Universitat Autònoma de Barcelona

escola d'enginyeria

Análisis y Predicción de los Retrasos de Vuelo

Estudio del Aeropuerto de Seattle-Tacoma

Memoria del Proyecto de Fin de Grado
Gestión Aeronáutica
realizado por
Raúl Monje Solá
y dirigido por
Liana Napalkova
Sabadell, 07 de Julio de 2015

Universitat Autònoma de Barcelona

escola d'enginyeria

El sotasignat, *Liana Napalkova*

Professor/a de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en/na *Raúl Monje Solá*

I per tal que consti firma la present.

Signat: .....

Sabadell, ...de.....de 2015

FULL DE RESUM – TREBALL FI DE GRAU DE L'ESCOLA D'ENGINYERIA

Títol del projecte: Análisis y Predicción de los Retrasos de Vuelo

Autor: Raúl Monje Solá **Data:** 10 de Juliol de 2015
Tutora: Liana Napalkova
Titulació: Gestió Aeronàutica

Paraules clau: Retards, vol, predicció, anàlisis.
Palabras clave: Retrasos, vuelo, predicción, análisis.
Key words: Delays, Flight, prediction, analysis.

Resum:

El projecte següent tracta d'investigar els retards de vol mitjançant l'anàlisi de l'impacte de les demores al sistema de transport aeri, previ repàs dels diferents mètodes de mineria de dades i de modelatge predictiu, amb la finalitat de crear un model de predicció real utilitzant dades directes de l'aeroport internacional de Seattle - Tacoma. S'han comparat els resultats dels models implementats, amb un AUC de 0.7724 al cas del model de regressió logística, i un AUC de 0.9033 al cas de Gradient Boosting Machine, dels quals s'han extret conclusions de vital importància per a comprendre i avaluar les causes i conseqüències dels retards de vol.

Resumen:

El proyecto siguiente trata de investigar los retrasos de vuelo mediante el análisis del impacto de las demoras en el sistema de transporte aéreo, previo repaso de los diferentes métodos de minería de datos y de modelado predictivo, con el propósito de crear un modelo de predicción real utilizando datos directos del aeropuerto internacional de Seattle-Tacoma. Se han comparado los resultados de los modelos implementados, con un AUC de 0.7724 en el caso del modelo de regresión logística, y un AUC de 0.9033 en el caso de Gradient Boosting Machine, de los cuales se han extraído conclusiones de vital importancia para comprender y evaluar las causas y consecuencias de los retrasos de vuelo.

Abstract:

The given project is dedicated to the investigation of flight delays through analysing the impact of the delays on the air transport system and reviewing different methods of data mining and predictive modelling, with the final objective to create predictive models using real data from the Seattle-Tacoma International Airport. The results obtained with two different models are compared (AUC of 0.7724 in case of logistic regression model, and AUC of 0.9033 in case of Gradient Boosting Machine). The conclusions vital to understand and assess the causes and consequences of flight delays are made in the thesis.

TABLA DE CONTENIDO

TABLA DE ILUSTRACIONES

INTRODUCCIÓN

Estado del Arte y Motivación

La operativa aeronáutica en general, requiere de constante supervisión y control. Una operativa constantemente afectada por multitud de factores externos e internos al propio operador, necesita de unos habituales y reiterados procedimientos de análisis de datos y modelos predictivos, con el fin de evitar y/o apaciguar los efectos nocivos de las inferencias en la operativa normal, así como con el fin de prever comportamientos y optimizarlos para lograr aumentar la eficacia y eficiencia de las acciones.

Cada año, una cantidad considerable de vuelos de las aerolíneas se retrasa o cancela, costando al sistema de transporte aéreo miles de millones de euros en pérdidas de tiempo y dinero (Aproximadamente 25€/minuto que el avión se retrasa o está en tierra parado). Las aerolíneas están constantemente buscando maneras de hacer vuelos más eficientes.

Las predicciones tienen una gran importancia en el sector aeronáutico, como por ejemplo, la predicción de tiempos de llegada de una aeronave. Para dicha predicción, necesitaremos un análisis de datos como inputs (Información meteorológica, tiempos de entrada y salida en el aeropuerto de origen, tiempo medio en espera, o la velocidad media de la aeronave) con el propósito de obtener unos outputs cuyo resultado nos será de gran ayuda en la toma de decisiones y en la creación de afirmaciones predictivas.

Otro claro ejemplo sería la predicción del punto álgido de movimiento de pasajeros en un aeropuerto en temporadas críticas, como la época estival. Para esta predicción necesitaremos otro tipo de datos como inputs, como el volumen de pasajeros de los años anteriores, la situación de la economía actual, el número de reservas o el porcentaje de ocupación hotelera. Con el análisis de dichos datos se obtendrán los resultados que nos proporcionarán información vital para el desarrollo de predicciones fiables. Dichas predicciones forman una de las partes más importantes de la planificación de operaciones aeronáuticas.

Una predicción fiable permite crear modelos de planificación más precisos y eficaces, ahorrando una considerable cantidad de tiempo y recursos a todas las empresas e instituciones que forman parte de la operativa diaria aeronáutica. Una buena predicción en la llegada de una aeronave, como en el ejemplo anterior, permite organizar las tareas de embarque y escala de forma precisa y continuada, evitando los "standby moments" y reduciendo tiempo y recursos.

Así, una predicción de alta precisión también en el volumen de pasajeros en un intervalo de tiempo específico, permite asignar los recursos justos y necesarios a dicha demanda, prever los colapsos y congestiones y mejorar la atención al cliente.

Las predicciones permiten en general, mejorar la eficacia y la eficiencia de los sistemas actuales, crear nuevas operativas o modificaciones adaptadas a dichas predicciones y optimizar los procesos aeronáuticos, mejorando en tiempo, recursos e incluso en seguridad.

