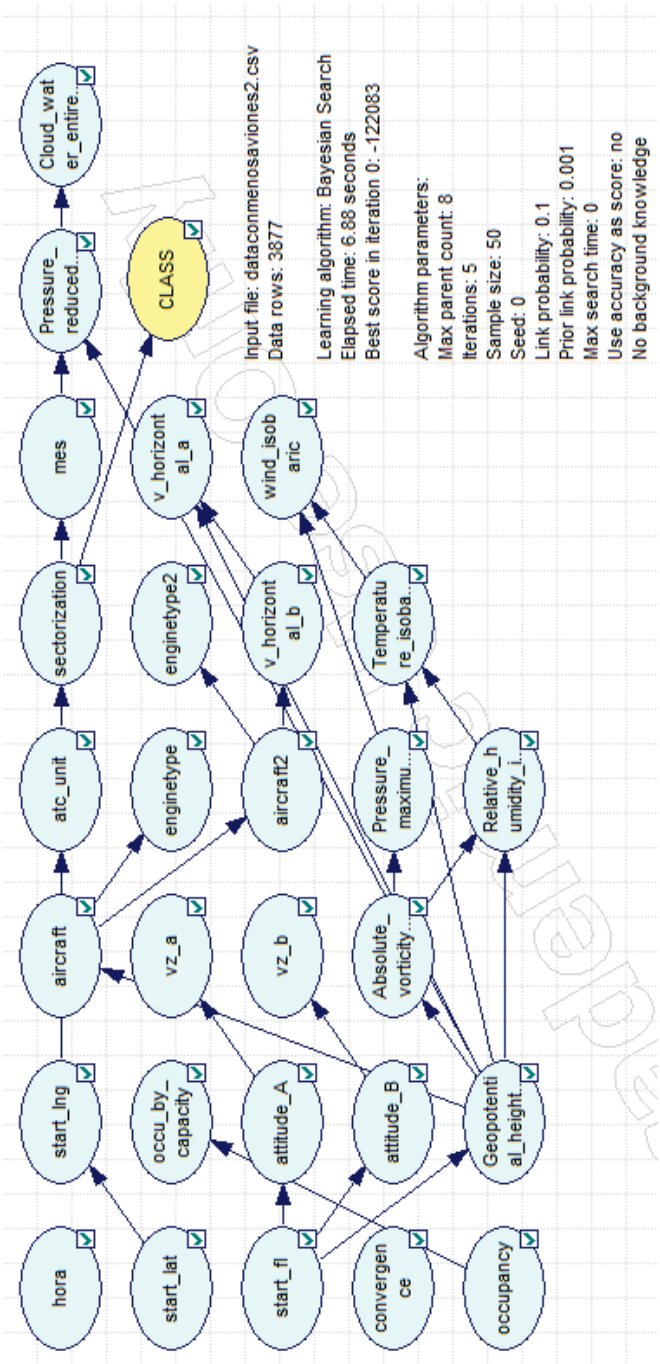

Figura 5.1: Red bayesiana con *GTT*.

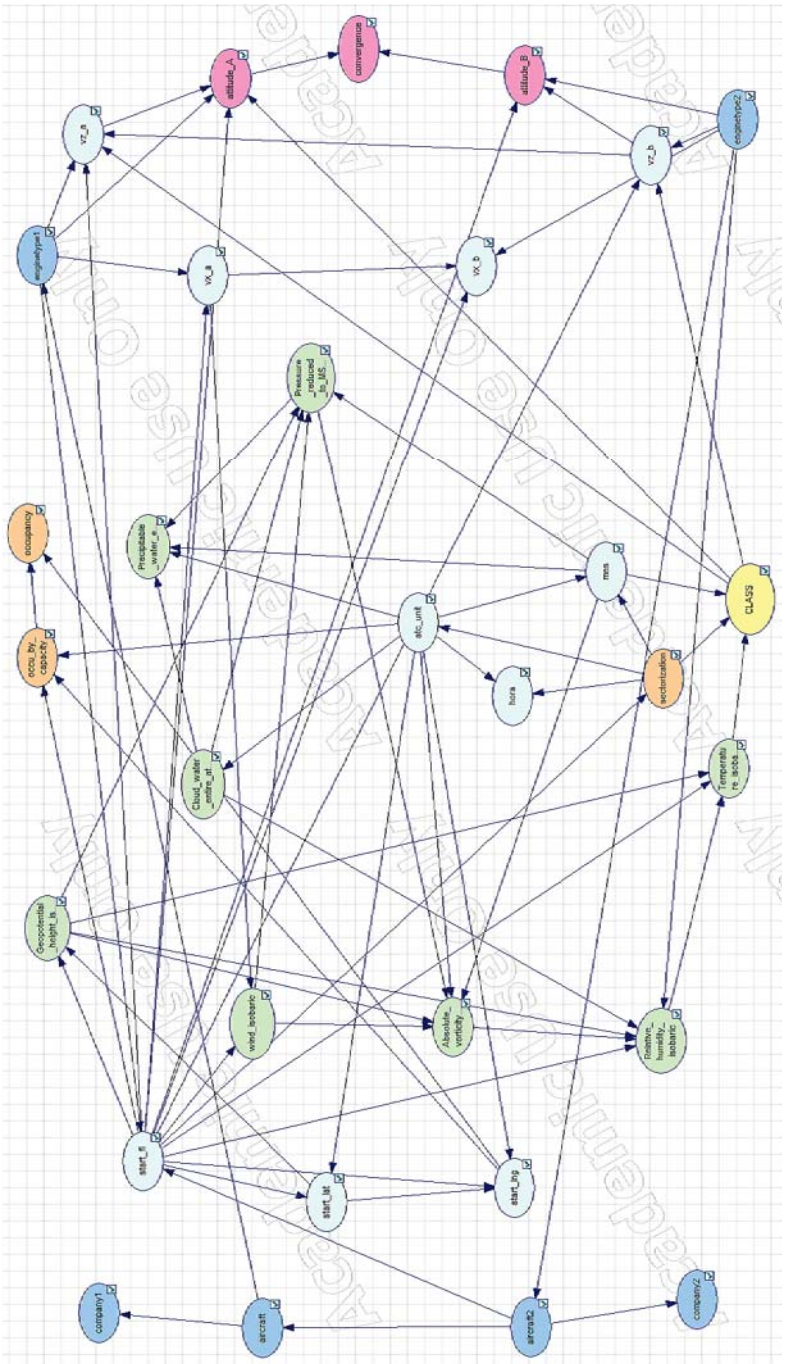

Figura 5.2: Red bayesiana creada con *GTT*.

La técnica de evaluación implementada en la herramienta mantiene la estructura de la red y re-aprende los parámetros del modelo en cada una de las iteraciones, por lo que debemos introducir también los parámetros del algoritmo EM que utiliza GeNIe para aprender los parámetros.

Se escoge el tipo *Uniformize* para comenzar la ejecución del EM con todos los parámetros cogidos de una distribución uniforme, sin utilizar así los parámetros previamente establecidos. El otro parámetro a rellenar es la confianza o *confidence*, también conocida como *equivalent sample size* (ESS), tamaño de muestra equivalente en castellano. Ésta se asigna a uno cuando se quieren obviar los parámetros existentes, ya que representa el número de instancias que dan lugar a los parámetros ya existentes.

Para poder realizar la validación sobre un modelo de carácter probabilístico, GeNIe elige para cada instancia el estado de la clase que es más probable sobre todos los otros estados.

Los Cuadros 5.1, 5.2 y 5.3 muestran la precisión del modelo:

Cuadro 5.1: Aciertos en la red bayesiana con GTT

CLASS = **0.770432** (2960/3842)
previolation = 0.726795 (1245/1713)
violation = 0.802255 (1708/2129)

Cuadro 5.2: Aciertos en la red bayesiana con BS

CLASS = **0.657725** (2550/3842)
previolation = 0.396183 (685/1714)
violation = 0.86825 (1865/2128)

Cuadro 5.3: Aciertos para la red bayesiana GTT con conocimiento previo

CLASS = **0.63899** (2455/3842)
previolation = 0.844717 (1447/1713)
violation = 0.473462 (1008/2129)

Así pues, el porcentaje más alto de aciertos obtenido es el de la red bayesiana obtenida con el algoritmo GTT y los parámetros establecidos en el apartado anterior, sin utilizar conocimiento previo. Con un 77 % de aciertos, supera ampliamente a los otros dos casos: al utilizar conocimiento experto, la red bayesiana que devuelve GTT consigue sólo un 63.9 % de acierto, mientras que la red que devuelve el algoritmo BS es también inferior al GTT simple, obteniendo un 65.8 % de aciertos.

Estas tablas de resultados también nos devuelven las sensibilidad y especificidad del modelo. Al ser una clase categórica de dos valores, se puede considerar una clase binaria en la que la existencia de pérdida de separación (*violation*) supone un positivo y la no existencia (*previolation*) un negativo.

La sensibilidad, también llamada ratio de positivos verdaderos (*true postive rate* (TP)), mide la proporción de positivos correctamente identificados como tales, mientras que la especificidad, ratio de negativos verdaderos o *true negative rate (TN)*, mide el porcentaje de negativos correctamente identificados.