Objetivo y tareas del proyecto

El objetivo de este trabajo es investigar los retrasos de vuelo mediante el análisis del impacto de las demoras en los vuelos en el sistema de transporte aéreo, cómo los retrasos se propagan entre los aeropuertos, que son los factores de influencia, etc. Así como predecir cuándo van a suceder y los posibles impactos sobre la operativa aeronáutica general.

Con el fin de realizar estos objetivos, se han realizado las siguientes tareas:

1. Analizar el estado del arte de los retrasos de vuelo, poniendo especial énfasis en los factores de impacto y en las consecuencias negativas de los retrasos en el sistema de aeronáutico general.

2. Desarrollar dos modelos (logística y gradient boosting machine) para predecir la ocurrencia de los retrasos de vuelo en los aeropuertos.

3. Aplicar métodos de minería de datos para explorar y procesar datos reales estadísticos del aeropuerto estadounidense de Seattle-Tacoma International Airport (Washington, USA).

4. Realizar un análisis comparativo de los modelos propuestos usando datos de vuelo reales procesados.

Valor práctico

El estudio y la aplicación práctica de este proyecto son de vital importancia para el correcto análisis, predicción y optimización de los procesos y variables que afectan e influyen en la operativa aeronáutica general.

Cada minuto que una aeronave lleva con retraso, tiene unos costes tangibles e intangibles altísimos, tanto para las compañías aéreas como para las empresas gestoras de aeropuertos, pasando por los comercios de los mismos que tienen que lidiar con situaciones de prisa de los pasajeros, viendo reducidos sus beneficios.

La implementación de dichos análisis y herramientas de predicción en la operativa aeroportuaria, permitirán conocer a fondo en qué manera afectan los factores y las variables al correcto desarrollo de cualquier acción dentro de la aviación y conocer sus consecuencias, así como crear predicciones altamente precisas de cuándo los vuelos van a retrasarse, cancelarse o demorarse, aportando un alto valor práctico.

Metodología

Para el desarrollo del Proyecto, se han usado diferentes métodos, programas y herramientas con el fin de implementar las diferentes tareas de forma óptima y correcta.

En primer lugar se ha realizado un análisis y minería de datos del aeropuerto de Seattle-Tacoma, seleccionando aquellas variables importantes para la implementación, así como los diferentes métodos de procesamiento e ingeniería de datos para limpiar y mejorar la eficiencia y efectividad, como el

suavizado de datos y la detección de anomalías, con el propósito de mejorar la calidad de los datos a utilizar.

Dichos datos se han incluido en una base de datos mediante SQLite, SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una pequeña biblioteca escrita en C.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

A continuación se ha efectuado un análisis y comprensión de los diferentes métodos modelado predictivo que existen en la actualidad:

- Regresión logística, Máquinas de soporte vectorial, Random Forests, Arboles de decisión, Redes Neuronales y Gradient Boosting Machine.

Se han implementado de forma práctica dos de los métodos de modelado predictivo citados anteriormente, regresión logística y Gradient Boosting Machine (GBM), para crear y comparar los resultados de ambos modelos con el fin de obtener datos reales de los retrasos de vuelo en el aeropuerto de Seattle-Tacoma.

Estos modelos se han creado mediante el lenguaje R[24], R es un lenguaje y entorno de programación para análisis estadístico y gráfico.

Se trata de un proyecto de software libre, resultado de la implementación GNU del premiado lenguaje S. R y S-Plus -versión comercial de S- son, probablemente, los dos lenguajes más utilizados en investigación por la comunidad estadística.

El programa utilizado para su creación mediante R, es RStudio, programa de código abierto y entorno de desarrollo integrado (IDE) para R, de programación para el cálculo estadístico y gráfico.

A continuación se han extraído unas conclusiones sobre la predicción de los retrasos en el aeropuerto objetivo.

Estructura de la tesis

7 Apartados, 54 Figuras, 88 páginas.

- **Introducción:** Describe la importancia de las predicciones en el mundo aeronáutico.
- **Análisis y estudio del problema:** Analiza la situación actual de los retrasos y deja entrever las causas del problema.
- **Métodos de Predicción**: Analisis de los metodos de modelado predictivo**.**
- **Caso de Estudio:** Caso de studio real, modelado predictivo.
- **Conclusiones**
- **Bibliografía**
- **Anexo**

1. ANÁLISIS Y ESTUDIO DEL PROBLEMA

1.1 Los Retrasos

En un entorno en el que suceden más de 93.000 vuelos diarios en más de 9.000 aeropuertos de todo el mundo, con casi 13.000 aeronaves simultáneas en el aire, se produce uno de los problemas más frecuentes en la operativa de cualquier usuario del mundo aeronáutico, los retrasos. Muchas de las aerolíneas actuales toman, por ejemplo, la decisión desafortunada de exprimir más vuelos en el horario. En el mundo de las aerolíneas, los retrasos se acumulan a medida que el día avanza. En la época estival, por ejemplo, las líneas aéreas gozaban de una puntualidad en torno al 85% o mejor hasta media mañana. A media tarde, la tasa se redujo a lo largo de la tarde a un 70%, a continuación, se sumergió en un 60% a la hora de la cena.

Según la CODA[1] (Central Office of Delay Analysis) de Eurocontrol, en el año 2013 los vuelos acumularon una media de 9.3 minutos de retraso por vuelo, de los cuales casi el 45% se deben a los retrasos denominados Reactionary, que son aquellos retrasos que se deben a un retraso en la llegada de la aeronave, de los pasajeros, de la tripulación o de la carga. Un 30% de los retrasos totales se atribuyen de tipo Airline, referentes a aquellos retrasos que son provocados por la aerolínea.

En Mayo de 2015[2], el retraso medio por vuelo a la salida (calculado como la diferencia entre el ETS y

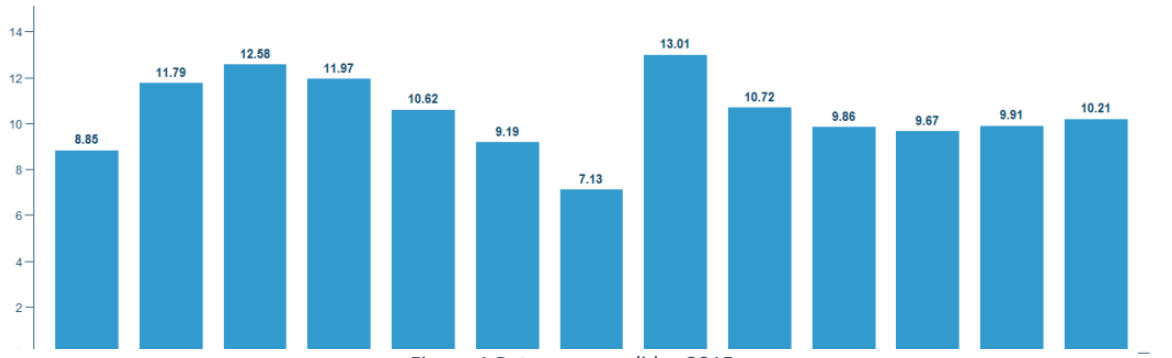


Figura 1 Retrasos en salidas 2015

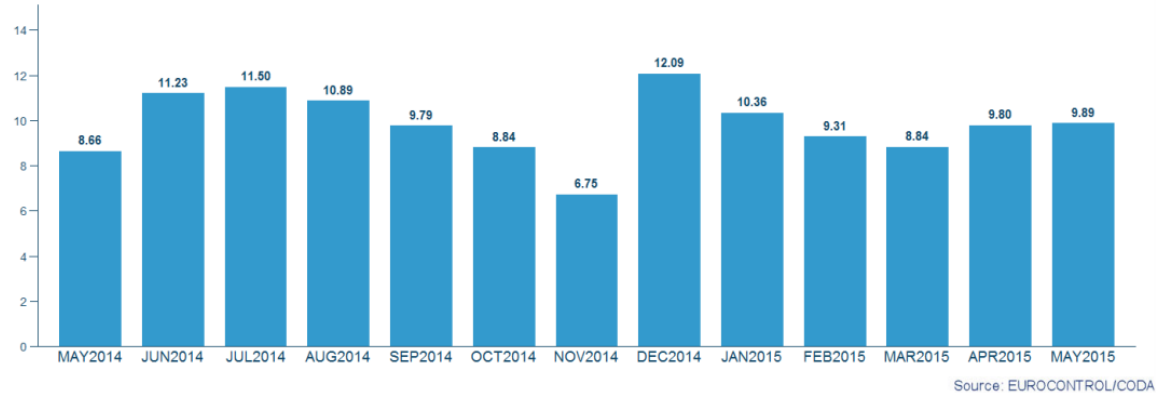


Figura 2, Retrasos en llegadas 2015

AOBT) aumentó de 1,3 minutos a 10,2 minutos por vuelo[3]. Para el tráfico de llegadas, el retraso medio por vuelo se incrementó de 1,2 minutos a 9,9 minutos en comparación con mayo de 2014. Para el cálculo de la demora media por vuelo esos vuelos que salen o llegan antes de lo previsto son contados como puntuales.

1.2 Términos monetarios de los retrasos

Los retrasos son extremadamente costosos para las compañías aéreas y sus pasajeros. Un estudio de 2010 encargado por la Administración Federal de Aviación estima que los retrasos de vuelos cuestan a la industria aérea 8 billones de dólares al año, en gran parte debido al aumento de los gastos en las tripulaciones, combustible y mantenimiento. Los retrasos cuestan a los pasajeros aún más, casi 17 billones de dólares.

En los primeros nueve meses del año, más de 1 millón de vuelos de las aerolíneas estadounidenses llegaron tarde - aproximadamente uno de cada cinco. De acuerdo con el Departamento de Transporte, los retrasos fueron ligeramente superiores en octubre respecto a septiembre.

Los retrasos crean otros problemas, incluyendo conexiones perdidas, bolsos perdidos y mal genio entre los viajeros frustrados.

Según un estudio realizado en la universidad de Westminster (Cook et al., 2004), cada minuto de retraso tiene un coste medio de 25€. Esto implica que el interés de las aerolíneas en minimizar esta ventana temporal sea muy alto. No únicamente provoca un elevado interés a las aerolíneas, también a los gestores aeroportuarios, ya que permite dotar de más capacidad a las infraestructuras aeroportuarias.

Un pequeño problema con un efecto dominó. (Mashable, Diciembre de 2014[4])

En una mañana de congelación recientemente en Dallas Love Field, supervisores de Southwest se presentaron a las 4 de la mañana, dos horas antes de los primeros vuelos. Asignan dos o más controladores de bolsa para cada vuelo.

La última bolsa debía estar en el avión y las puertas del compartimiento cerrado cinco minutos antes de la salida programada.

Los pilotos de vuelo 454 a Phoenix inspeccionaron su Boeing 737 en la oscuridad. Los trabajadores llamados "ops agents" revisan el papeleo, calculan el peso de la carga y el centrado. Un operario refuele a el avión. Dentro de la terminal, los agentes en la puerta de embarque 12 comenzaron la entrada de los 136 pasajeros, lo que supone cerrar la puerta del avión cinco minutos antes de la salida programada.

Y entonces, un imprevisto. Una radio de comunicaciones rota. El reemplazo fue ordenado e instalado, pushback a el avión de vuelta desde la puerta, y los pilotos rodaron a su posición para el despegue. Pero el daño ya estaba hecho.