Para el modelo GTT sin conocimiento previo, la sensibilidad es del 80.2 %, mientras que la especificidad es menor, con un 72 % de negativos identificados correctamente. Para el modelo BS, la diferencia entre ambos es aún mayor, la sensibilidad es muy alta, del 86.8 %, pero la especificidad es inferior a un lanzamiento de moneda, 39.6 %. Lo mismo pero a la inversa ocurre en el modelo GTT con conocimiento previo: tiene una especificidad del 84.5 % pero una sensibilidad del 47.3 %. La sensibilidad más alta la proporciona pues el modelo BS mientras que la especificidad más alta el GTT sin conocimiento previo.

Se podría aseverar que, al ser una asunto de seguridad, es más importante la sensibilidad, es decir, evitar falsos negativos, o, dicho coloquialmente, es mejor ser precavido en exceso. Sin embargo, dados los bajísimos niveles de especificidad, un sistema que con tanta facilidad prediga positivos no sería manejable.

Se escoge pues como mejor modelo considerado la red bayesiana obtenida con el algoritmo GTT, cuya validación continuaremos mostrando a continuación:

La *matriz de confusión* nos da un poco más de información sobre cómo se distribuye la precisión del modelo. Se muestra en la Figura 5.3. Una matriz de confusión es un tipo particular de tabla de contingencia que se utiliza para mostrar si el sistema está *confundiendo* clases, mostrando cuántas de un tipo han sido consideradas como de otro tipo. Al ser una clase bivaluada, es similar a los

resultados de precisión anteriores, pero podemos obtener alguna medida más de ella:

- P (número total de positivos): 2129 (0.55)
- N (número total de positivos): 1713 (0.45)
- TP (equivalente a sensibilidad): 1708 (0.8)
- TN (equivalente a especificidad): 1245 (0.73)
- FP (falsos positivos o falsas alarmas): 468
- FPR (ratio de falsos positivos, error de tipo I): 0.27
- FN (falsos negativos, fallos): 414
- FNR (ratio de falsos negativos, error de tipo II): 0.19
- Número total de aciertos: 2953

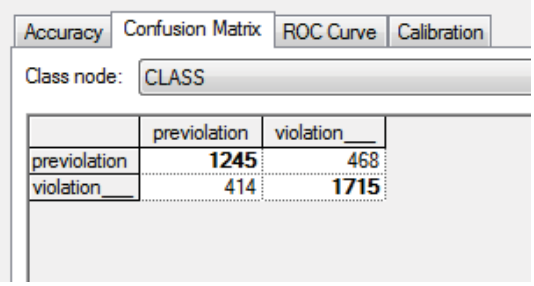

Figura 5.3: Matriz de confusión

Una curva ROC (*Receiver Operating Characteristic*, en español característica operativa del receptor) es una representación gráfica de la sensibilidad frente a la especificidad para una clasificación binaria según varía el umbral de discriminación [19], con origen en la teoría de señales.

Un clasificador discreto daría un único punto en el espacio ROC, pero en una red bayesiana la salida son valores de probabilidad que representan hasta qué punto una instancia pertenece a una de las dos clases, y es el valor umbral que determina

el límite entre una clase y otra lo que variamos para obtener diferentes puntos en el espacio ROC.

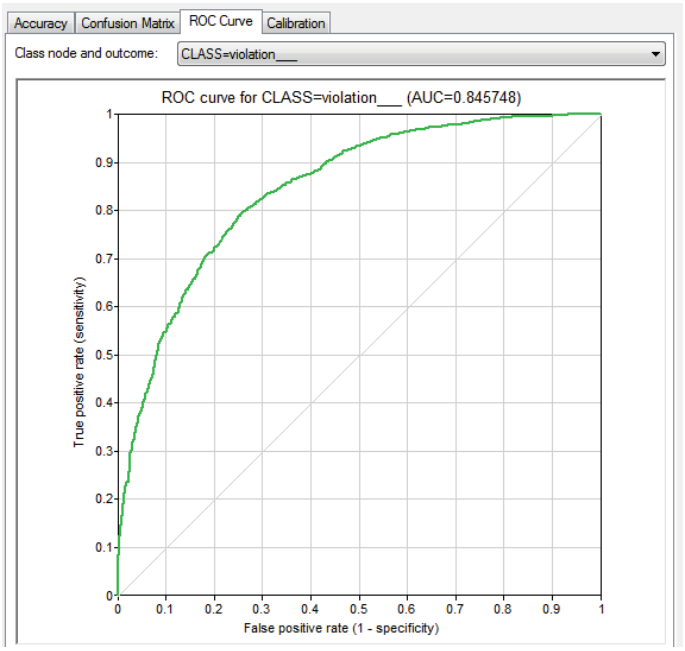

Figura 5.4: Curva ROC para la clase *violation*

La curva ROC obtenida del programa GeNIe para nuestra clase en cuestión se muestra en la Figura 5.4. De ésta, por su forma curva por encima de la línea de referencia, se puede considerar propia de un *test* bueno, aunque por supuesto mejorable (cuanto más cerca del uno llegue la curva, mejor será ésta).

El valor AUC, acrónimo de *area under the curve* o área bajo la curva es una métrica numérica de la bondad de la curva. En este caso, el valor es 0.85, lo cual está en un intervalo de valores por encima de regular, considerados buenos, aunque no excelentes [4]. Mencionar también que el AUC de los otros modelos no superaba el 0.7.

Otra importante medida del rendimiento de la clasificación realizada es la *curva de calibración* [63]. Ésta compara la probabilidad de salida respecto a las frecuencias observadas en los datos. Por cada probabilidad producida por el modelo, en el eje horizontal, la gráfica muestra las frecuencias reales en los datos, en el eje vertical, observadas para todos los casos para los que el modelo produjo dicha

probabilidad.

La línea diagonal muestra una calibración ideal. Como la probabilidad es una variable continua, la gráfica agrupa las probabilidades en depósitos de manera que haya suficientes instancias correspondientes. Una posible forma de hacer esto y la utilizada en la Figura 5.5 es a la manera de un histograma, en este caso con 10 depósitos. GeNIe también provee el método de *Moving average*, que utiliza una ventana deslizante considerando un cierto número de vecino, pero su interpretación es un poco más confusa.

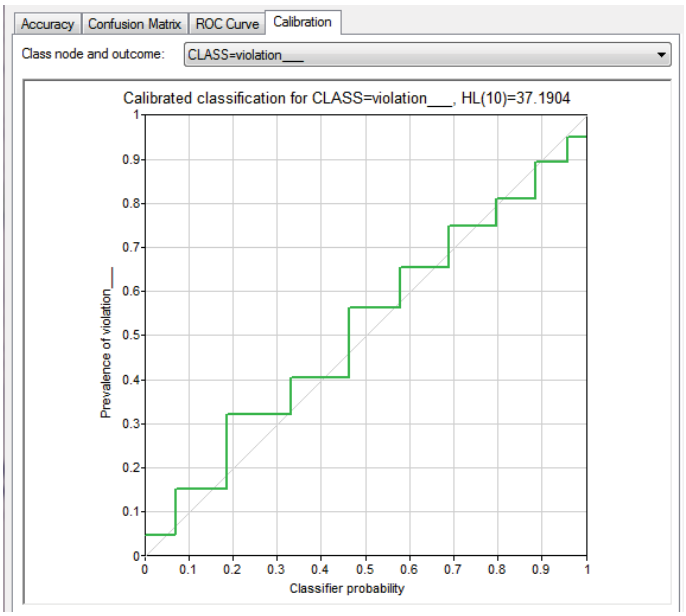

Figura 5.5: Curva de calibración para la clase *violation*

En la Figura 5.5 podemos ver los resultados de la gráfica de calibración para el estado *violation* de nuestra clase. Como vemos, la calibración puede ser considerada buena puesto que se mueve entorno a la línea ideal, ya que los valores de probabilidad se aproximan bastante a los de la frecuencia, especialmente para valores altos, con un error un poco mayor en torno a los valores entre 0.2 y 0.5.

Es también interesante mencionar los resultados de un superficial *análisis de sensibilidad* de la red bayesiana. Los análisis de sensibilidad son una técnica que ayuda a validar los parámetros de una red bayesiana, midiendo el efecto que pequeños cambios producen en probabilidades a *posteriori* para una serie de nodos

objetivo. La implementación de GeNIe es sencilla y basada en el cálculo de derivadas [33].

Los parámetros más sensibles tienen mayor efecto el resultado de la predicción y son por ello los que deben ser tratados con mayor precisión. Para la red bayesiana considerada, las variables más sensibles utilizando el nodo clase como único nodo objetivo son: *sectorization*, *aircraft* y *enginetype*, seguidas de *start_fl* y *mes*.

5.3. Interpretación de los resultados

Se ha obtenido pues una compleja red bayesiana para las variables asociadas a una aproximación entre aeronaves que predice si ésta va a llegar o no a convertirse en pérdida de separación con un 77 % de acierto en los datos estudiados.