El vuelo 454 dejó 29 minutos de retraso y llegó a Phoenix con 34 minutos de retraso. El centro de control de Southwest transmitió la información de demora a los empleados de Phoenix, que "operaron" el avión más rápido de lo normal antes de su próximo vuelo. Aun así, el avión permaneció de nueve a 28 minutos de retraso para los restantes cuatro tramos de la jornada, de acuerdo con el servicio de seguimiento FlightAware.com.

En el marco de las aerolíneas estadounidenses (caso de estudio de este proyecto), Southwest lideró las grandes aerolíneas en llegadas puntuales.

En septiembre, el porcentaje de Southwest de los vuelos que llegaron dentro de los 14 minutos previstos - que es la definición del gobierno de puntualidad – se deslizó por encima del 80% por primera vez este año. Desde el servicio de seguimiento FlightStats se dijo que en octubre, Southwest fue segundo respecto

a Delta entre las cinco mayores compañías en volumen de pasajeros, aunque todavía arrastraba rivales más pequeños, incluyendo Alaska Airlines, JetBlue y Virgin America.

Entre las grandes aerolíneas, Delta fue de peor lugar a primera en puntualidad en 2011 y ha permanecido allí desde entonces. David Holtz, vicepresidente senior de operaciones, atribuye los cambios de horario, las primas mensuales de hasta $ 100 por empleado por llegar a tiempo y otros objetivos, así como la memoria, la tarea repetible de apegarse a una lista de comprobación previa a la salida minuto a minuto, etc. Los retrasos producidos a primera hora del día son particularmente problemáticos.

Muchos viajeros asumen que las compañías aéreas tienen un montón de aviones de repuesto. Pero las grandes compañías utilizan alrededor de 600 aviones en un día promedio y mantiene sólo una docena o así en reserva en los grandes aeropuertos.

Las aerolíneas pueden aumentar su grado de puntualidad aflojando o rellenando el calendario, permitiendo más tiempo entre el despegue y llegada.

Pero esta operación tiene inconvenientes. Los aviones no vuelan tanto, por lo que ganan menos dinero. Se aumenta los costes de mano de obra debido a que las tripulaciones se pagan en base a los tiempos de vuelo programados. Y se crea el problema inverso, vuelos que llegan tan temprano que no hay puerta disponible, obligando a los pasajeros a permanecer en el avión después de que toque tierra.

Los viajeros pueden simpatizar con las compañías aéreas cuando el retraso es causado por el mal tiempo, sin embargo, son menos tolerantes cuando piensan que la compañía aérea podría haberlo hecho mejor.

Las aerolíneas se sienten cómodas con su promedio de llegar con retrasos de un 30 a 40% de las veces, utilizando la definición estricta de ser siquiera un minuto considerado retraso, no el colchón de 15 minutos. "¿Quién compraría un teléfono que funciona el 60-70% de las veces?"

1.3 Situación actual de los retrasos en Europa

En mayo 2015[2] los datos preliminares de las compañías aéreas que describen las causas de retrasos de vuelo muestran un retraso medio por vuelo demorado (ADD) de 26 minutos, un aumento de 2 minutos en comparación con mayo de 2014. Un 40% de los vuelos aproximadamente se retrasaron en la salida (PDF> = 5 minutos), un aumento de 4 puntos porcentuales en comparación con el mismo mes de 2014.

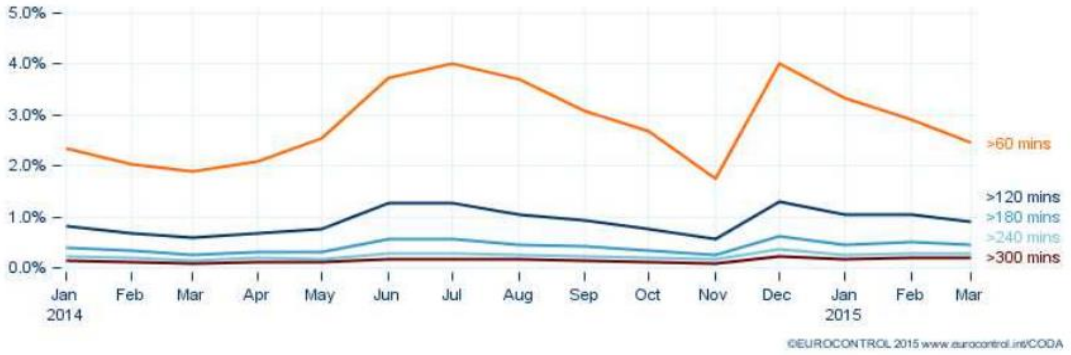


Figura 3, Porcentaje y minutos de retraso 2015

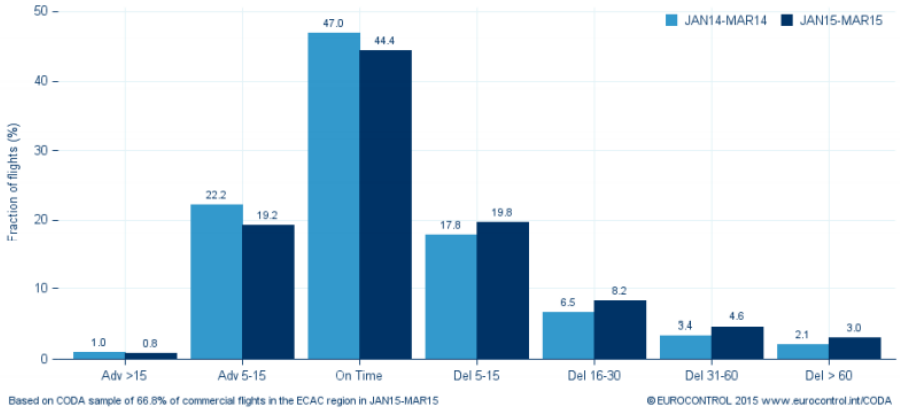


Figura 4, Porcentaje de retrasos en salidas, Mayo 2015

Como puede observarse en el grafico anterior, la puntualidad de las aerolíneas a la salida de sus vuelos, ha caído en 2015 aproximadamente un 3% con respecto al año anterior.

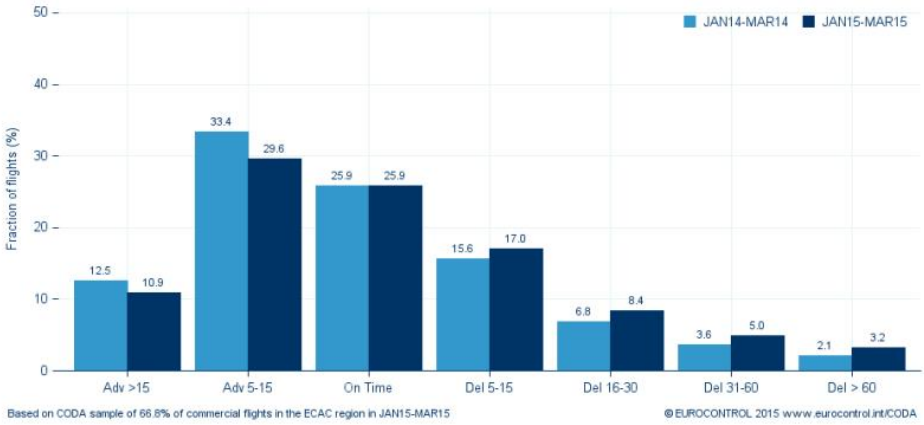


Figura 5, Porcentaje de retrasos en llegadas, Mayo 2015

En lo que respecta a la puntualidad de la llegada, el 26 % de los vuelos llegó a tiempo en el umbral de los 5 minutos antes o después de la hora de llegada prevista.

En cambio en los niveles más altos de tendencias observadas a lo largo de 2013 y principios de 2014, los vuelos con llegadas 15 minutos antes de lo previsto se redujeron de 13 % a 11 % después de los aumentos de los retrasos observados durante el trimestre. Esta proporción es todavía más alta y podría afectar las operaciones del aeropuerto en el caso de las aeronaves que llegan con frecuencia excesivamente por delante de su horario.

Programar operaciones aeronáuticas correctamente es un arte difícil: si está demasiado tiempo bloqueado un vuelo, la compañía aérea no será capaz de utilizar sus recursos de manera óptima y eficaz - personal, aeronaves, infraestructura.

Con una ventana temporal muy pequeña, sin duda puede ser peor, ya que un retraso de los vuelos puede generar retraso rotacional en la entrada de aeronaves, debiendo reacondicionar y acomodar a los pasajeros. Cuando los vuelos salen a tiempo, pero llegan después de la hora prevista de llegada causan graves retrasos en cadena.

El **Tiempo Bloque Sobreimpulso (BTO)** es el porcentaje de vuelos con un tiempo de bloque real que excede el tiempo de bloque programado. Los BTO Europeos en Mayo de 2015 fue del 31%; un aumento de 2 puntos porcentuales en comparación a mayo de 2014.

El **Indicador de Diferencia de Retraso en Vuelo (DDI -F)** es la diferencia entre la salida y la puntualidad de llegada expresada en minutos.

Esto puede ser indicado como una figura positiva o negativa, por ejemplo, en el caso de un vuelo de salida con 20 minutos de retraso y llegada con 30 minutos, el retraso de la llegada tendrá un DDI -F de 10 minutos. El DDI -F Europeo en Mayo de 2015 disminuyó en -2,9 minutos por vuelo en comparación a mayo de 2014.

BTO, DDI-F Europa, Eurocontrol 2015

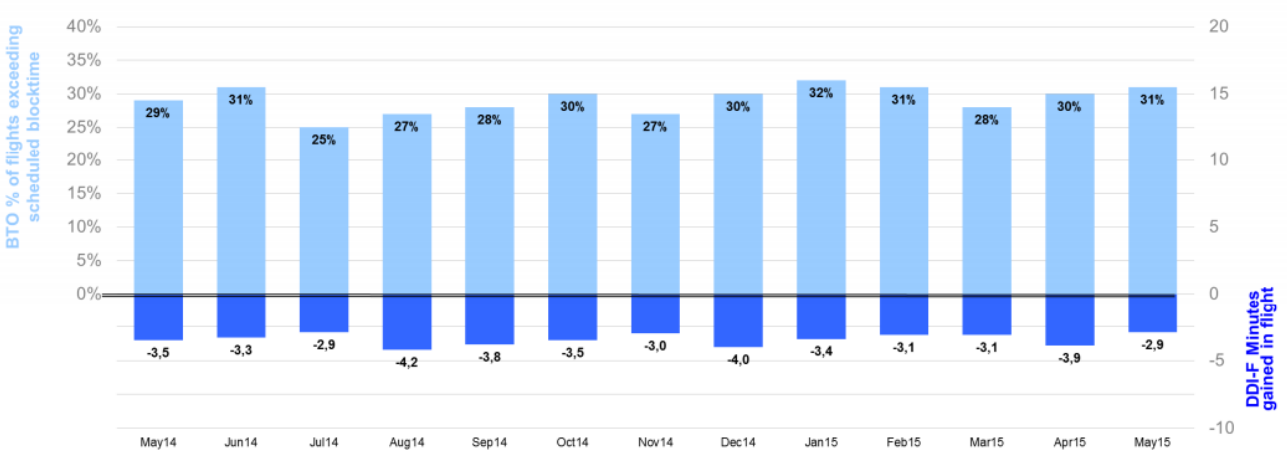


Figura 6,BTO, DDI-F Europa, Eurocontrol 2015

1.4 Marco legal y Reembolsos

El Reglamento[5] (CE) 261/2004 del Parlamento Europeo y del Consejo, de 11 de febrero (en vigor desde el día 17 de febrero de 2005), establece normas comunes sobre compensación y asistencia a los pasajeros aéreos en caso de denegación de embarque y de cancelación o gran retraso de los vuelos.