Sin embargo, no toda la red parece tener relación con la clase, la cual sólo posee conexiones con otros 6 nodos, de un total de 31. Sabemos que por la condición de Markov, un nodo es condicionalmente independiente de todos sus no sucesores dados sus padres. No obstante, podemos especificar aún más: gracias al concepto de manto de Markov o *Markov's blanket* podemos llevar más allá las restricciones de independencia, ya que se puede decir que un nodo es condicionalmente independiente de todos los otros nodos dado su manto de Markov, esto es, sus padres, hijos y *esposas* (otros padres de hijos comunes).

En la Figura 5.6 se muestra el manto de Markov de la variable clase. Se representa en amarillo el nodo clase, en naranja, los nodos de los padres, en rosa los nodos hijos y en morado los nodos *esposa*. Podemos ver que los nodos padre son *sectorization*, *temperatura_isobaric* y *mes*; los nodos hijos son *vz_a, vz_b* y *attitude_A* mientras que los padres de estos últimos son *enginetype1*, *enginetype2* y start_fl. Si en un principio suponemos la no existencia de datos ausentes, el estado de la clase depende únicamente de estas variables, y la red bayesiana podría decirse que queda reducida al manto de Markov. La mayoría de las variables que forman parte del marco de Markov coinciden además con aquellas cuya sensibilidad era mayor según el análisis de sensibilidad, con la notable excepción de las dos variables relativas a *aircraft*, cuya alta sensibilidad es posiblemente debida a su gran número de estados posibles, y de las velocidades verticales, que no aparecen en el análisis de sensibilidad al ser únicamente descendientes.

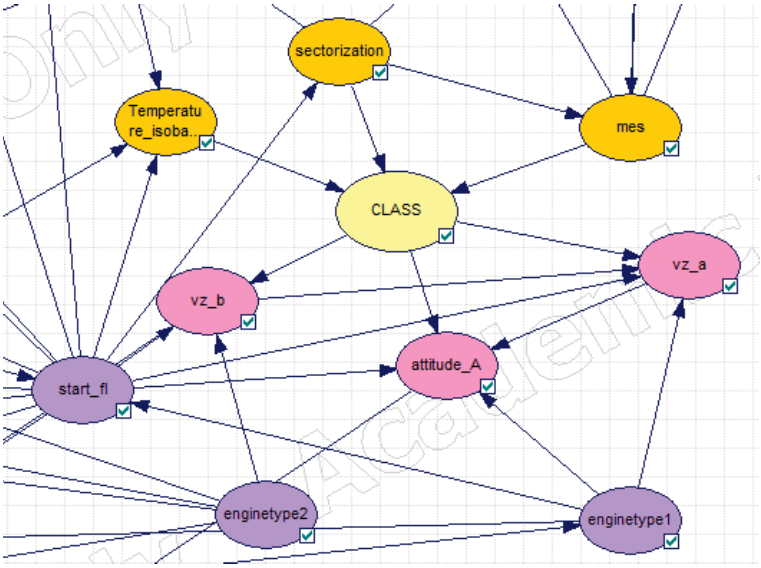

Figura 5.6: Manto de Markov del nodo clase o *CLASS*, en amarillo.

Aplicando el razonamiento humano, ¿tiene sentido que éstas sean las variables más correlacionadas con la variable clase?

Una de las circunstancias que más se suponía que afectarían a la existencia de una pérdida de separación era la situación de los controladores, su nivel de estrés y de atención, lo cual depende directamente del tráfico en el sector. Para ello se realizó la métrica de *occupancy*, que no parece tener una influencia directa en la red bayesiana. Esta métrica es la medida del número de aviones que había en vuelo en el sector correspondiente, es decir, que manejaba el controlador, en el momento determinado de la aproximación.

Sin embargo, es una exagerada simplificación de la realidad, ya que la cantidad de aviones en un momento particular no afecta directamente a lo ocupado que pueda estar un controlador, sino que es el cómo se dispongan estos, la cantidad de conflictos o el uso de rutas poco habituales lo que más afecta. La sectorización es también una medida del nivel del tráfico aéreo, ya que las sectorizaciones mayores aparecen cuánto más tráfico inmediato existe o se espera. Además, las sectorizaciones poco habituales también pueden perjudicar al rendimiento del controlador.

La tabla de probabilidades condicionales del nodo clase es grande y difícil de obtener en texto plano de la herramienta GeNIe, pero haciendo un estudio super-

ficial y utilizando la inferencia podemos ver cómo en general las sectorizaciones entre 1 y 3 y 8 y 9 muestran más *violations*, mientras que las entre 4 y 7 se quedan en aproximaciones.

La variable *mes* también es padre del nodo clase. Una posible razón es que la época del año también condiciona el tráfico, los meses de verano tienen más tráfico y complicaciones. Por ejemplo, en los meses de enero, febrero y marzo vemos una prevalencia del estado *previolation*.

El último padre del nodo clase es el nodo *temperature_isobaric*, es decir, la temperatura a la altura que se produjo la aproximación. La temperatura isobárica no sólo afecta a la forma de comportarse de un vuelo y a la aparición de turbulencias, especialmente en verano, sino que además depende del mes y también de la altura a la que se produjo la aproximación. En la red bayesiana obtenida, parece ser que las temperaturas más altas están correlacionadas con la producción de una pérdida de separación.

Los nodos hijos son las velocidades verticales (*vz_a, vz_b*) de cada avión y la actitud del primer avión (*attitude_A*). El hecho de que sí aparezcan las velocidades de A y B pero no las actitudes de ambos vuelos indica algún problema en la creación de la red bayesiana. Aun así, como existe una relación, aunque no una dependencia, entre la velocidad en vertical y la actitud (habitualmente, una velocidad vertical positiva indica despegue y una negativa aterrizaje, pero no debemos olvidar que se pueden producir cambios de nivel y por tanto velocidad vertical en cualquier fase), es posible que simplemente la actitud de B haya quedado englobada en la variable *vz_b* para los casos dados. Para velocidades altas, mayor la posibilidad de *violation*, lo cual es coherente con la afirmación de [39] de que una de las principales causas de pérdidas de separación son los cambios de nivel equivocados.

Respecto a los nodos *esposa*, los tipos de motor de cada avión afectan a la velocidad vertical que pueden llevar, especialmente a la velocidad en z, los de piston y los turboprop no pueden alcanzar las mismas velocidades que los jet. Por otro lado, *start_fl* o la altura a la que se produjo el incidente es también bastante determiante. Afecta no sólo a las velocidades verticales y actitudes (plausiblemente, porque a alturas bajas es más probable que estén en despegue o aterrizaje), sino obviamente también a todas las variables meteorológicas isobáricas, ya que su cálculo está relacionado.

Podemos ver un ejemplo de inferencia seleccionando evidencias para *start_fl* y *vz_b* en la Figura 5.7, en la que vemos que una altura baja y una velocidad vertical

media en uno de los aviones lleva a que la probabilidad de pérdida de separación sea 0.75.

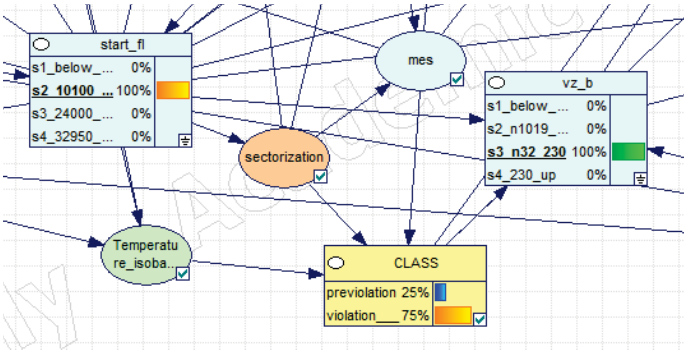

Figura 5.7: Ejemplo de inferencia con variables ocultas e evidencias en *start_fl* y *vz_a*

El hecho de que las alturas afecten puede ser consecuencia también de la existencia de vuelos militares que sólo se mueven en cotas bajas. Los vuelos militares no están controlados por los mismos controladores que los civiles, lo que dificulta la coordinación y aumenta la probabilidad de pérdida de separación.

El resto de la red expresa también relaciones coherentes y simétricas para los vuelos A(1) y B(2), como vemos a los lados en la Figura 5.1. Entre ellas, las relaciones entre modelos de avión y la compañía que organiza los vuelos, las diferentes variables meteorológicas (humedad y agua en las nubes) o la geometría de la aproximación y las actitudes de los vuelos.

Añadir finalmente que aunque las relaciones entre nodos tengan sentido la dirección de los arcos puede no tenerlo si tratamos de suponer que la red obtenida es una red causal. Los nodos hijos de la clase no podrían ser consecuencia sino causa de ésta, salvo que se interprete como que los aviones realizan un cambio de nivel en respuesta al apercibimiento de la aproximación por parte del controlador, que lo solicita.

Capítulo 6

Conclusiones y líneas futuras de investigación

En el presente trabajo se ha logrado la obtención de una red bayesiana que represente las relaciones entre las posibles circunstancias consideradas en la producción de una pérdida de separación dada una aproximación, y las prediga con un buen porcentaje de aciertos, un 77 %, y buenas características como clasificador (área bajo la curva y calibración). La red bayesiana final se obtuvo tras la consideración y comparación con las diferentes opciones que incluía el software utilizado.