El Reglamento no se aplicará si viaja gratuitamente o con un billete de precio reducido que no esté directa o indirectamente a disposición del público.

El Reglamento 261/2004 se aplica:

En el caso de que el vuelo salga de:

- un aeropuerto comunitario, o
- un aeropuerto situado en un país no comunitario -cuyas normas no dispongan compensaciones y asistencia- con destino a un aeropuerto comunitario y el transportista encargado de efectuar el vuelo sea comunitario.

Siempre que el pasajero:

- disponga una reserva confirmada en el vuelo, disponiendo de un billete (impreso o electrónico) o de otra prueba de que ha sido aceptada y registrada por la compañía aérea, y se presente a facturación en las condiciones requeridas y a la hora indicada previamente y por escrito, incluso por medios electrónicos (en el caso de no indicarse hora alguna, con una antelación mínima de 45 minutos respecto de la hora de salida), o

- haya sido transbordado del vuelo para el que disponía de una reserva a otro vuelo.

Y se vea afectado por un retraso con respecto a la hora de salida prevista de:

- 2 horas o más en el caso de todos los vuelos de 1.500 kilómetros o menos, o
- 3 horas o más en el caso de todos los vuelos intracomunitarios de más de 1.500 kilómetros y de todos los demás vuelos de entre 1.500 y 3.500 kilómetros, o
- 4 horas o más en los vuelos no comprendidos en los apartados anteriores.

Cuando el retraso sea de 5 horas como mínimo, la compañía ofrecerá al pasajero el reembolso en 7 días del coste íntegro del billete al precio en el que se compró, correspondiente a la parte del viaje no efectuada (es decir, si decide no volar), y a la parte del viaje efectuada, si el vuelo ya no tiene razón de ser según el plan de viaje inicial del pasajero, y, si procede, un vuelo de vuelta al primer punto de partida lo más rápidamente posible.

1.5 Identificación de las tareas críticas que originan retrasos

Desde junio de 2003, las compañías aéreas informan de las causas totales de los retrasos y cancelaciones a la Oficina de Estadísticas de Transporte. Las aerolíneas que tienen un 1 por ciento del total de los ingresos de pasaje informan sobre el tiempo y las causas de la demora. En 2015, hay 14 aerolíneas que reportan estos datos.

Las aerolíneas informan sobre las causas de los retrasos amplias categorías que fueron creadas por la Air Carrier On-Time Reporting Advisory Committe[6]. Las categorías son Air Carrier, Sistema Nacional de Aviación (EEUU NAS), El Tiempo, retraso de aeronave y retrasos por Seguridad. Las causas de cancelaciones son las mismas, excepto que no hay categoría de retraso de aeronave.

¿Cómo se definen estas categorías[28]?

- **Air Carrier**: La causa de la cancelación o el retraso se debió a circunstancias dentro del control de la línea aérea (por ejemplo, de mantenimiento o de la tripulación problemas, limpieza de aviones, el equipaje de carga, abastecimiento de combustible, etc.).
- **Tiempo Extreme**: condiciones meteorológicas significativas (reales o previstas) que, a juicio de los portadores, retrasa o impide la realización de un vuelo.
- **Sistema Nacional de Aviación (NAS)**: Retrasos y cancelaciones atribuibles al sistema nacional de aviación en una amplia gama de condiciones, como las condiciones no extremas del clima, las operaciones aeroportuarias, el volumen de tráfico pesado, y el control del tráfico aéreo.
- **Retrasos de aeronaves**: Un vuelo anterior con el mismo avión llegó tarde, haciendo que el presente vuelo vuelva a salir tarde.
- **Seguridad**: Los retrasos o cancelaciones causadas por evacuación de una terminal o explanada, re-acceso a las aeronaves debido a fallo de seguridad, equipos de control averiados y / o las largas colas de más de 29 minutos a las zonas de detección.

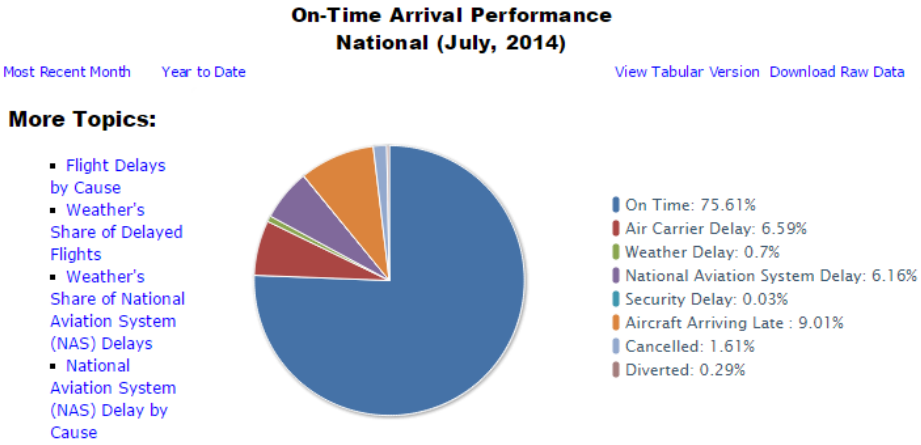


Figura 7, Porcentaje de puntualidad y causas, EEUU 2014

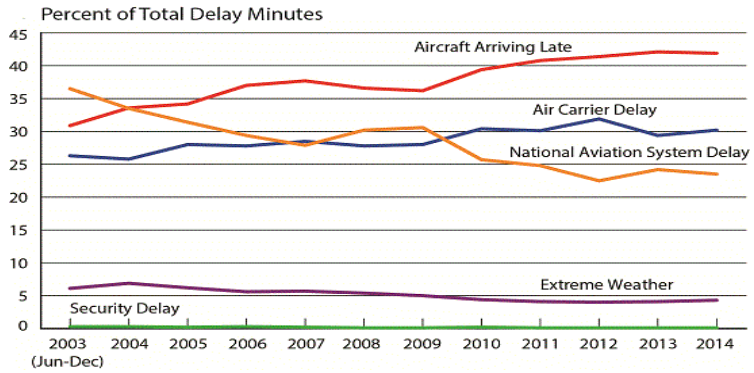


Figura 8, Porcentaje total de retrasos por año y causa

	Percent of Total Delay Minutes											
	2003 (Jun-Dec)	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Air Carrier Delay	26.3%	25.8%	28.0%	27.8%	28.5%	27.8%	28.0%	30.4%	30.1%	31.9%	29.4%	30.2%
Aircraft Arriving Late	30.9%	33.6%	34.2%	37.0%	37.7%	36.6%	36.2%	39.4%	40.8%	41.4%	42.1%	41.9%
Security Delay	0.3%	0.3%	0.2%	0.3%	0.2%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.1%
National Aviation System Delay	36.5%	33.5%	31.4%	29.4%	27.9%	30.2%	30.6%	25.7%	24.8%	22.5%	24.2%	23.5%
Extreme Weather	6.1%	6.9%	6.2%	5.6%	5.7%	5.4%	5.0%	4.4%	4.1%	4.0%	4.1%	4.3%

Figura 9, Porcentaje total de retrasos por año y causa, tabla

Una de las actividades más susceptible de originar retrasos por el volumen de actividades que conlleva es la **escala de aeronaves**[22].

La **escala** de una aeronave (*turnaround*) es el proceso ordenado de actividades que se realizan desde que se ha estacionado la aeronave, hasta que la posición queda libre para otra aeronave. La finalidad es preparar la aeronave, la cual procede de otro aeropuerto, para el vuelo. Las actividades, por su naturaleza y por el espacio físico en el que se realizan, pueden desarrollarse de forma secuencial o paralela. Por ejemplo, las actividades de desembarque y embarque de pasajeros deben ser secuenciales, mientras que la actividad de mantenimiento de la aeronave se puede realizar paralelamente a las dos anteriores, debido a que el espacio físico en el que se realizan no es el mismo.

Consta de las siguientes actividades:

- Desembarque/Embarque de pasajeros y tripulación

El embarque y desembarque de los pasajeros y de la tripulación se efectúa teniendo en cuenta dos factores principales: En primer lugar, la posición de la aeronave en el aeropuerto, es decir, en caso de

que el aeropuerto este situado en remoto, el desembarque y embarque se realizará mediante las escaleras de pasajeros y éstos se trasladaran a la terminal mediante autobuses o jardineras.

- Descarga/Carga de equipaje y carga

La carga y descarga del equipaje y la carga se puede realizar de dos formas diferentes, dependiendo del tipo de carga. La carga se puede transportar mediante contenedores ULD o pallets, éstos permiten un manejo de la carga más rápido y una mejor organización de la misma dentro de la bodega de la aeronave. Por otro lado también se puede transportar la carga suelta (bulk-load), es decir, que cada unidad de carga se maneja individualmente.

- Carga de Combustible

La carga de combustible es un proceso que aunque depende de la aeronave, la carga suele realizarse de la misma forma para asegurar la estabilidad de la aeronave. El proceso normal suele ser el siguiente: En primer lugar se llenan los tanques de combustible situados en las alas, asegurando así la estabilidad de la aeronave. Una vez se han llenado estos tanques, el resto se llenan en caso de que resulte necesario para el siguiente vuelo. La carga de combustible se realiza con mangueras que se enganchan a la aeronave.

- Mantenimiento rutinario

El mantenimiento rutinario se entiende cómo aquellas tareas que en la mayoría de los casos se trata de comprobación de los sistemas de la aeronave para asegurar el correcto funcionamiento de la aeronave durante el siguiente vuelo. Por ejemplo algunas de las tareas más comunes son las de comprobación de nivel de aceite de los motores, revisión del líquido hidráulico, niveles de oxígeno o presión de las ruedas.

- Descarga/Carga de Catering

La descarga y carga del catering consiste en el abastecimiento de alimentos y bebidas para los pasajeros. Cómo se ha comentado previamente, este servicio suele ofrecerse sobre todo en vuelos de largo recorrido y en vuelos operados por aerolíneas tradicionales.

- Limpieza de cabina

La limpieza de la cabina se refiere a la limpieza general de la cabina de pasajeros tras un vuelo, con el objetivo de que la aeronave se encuentre en las condiciones higiénicas correctas para el siguiente vuelo.

- Procedimientos de seguridad

Los procedimientos de seguridad hacen referencia a ciertas comprobaciones que se han de llevar a cabo, antes de que embarquen los pasajeros, por parte de la tripulación dentro de la cabina. El objetivo es asegurar el buen funcionamiento de sistemas orientados a preservar la seguridad de los pasajeros durante el vuelo en caso de emergencia. Tales procedimientos pueden ser, por ejemplo, comprobar el sistema de inflación de rampas de evacuación de la aeronave, máscaras de oxígeno o sistemas de prevención de incendios. Esta tarea suele llevarse a cabo entre las tareas de limpieza y embarque de pasajeros.