Todo ello gracias la obtención de un amplio histórico de datos a través de diversas fuentes de información, del uso y modificación de algoritmos de comparación de trayectorias y del cruce de todo lo anterior. Este histórico de datos fue además limpiado de valores ausentes y atípicos, reducido y discretizado para optimizar sus resultados.

Se puede decir pues que la hipótesis de que los factores considerados son posibles contribuyentes a la existencia de una pérdida de separación ha sido probada. Gracias a la sencilla interpretación de la red bayesiana, se han identificado algunos de los factores que más afectan a la ocurrencia de una pérdida de separación entre los tenidos en cuenta: la sectorización, el momento del año, la temperatura isobárica, la altura a la que se produjo y las velocidades en vertical. Lo cual es coherente además con la salida del algoritmo Boruta de ordenación de variables por su importancia en la clasificación de las aproximaciones. Se pueden ver además cuáles

son algunas de las circunstancias que pueden llevar a pérdida de separación, como alturas entre 10100 y 24000, sectorizaciones extremas o velocidades verticales altas y medias-altas.

Un sistema de prevención que incluya la red bayesiana obtenida como herramienta de diagnóstico podría ayudar a los controladores a la identificación de aproximaciones más peligrosas de lo habitual. Además, dada las características de las redes bayesianas y gracias a la gran variedad de variables disponibles, la falta de datos, especialmente a tiempo real, que puede producirse a menudo en ambientes no simulados no sería tan problemática como si se hubiera utilizado algunos de los clasificadores más conocidos, salvo por ejemplo redes neuronales.

Es bien sabido que la obtención de los datos ocupa la mayor parte del tiempo del desarrollo de un proyecto de aprendizaje automático y es eso también lo que ha supuesto la mayor parte de la problemática en la realización de este trabajo de fin de máster. Entramos ahora en la cuestión de posibles líneas de investigación futura. Entre otras mejoras relacionadas con este hecho, nos encontramos las siguientes:

- No sólo aumentar la cantidad de datos, sino aumentar las dispersión de estos en el tiempo sería altamente necesario, ya que contar con sólo 6 meses de datos impide la comparación entre distintas fechas de forma adecuada, y en este caso no tiene en cuenta el periodo más complicado para los vuelos comerciales: el verano. También es interesante mejorar la distribución de los datos de aproximaciones que no han llegado a ser pérdida de separación, al ser reducidos para igualarlos con las pérdidas de separación, sólo se tuvieron en cuenta las fechas de los mismos, lo cual puede llevar a sesgos o sobreentrenamiento.

- Además, muchos de los factores que se querían tener en cuenta no han podido serlo por falta de una fuente adecuada de datos, entre ellos: datos de las alarmas TCAS de los aviones, registros de AIRPROX rellenados por los propios responsables e información personal de los controladores y los pilotos. Todos esos datos existen pero no estaban disponibles en las circunstancias de la realización de este trabajo, siendo el más sencillo de obtener el también más interesante para el planteamiento del sistema: los datos de los TCAS.

- Uno de los resultados más decepcionantes es la falta de relación directa entre la *occupancy* y la clase, así que se propone también la creación de una mejor métrica de carga de controlador que se pueda obtener a tiempo real.

- También se intentó tratar de cambiar la variable clase a una que además distinguiera entre pérdidas de separación de diferentes severidades (por ejemplo, altas, medias y bajas). Sin embargo, de los 2000 datos disponibles de pérdidas de separación, sólo unas pocas decenas resultaron ser de severidad alta, por lo que no se llevo a cabo.
- Otro posible enfoque sería uno que separara los vuelos militares de los comerciales y estudiara sólo las causas que afectan a las pérdidas de separación en vuelos comerciales, ya que la existencia de una problemática en las aproximaciones con vuelos militares es conocida, al igual que sus causas. Sin embargo, en este trabajo se quisieron tener en cuenta puesto que un sistema de alerta tendría que tenerlos también en consideración.

Por otro lado, sería interesante también aplicar otros métodos de aprendizaje automático al problema. Por ejemplo, un clasificador de redes neuronales, aunque sea opaco y no muestre el funcionamiento del sistema, es probable que mejorara el porcentaje de clasificación obtenido, ya que podría por ejemplo tener en cuenta datos híbridos entre discretos y continuos sin los problemas que esto conlleva en la creación de una red bayesiana a la vez que es también capaz de manejar con facilidad datos ausentes.

Bibliografía

[1] Silvia Acid and Luis M de Campos. A hybrid methodology for learning belief networks: BENEDICT. *International Journal of Approximate Reasoning*, **27** (3):235–262, 2001.

[2] Concha Bielza, Guangdi Li, and Pedro Larrañaga. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, **52**(6):705 – 727, 2011.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.

[4] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, **30**(7):1145–1159, 1997.

[5] Wray Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.

[6] Hsu-Yung Cheng, Chih-Chia Weng, and Yi-Ying Chen. Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks. *IEEE Transactions on Image Processing*, **21**(4):2152–2159, 2012.

[7] Jie Cheng, David A Bell, and Weiru Liu. An algorithm for bayesian belief network construction from data. In *Proceedings of Artificial Intelligence and Statistics*, pages 83–90, 1997.

[8] Elvira Consortium. Elvira: An environment for creating and using probabilistic graphical models. In *Proceedings of the first European workshop on probabilistic graphical models*, pages 222–230, 2002.

[9] Gregory F Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, **9**(4):309–347, 1992.

[10] Oracle Corporation. MySQL Workbench, 2005. URL `https://www.mysql.com/products/workbench/`.

[11] CRIDA. Centro de Referencia de Investigación, Desarrollo e Innovación ATM, A.I.E., 2008. URL `http://www.crida.es/`.

[12] Rónán Daly, Qiang Shen, and Stuart Aitken. Learning bayesian networks: approaches and issues. *The Knowledge Engineering Review*, **26**(2):99–157, 2011.

[13] Scott Davies and Andrew Moore. Mix-nets: Factored Mixtures of Gaussians in Bayesian Networks with Mixed Continuous and Discrete Variables. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 168–175. Morgan Kaufmann, 2000.

[14] División de Simulación CNS/ATM. Proyecto perseo, 2008. URL `http://www.enaire.es/csee/Satellite/navegacion-aerea/es/Page/1047658427197/`.

[15] Marek J Druzdzel. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a development environment for graphical decision-theoretic models. In *AAAI Innovative Applications of Artificial Intelligence Conferences*, pages 902–903, 1999.

[16] European Aviation Safety Agency (EASA). Annual safety review 2017, 2017. URL `https://www.easa.europa.eu/document-library/general-publications/annual-safety-review-2017`.

[17] ENAIRE. Familiarización con el tránsito aéreo. Documentos de la convocatoria de acceso a ENAIRE, 2016.

[18] Eurocontrol. Eurocontrol Seven Year Forecast February 2017, 2017. URL `https://www.eurocontrol.int/sites/default/files/content/documents/official-documents/forecasts/seven-year-flights-service-units-forecast-2017-2023-Feb2017.pdf`.

[19] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, **27**(8):861–874, 2006.

[20] FOCA. Aircraft proximity hazard (AIRPROX), 2015. URL `https://www.bazl.admin.ch/`.

[21] Eurocontrol for SKYbrary. AIRPROX (Aircraft Proximity), 2016. URL `https://www.skybrary.aero/index.php/AIRPROX`.

[22] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, **7**(3-4):601–620, 2000.

[23] Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, and Francisco Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, **25**(4):734–750, 2013.

[24] Dan Geiger and David Heckerman. A characterization of the dirichlet distribution with application to learning bayesian networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 196–207. Morgan Kaufmann Publishers Inc., 1995.

[25] Marko Grönroos. *Book of Vaadin*. Lulu.com, Morrisville, 2011.

[26] David J Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*. MIT press, Cambridge (Massachusetts), 2001.

[27] David Heckerman. A Bayesian Approach to Learning Causal Networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 285–295. Morgan Kaufmann, 1995.

[28] David Heckerman. A tutorial on learning with Bayesian networks. *Nato Asi Series D Behavioural And Social Sciences*, **89**:301–354, 1998.

[29] Norsys Inc. Netica Application, 1994. URL `https://www.norsys.com/netica.html`.

[30] RStudio Inc. RStudio - Open source and enterprise-ready professional software for R, 2011. URL `https://www.rstudio.com/`.

[31] Innaxis. SafeClouds - Sharing data to make aviation safer, 2016. URL `http://innaxis.org/safeclouds/`.

[32] Mehran Kafai and Bir Bhanu. Dynamic Bayesian Networks for Vehicle Classification in Video. *IEEE Transactions on Industrial Informatics*, **8**(1):100–109, 2012.

[33] Uffe Kjærulff and Linda C. van der Gaag. Making Sensitivity Analysis Computationally Efficient. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 317–325. Morgan Kaufmann, 2000.

[34] Sotiris B. Kotsiantis, Ioannis D. Zaharakis, and Panagiotis E. Pintelas. Machine Learning: A Review of Classification and Combining Techniques. *Artificial Intelligence Review*, **26**(3):159–190, 2006.

[35] Miron B. Kursa and Witold R. Rudnicki. Feature Selection with the Boruta Package. *Journal of Statistical Software*, **36**(11):1–13, 2010. URL `http://www.jstatsoft.org/v36/i11/`.

[36] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring Private Information Using Social Network Data. In *Proceedings of the 18th International Conference on World Wide Web*, pages 1145–1146. ACM, 2009.

[37] Shayne Loft, Penelope Sanderson, Andrew Neal, and Martijn Mooij. Modeling and predicting mental workload in en route air traffic control: Critical review and broader implications. *Human Factors*, **49**(3):376–399, 2007.

[38] Anders L Madsen, Michael Lang, Uffe B Kjærulff, and Frank Jensen. The Hugin tool for learning Bayesian networks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 594–605. Springer, 2003.

[39] Arnab Majumdar, Washington Ochieng, and Peter Nalder. Trend analysis of controller-caused airspace incidents in New Zealand, 1994-2002. *Transportation Research Record: Journal of the Transportation Research Board*, **1888**: 22–33, 2004.

[40] Microsoft. SQL Server Management Studio (SSMS), 2005. URL `https://msdn.microsoft.com/library/bb545450.aspx`.

[41] Richard E Neapolitan. *Probabilistic reasoning in expert systems: theory and algorithms*. Wiley, New York, 1989.

[42] Richard E Neapolitan. *Learning bayesian networks*, volume **38**. Pearson Prentice Hall, Upper Saddle River, 2004.

[43] University of Piraeus Research Center (UPRC). DART Project – Data-Driven Aircraft Trajectory Prediction Research, 2016. URL `http://dart-research.eu/`.

[44] International Civil Aviation Organization. AIRPROX investigation in the United Kingdom. Technical report, Assembly — 36th Session Technical Commission, 2007.

[45] Judea Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, 1985*, pages 329–334, 1985.

[46] Judea Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kauffman, Burlington, 1988.

[47] Judea Pearl. *Causality*. Cambridge University Press, Cambridge, 2009.

[48] Andrzej Ratajczyk. *Regional aviation safety organisations: enhancing air transport safety through regional cooperation*. PhD thesis, International Instiute of Air and Space Law, Faculty of Law, Leiden University, 2014.

[49] Roger Ratcliff. Methods for dealing with reaction time outliers. *Psychological Bulletin*, **114**(3):510, 1993.

[50] Ferat Sahin, M. Çetin Yavuz, Ziya Arnavut, and Önder Uluyol. Fault diagnosis for airplane engines using bayesian networks and distributed particle swarm optimization. *Parallel Computing*, **33**(2):124 – 143, 2007.

[51] Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. The TETRAD project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, **33**(1):65–117, 1998.

[52] Marco Scutari. Learning Bayesian networks with the BNlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

[53] Marco Scutari and Jean-Baptiste Denis. *Bayesian networks: with examples in R*. CRC press, Boca Ratón, 2014.

[54] Tomi Silander, Teemu Roos, and Petri Myllymäki. Learning locally minimax optimal Bayesian networks. *International Journal of Approximate Reasoning*, **51**(5):544–557, 2010.

[55] Eurocontrol SKYbrary. Operational Issues, Loss of Separation, 2016. `http://www.skybrary.aero/index.php/Loss_of_Separation`.

[56] Padhraic Smyth. Belief Networks, Hidden Markov Models, and Markov Random Fields: A Unifying View. *Pattern Recognition Letters*, **18**(11-13):1261–1268, 1997.

[57] Manuel Soler. *Fundamentals of Aerospace Engineering: An introductory course to aeronautical engineering*. Manuel Soler, 2014.

[58] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, **9**(1):62–72, 1991.

[59] Talend. Talend Open Studio for Data Integration, 2005. URL `https://www.talend.com/products/talend-open-studio/`.

[60] R Core Team. R language definition. *Vienna, Austria: R foundation for statistical computing*, 2000.

[61] Claire Tomlin, George J. Pappas, and Shankar Sastry. Conflict resolution for air traffic management: a study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, **43**(4):509–521, 1998.

[62] SESAR Joint Undertaking. Single Programming Document for 2017–2019, 2016. URL `https://www.sesarju.eu/newsroom/brochures-publications/single-programming-document-year-2017-2019`.

[63] Miha Vuk and Tomaz Curk. Roc curve, lift chart and calibration plot. *Metodoloski Zvezki*, **3**(1):89, 2006.

[64] Man Leung Wong, Wai Lam, and Kwong Sak Leung. Using evolutionary programming and minimum description length principle for data mining of bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(2):174–178, 1999.

[65] Peng Xie, J. H. Li, Xinming Ou, Peng Liu, and R. Levy. Using bayesian networks for cyber security analysis. In *2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, pages 211–220, 2010.

[66] Ning Xu, George Donohue, Kathryn Blackmond Laskey, and Chun-Hung Chen. Estimation of delay propagation in the national aviation system using bayesian networks. In *6th USA/Europe Air Traffic Management Research and Development Seminar*, 2005.

Apéndice A

Diagramas de cajas e histogramas del conjunto de datos inicial

En este apéndice se muestran los diagramas de cajas realizados en R para cada variable de la base de datos inicial con el objetivo de mostrar la existencia de valores atípicos, y los histogramas realizados para las variables que mostraban comportamientos extraños en los diagramas de cajas.

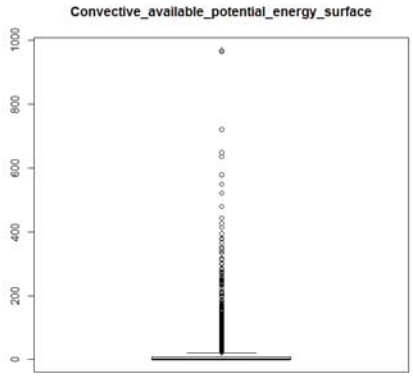

Figura A.1: Diagrama de cajas para la *convective_available_potential_energy* o CAPE. Muestra la gran cantidad de valores que tiene en 0, como era de esperar, lo cual hace que sólo podamos considerar valores atípicos los más extremos.

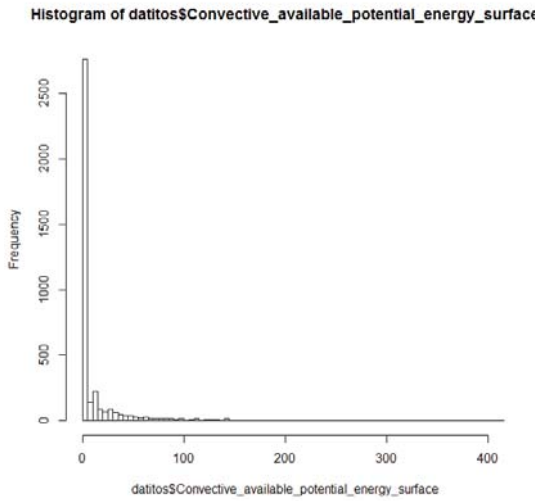

Figura A.2: Histograma para la *convective_available_potential_energy* o CAPE. No es una distribución normal, esperable dada su naturaleza de medida de tormentas, lo cual hace que sólo podamos considerar valores atípicos los valores extremos separados del grueso del histograma.

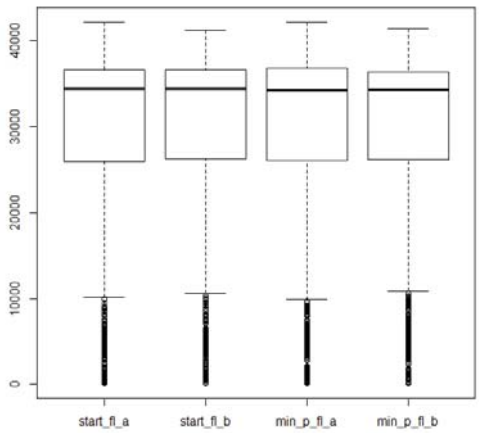

Figura A.3: Diagramas de cajas para los *start_fl* o niveles de vuelo iniciales. Vemos que existen muchos valores por debajo del bigote inferior

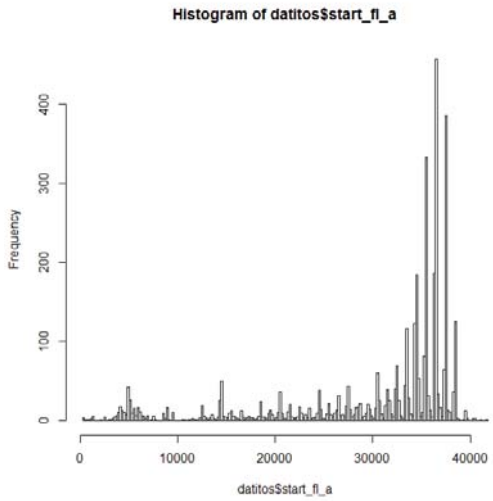

Figura A.4: Histograma para los *start_fl* o niveles de vuelo iniciales. Vemos como los valores exteriores no parecen atípicos sino que se deben a la forma asimétrica a la derecha de la distribución.

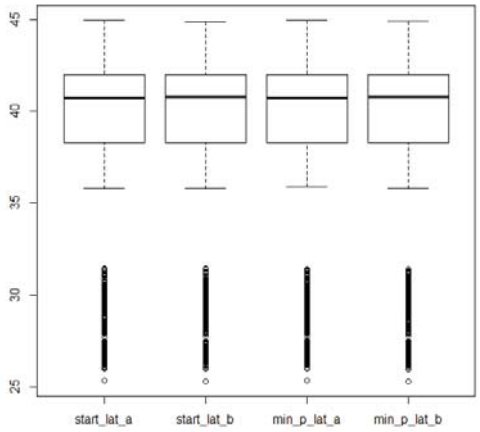

Figura A.5: Diagrama de cajas para las latitudes iniciales. Existe un espacio vacío por debajo del bigote inferior, posible distribucón bimodales.

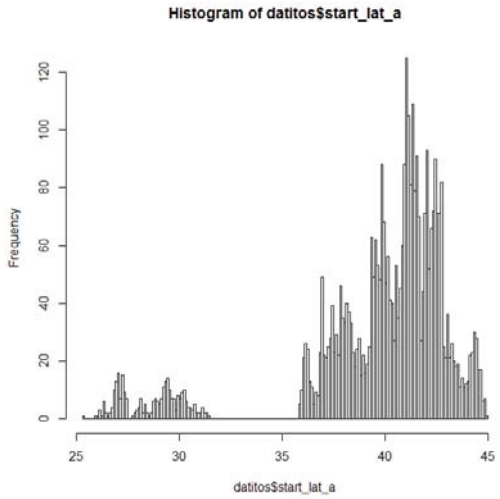

Figura A.6: Histograma para las latitudes iniciales. Está sesgado hacia la derecha y existe una discontinuidad en sus valores. Este hueco es debido a la no existencia de datos entre España y Canarias por su pertenencia al ACC de Marruecos.

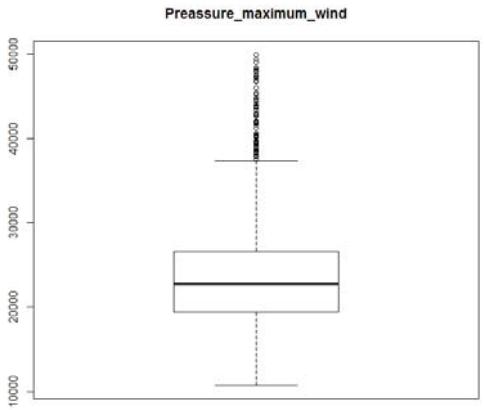

Figura A.7: Diagramas de cajas para *pressure_maximum_wind*, improbables valores atípicos.

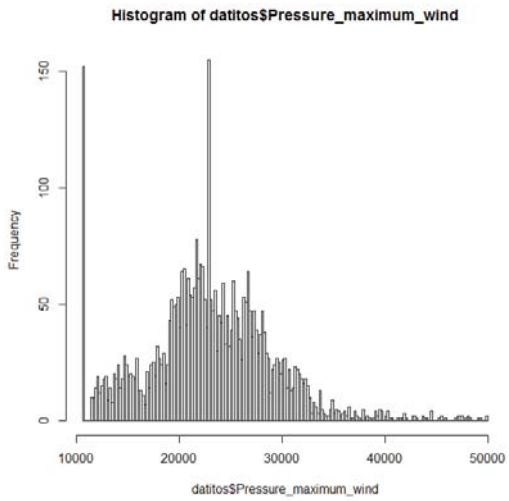

Figura A.8: Histograma para *pressure_maximum_wind*, muestra una forma de aguja y no valores atípicos.

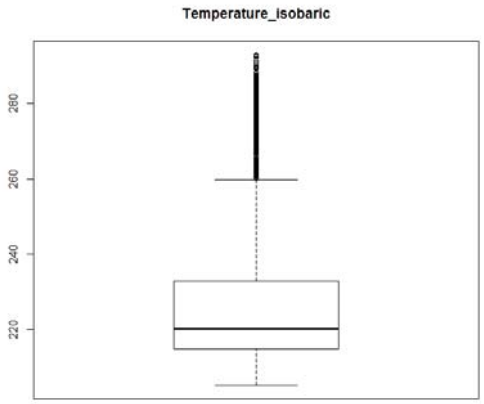

Figura A.9: Diagrama de cajas para la temperatura isobárica, improbables valores atípicos.

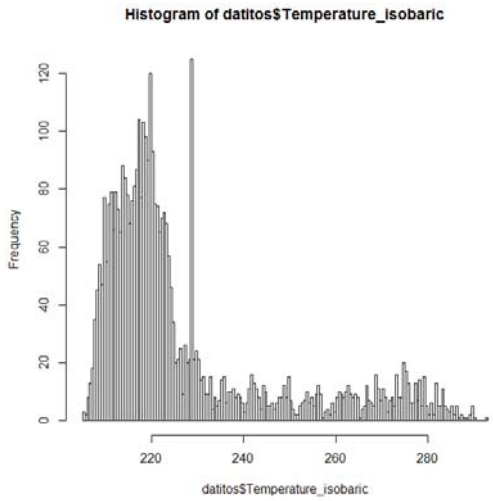

Figura A.10: Histograma de la temperatura. Es una distribución bimodal. Sin valores atípicos.

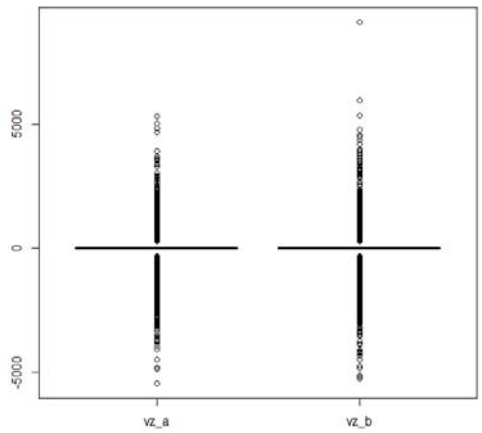

Figura A.11: Diagrama de cajas para las velocidades verticales. Gran cantidad de valores en cero, posibles valores atípicos extremos.

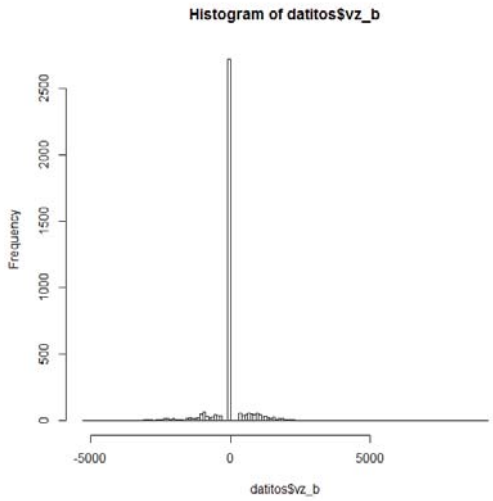

Figura A.12: Histograma para las velocidades verticales. Esperable gran cantidad de valores en 0 (crucero). Se consideran valores atípicos los que están por encima de 5000, posible error de medida o no habitual avión militar.

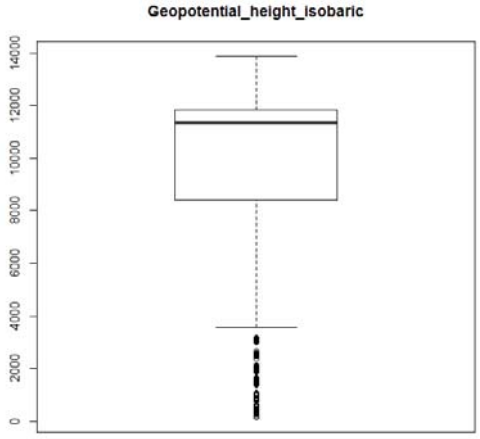

Figura A.13: Diagrama de cajas para *Geopotential_height*, distribución uniforme.

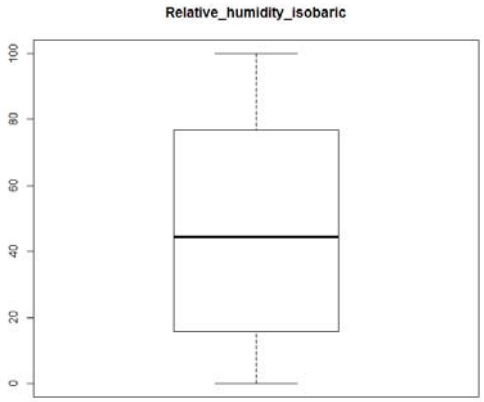

Figura A.14: Diagrama de cajas para *Relative_humidity* o humedad relativa, distribución normal.

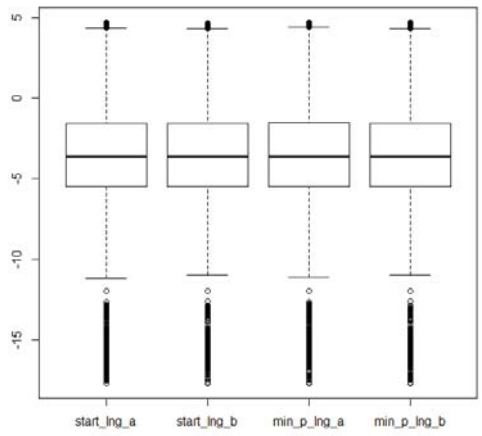

Figura A.15: Diagrama de cajas para las longitudes, muestran el mismo problema que las latitudes por el hueco de Marruecos.

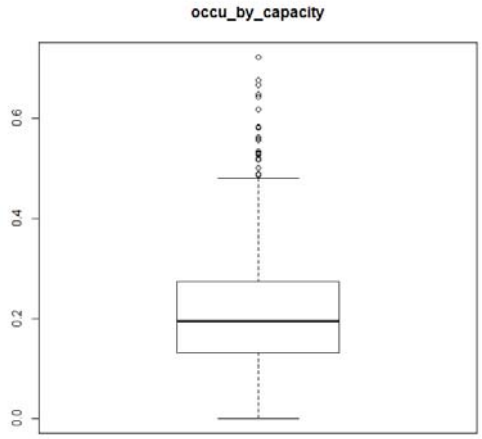

Figura A.16: Diagrama de cajas para *ocuppancy by capacity*, los valores atípicos son interesantes para el estudio.

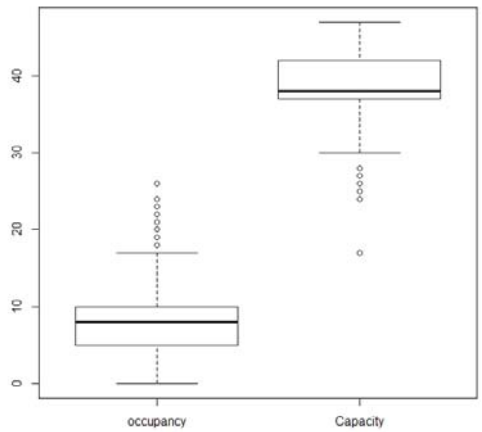

Figura A.17: Diagrama de cajas para *ocuppancy* y para *capacity*, los valores atípicos son interesantes para el estudio.

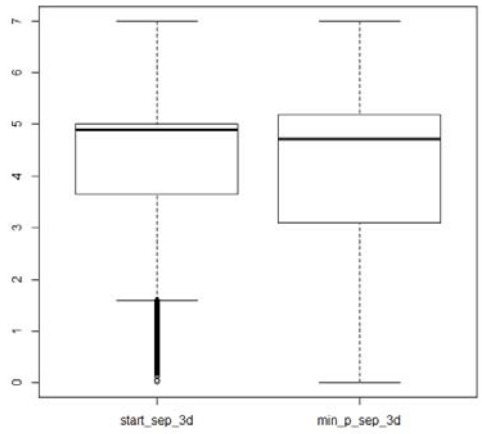

Figura A.18: Diagrama de cajas para separaciones diagonales, asimétricas, sin valores atípicos.

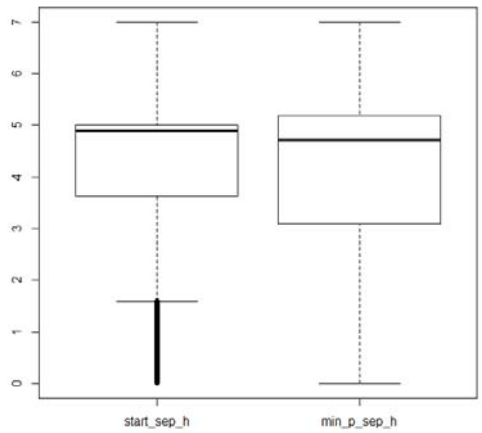

Figura A.19: Diagrama de cajas para separaciones horizontales, asimétricas, sin valores atípicos.

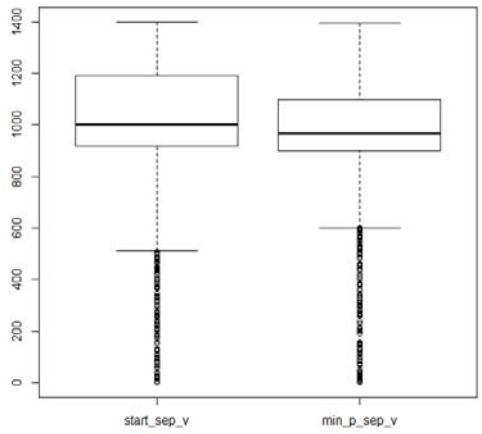

Figura A.20: Diagrama de cajas para separaciones verticales, asimétricas, sin valores atípicos.

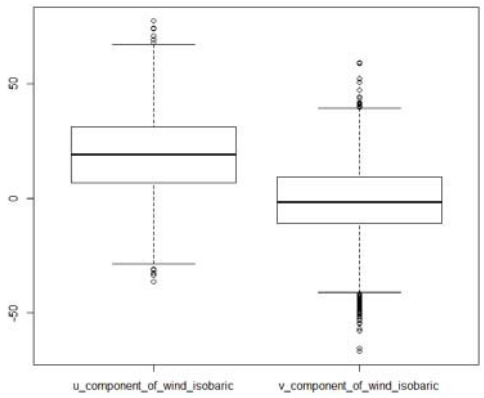

Figura A.21: Diagrama de cajas para las componentes isobáricas del viento, con valores atípicos.

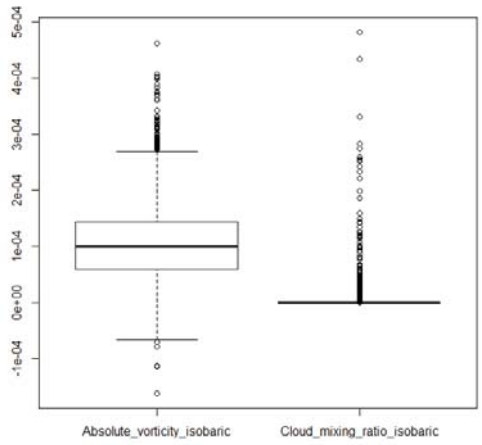

Figura A.22: Diagrama de cajas para la vorticidad absoluta, con valores atípicos, y para la mezcla de agua en las nubes, con gran cantidad de valores en 0.

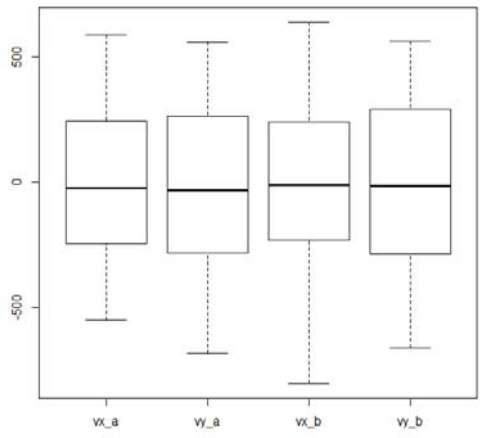

Figura A.23: Diagramas de cajas para las velocidades horizontales, normales.

Apéndice B

Consultas a bases de datos

```
DECLARE @longitude FLOAT ="+(Math.round(((row1.start_lng_a+row1.start_lng_b)/2.0 )* 2) / 2.0)+";
DECLARE @latitude FLOAT ="+(Math.round(((row1.start_lat_a+row1.start_lat_b)/2.0 )* 2) / 2.0)+";
DECLARE @referenceDateTime DATETIME ="'+((String)globalMap.get("row1.startime"))+"';
DECLARE @time TIME=CAST(@referenceDateTime AS TIME) ;
DECLARE @date DATE=CAST(@referenceDateTime AS DATE) ;

IF @longitude<0
SET @longitude=360+@longitude;


IF (@time >= '03:00:00' and @time<'09:00:00')
SET @referenceDateTime= CAST(@date AS datetime) + CAST('06:00:00' AS datetime);
IF (@time >= '09:00:00' and @time<'15:00:00')
SET @referenceDateTime= CAST(@date AS datetime) + CAST('12:00:00' AS datetime);
IF (@time >= '15:00:00' and @time<'21:00:00')
SET @referenceDateTime= CAST(@date AS datetime) + CAST('18:00:00' AS datetime);
IF (@time >= '21:00:00' and @time<'23:59:59')
SET @date=DATEADD(day,1,@date)
SET @referenceDateTime=CAST(@date AS datetime) + CAST('00:00:00' AS datetime);
IF (@time >= '00:00:00' and @time<'03:00:00')
SET @referenceDateTime= CAST(@date AS datetime) + CAST('00:00:00' AS datetime);

WITH Presion_Altura (Presion, Altura)
AS
(
   SELECT [thirdDimensionValue] , [variableValue]
   FROM [SOURCES].[dbo].[f_gfs_noaa]
   WHERE  ([dateReference]=@date and [variableName]='Geopotential_height_isobaric' and [referenceDateTime]=
   @referenceDateTime and latitude=@latitude and longitude=@longitude  and forecastHour=0)
)


SELECT  "+((Integer)globalMap.get("row1.id"))+",
    [referenceDateTime]
   ,[forecastHour]
   ,[latitude]
   ,[longitude]
   ,[thirdDimensionName]
   ,[thirdDimensionValue]
   ,[variableName]
   ,[variableValue]

FROM [SOURCES].[dbo].[f_gfs_noaa] where ([dateReference]=@date and [referenceDateTime]=@referenceDateTime
and latitude=@latitude and longitude=@longitude and forecastHour=0
and (thirdDimensionValue=(SELECT top 1 [Presion] from Presion_Altura where Altura>"
+(((Integer)globalMap.get("row1.start_fl_a"))+((Integer)globalMap.get("row1.start_fl_b")))/2.0* 0.3048+" order by Altura)
or thirdDimensionValue IS NULL));
```

Figura B.1: Consulta para base de datos de meteorología.

```
DECLARE @DATE NVARCHAR(max) = '"+TalendDate.formatDate("yyyy-MM-dd", TalendDate.parseDate("yyyy-MM-dd HH:mm:ss", (((String)globalMap.get("row3.startime")))))+"'
DECLARE @DATE2 NVARCHAR(max) = '"+TalendDate.formatDate("yyyy-MM-dd", TalendDate.parseDate("yyyy-MM-dd HH:mm:ss", (((String)globalMap.get("row3.startime")))))+"'
DECLARE @SECTOR NVARCHAR(max) = '"+((String)globalMap.get("row3.sector"))+"'
DECLARE @HOUR NVARCHAR(max) = '"+ (TalendDate.formatDate("yyyy-MM-dd HH:mm:ss",TalendDate.addDate(TalendDate.parseDate("yyyy-MM-dd HH:mm:ss",
((String)globalMap.get("row3.startime"))), -3, "mm")))+"'
DECLARE @HOUR2 NVARCHAR(max) = '"+ (TalendDate.formatDate("yyyy-MM-dd HH:mm:ss",TalendDate.addDate(TalendDate.parseDate("yyyy-MM-dd HH:mm:ss",
((String)globalMap.get("row3.startime"))), 2, "mm")))+"' ;

WITH flights AS (
    SELECT dff.*
      FROM   dwh.dbo.dimflowsflights dff
      WHERE  dff.processDateReference = @DATE and dff.ifs_DateTo >= @HOUR and dff.ifs_DateFrom <= @HOUR2
      ),
    flights_in_sector
    AS (SELECT F.callsign,
            F.adep,
            F.ades,
            F.aircraft,
            F.processdatereference,
            se.flightkey,
            se.sactastructureskey,
            se.[airspacestructurekey],
            se.sectorentrytime,
            se.sectorexittime,
            ass.sectorcode
        FROM   dwh.dbo.flowssactaentriesfacts se
            INNER JOIN flights F
                ON F.flightkey = se.flightkey
            INNER JOIN dwh.[dbo].[airspacestructuresspatial] ass
                ON ass.[airspacestructurekey] =
                  se.[airspacestructurekey]
                  AND ass.sectorcode LIKE @SECTOR),
    tracks
    AS (SELECT f.callsign,
            f.adep,
            f.ades,
            f.aircraft,
            t.*,
            f.sectorcode
        FROM   dwh.dbo.flowstracksfacts t WITH (INDEX(ni_flightkey))
            INNER JOIN flights_in_sector f
                ON f.flightkey = t.flightkey
                  AND t.[time] BETWEEN f.sectorentrytime AND
                                       f.sectorexittime)
  SELECT Count(DISTINCT ttt.flightkey)
  , '"+((Integer)globalMap.get("row3.id"))+"'

  FROM   tracks AS ttt
      INNER JOIN [DWH].dbo.calendardate CDe
          ON ttt.date = CDe.datekey
      INNER JOIN [DWH].dbo.calendartime CTe
          ON ttt.time = CTe.timekey
  WHERE  Cast(CDe.date AS DATETIME)
      + Cast(Cte.time AS DATETIME) > @HOUR
      AND Cast(CDe.date AS DATETIME)
```

Figura B.2: Consulta para calcular el número de vuelos en un sector.

Apéndice C

Descripción de las variables del conjunto de datos final

Cuadro C.1: Descripción de las variables del conjunto final de datos

	mes	hora	start_lat	start_lng
1	M 01:796	H 12 : 364	Min. :25.32	Min. :-17.672
2	M 02:689	H 11 : 339	1st Qu.:38.54	1st Qu.: -5.507
3	M 03:721	H 08 : 310	Median :40.85	Median : -3.612
4	M 04:823	H 10 : 301	Mean :39.73	Mean : -3.865
5	M 05:764	H 13 : 289	3rd Qu.:42.05	3rd Qu.: -1.552
6	M 06: 49	H 09 : 288	Max. :44.92	Max. : 4.660
7		(Other):1951		
8	Multimodal	Multimodal	Unimodal	Unimodal

	start_fl	atc_unit	sectorization	convergence
1	Min. : 250	GCCC : 295	8A2 : 677	0:2497
2	1st Qu.:27602	LECBRTE : 270	7A : 448	1:1345
3	Median :34550	LECBRTW : 558	6A : 363	
4	Mean :30497	LECMN :1030	5A : 273	
5	3rd Qu.:36500	LECMS : 974	4A : 262	
6	Max. :41675	LECS : 715	8B2 : 173	
7			(Other) :1646	
8	Unimodal	Unimodal	Unimodal	Unimodal

	company_a	aircraft_a	enginetype_a	company_b
1	MILITAR : 521	B738 :1046	Jet :3504	MILITAR : 532
2	RYR : 496	A320 : 911	Piston : 65	RYR : 451
3	VLG : 289	A319 : 393	Turboprop : 273	VLG : 269
4	TAP : 243	A321 : 238		EZY : 253
5	EZY : 228	CAZAs : 172		TAP : 216
6	Less_infrequent: 215	ERJs : 99		Frequent: 189
7	(Other) :1850	(Other) : 983		(Other) :1932
8	Unimodal	Unimodal	Unimodal	Unimodal

	aircraft_b	enginetype_b	occupancy	occu_by_capacity
1	A320s :1523	: 0	Min. : 0.000	Min. :0.0000
2	B737s :1173	Jet :3525	1st Qu.: 5.000	1st Qu.:0.1351
3	CAZAs : 177	Piston : 50	Median : 8.000	Median :0.2000
4	CRJs : 101	Turboprop : 267	Mean : 7.916	Mean :0.2068
5	S76 : 80		3rd Qu.:11.000	3rd Qu.:0.2750
6	CN35 : 79		Max. :26.000	Max. :0.7222
7	(Other) : 709			
8	Unimodal	Unimodal	Unimodal	Unimodal

	v_horizontal_a	vz_a	attitude_A	v_horizontal_b
1	Min. : 40.7	Min. :-5438.00	CLIMB : 534	Min. : 1.325
2	1st Qu.:375.8	1st Qu.: 0.00	CRUISE :2513	1st Qu.: 378.139
3	Median :423.4	Median : 0.00	DESCEND : 795	Median : 426.746
4	Mean :402.5	Mean : -13.68		Mean : 405.188
5	3rd Qu.:466.1	3rd Qu.: 0.00		3rd Qu.: 468.151
6	Max. :777.2	Max. : 5319.00		Max. :1042.404
7	Unimodal	Unimodal	Unimodal	Unimodal

	vz_b	attitude_B	Absolute_vorticity	Cloud_water
1	Min. :-5225.000	CLIMB : 564	Min. :-0.0001630	Min. :0.00000
2	1st Qu.: 0.000	CRUISE :2475	1st Qu.: 0.0000590	1st Qu.:0.00000
3	Median : 0.000	DESCEND : 803	Median : 0.0001020	Median :0.00000
4	Mean : 1.925		Mean : 0.0001052	Mean :0.09108
5	3rd Qu.: 0.000		3rd Qu.: 0.0001444	3rd Qu.:0.08000
6	Max. : 9119.000		Max. : 0.0004610	Max. :2.49000
7	Unimodal	Bimodal	Unimodal	Bimodal

	Geopotential_height	Precipitable_water	Pressure_maximum_wind	Pressure_reduced_to_MSL
1	Min. : 134.3	Min. : 1.70	Min. :10733	Min. : 99181
2	1st Qu.: 9231.0	1st Qu.:10.30	1st Qu.:19477	1st Qu.:101208
3	Median :11587.3	Median :13.90	Median :22879	Median :101705
4	Mean : 9896.4	Mean :14.19	Mean :22941	Mean :101700
5	3rd Qu.:11845.6	3rd Qu.:17.70	3rd Qu.:26721	3rd Qu.:102163
6	Max. :13812.9	Max. :34.70	Max. :49970	Max. :103989
7	Unimodal	Unimodal	Unimodal	Unimodal

	Relative_humidity	Temperature_isobaric	Total_wind	CLASS
1	Min. : 0.00	Min. :205.2	Min. : 0.3265	previolation:1713
2	1st Qu.: 15.50	1st Qu.:214.5	1st Qu.:17.2834	violation :2129
3	Median : 43.40	Median :219.9	Median :27.0283	
4	Mean : 46.51	Mean :227.2	Mean :27.3670	
5	3rd Qu.: 77.15	3rd Qu.:229.3	3rd Qu.:37.4318	
6	Max. :100.00	Max. :292.7	Max. :74.2114	
7	Unimodal	Bimodal	Unimodal	Unimodal