- Lista de comprobación previa al vuelo

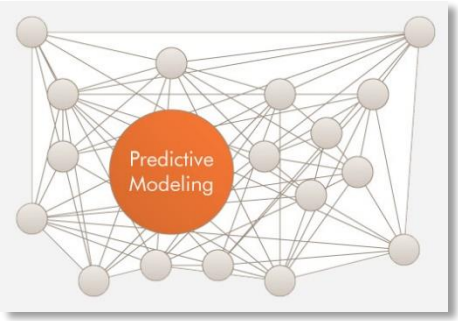
La lista de comprobación previa al vuelo se refiere a la comprobación, por parte de la tripulación, del correcto funcionamiento de todos los sistemas de vuelo de la aeronave. Esta tarea se lleva a cabo una vez se ha finalizado el embarque de los pasajeros y previamente la salida de la aeronave de la posición de estacionamiento.

2. MODELOS DE PREDICCIÓN

2.1 Definición e importancia de la predicción

La operativa aeronáutica en general, requiere de constante supervisión y control. Una operativa constantemente afectada por multitud de factores externos e internos al propio operador, necesita de unos habituales y reiterados procedimientos de análisis de datos y modelos predictivos, con el fin de evitar y/o apaciguar los efectos nocivos de las inferencias en la operativa normal, así como con el fin de prever comportamientos y optimizarlos para lograr aumentar la eficacia y eficiencia de las acciones.

El análisis predictivo[7] utiliza la estadística junto con algoritmos de data mining. Se basan en el análisis de los datos actuales e históricos para hacer predicciones sobre futuros eventos. Dichas predicciones raramente suelen ser afirmaciones absolutas, pareciéndose más a eventos y su probabilidad de que suceda en el futuro.



Los modelos predictivos analizan los resultados anteriores para evaluar qué probabilidad tiene un cliente para mostrar un comportamiento específico en el futuro con el fin de mejorar la eficacia de una operación.

Una teoría científica cuyas predicciones no son corroboradas por las observaciones, por las pruebas o por experimentos probablemente será rechazada. Las teorías que generan muchas predicciones que resultan de gran valor (tanto por su interés científico como por sus aplicaciones) se confirman o se falsean fácilmente y, en muchos campos científicos, las más deseables son aquéllas que, con número bajo de principios básicos, predicen un gran número de sucesos.

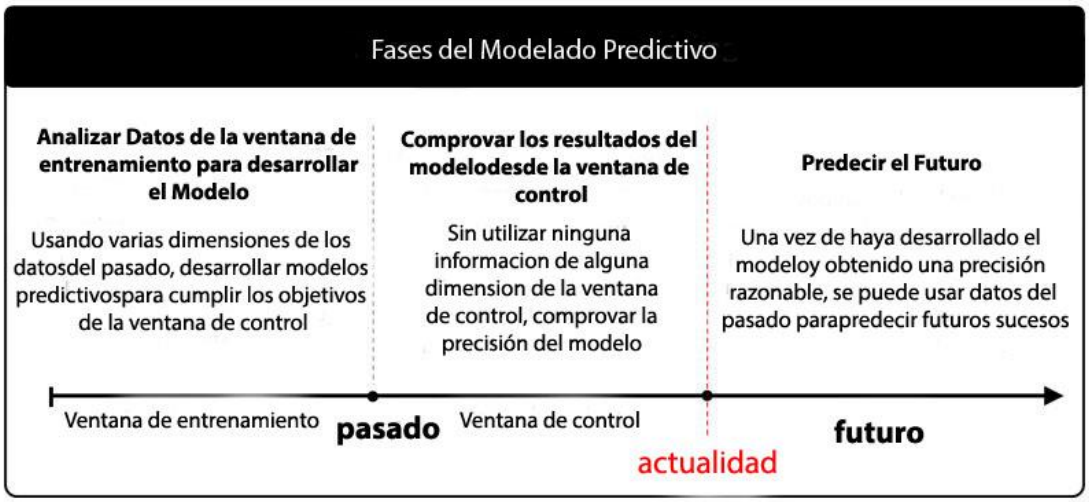


Figura 10, Fases del modelado predictivo

El modelo de predicción utiliza modelos estadísticos para la producción de outputs. Habitualmente se pretenden predecir acciones o eventos que sucederán en un futuro, lejano o inmediato, pero los modelos de predicción se pueden aplicar a cualquier tipo de evento desconocido, independientemente del momento en que ocurrió. Por ejemplo, los modelos predictivos se utilizan a menudo para detectar delitos e identificar a los sospechosos, después de que un crimen haya tenido lugar.

En muchos casos se elige el modelo sobre la base de la teoría de detección para tratar de adivinar la probabilidad de un resultado dado una cantidad de datos de entrada, por ejemplo, dado un correo electrónico se intenta determinar qué tan probable es que se trate de un correo spam.

En los modelos se pueden utilizar uno o más clasificadores para tratar de determinar la probabilidad de un conjunto de datos que pertenecen a otro grupo, es decir spam o 'publicitario'.

En función de los límites de definición, modelado predictivo es sinónimo de, o en gran medida con el ámbito del aprendizaje de máquina, como se le conoce más comúnmente en contextos académicos o de investigación y desarrollo. Cuando se implementa en el comercio, el modelado predictivo se conoce como análisis predictivos a menudo.

En el caso de la predicción, por ejemplo, es necesario predecir si cambiará la dirección del viento para poder optimizar el sistema de tráfico aéreo y elegir la pista adecuada para que las aeronaves aterricen o despeguen con viento en cara, o reenrutar su aproximación en base a este cambio de dirección. Una predicción fiable y de precisión maximiza la previsión de ocurrencia de ciertos fenómenos vitales en la actividad aeronáutica.

Las predicciones tienen una gran importancia en el sector aeronáutico, como por ejemplo, la predicción de tiempos de llegada de una aeronave. Para dicha predicción, necesitaremos un análisis de datos como inputs (Información meteorológica, tiempos de entrada y salida en el aeropuerto de origen, tiempo medio en espera, o la velocidad media de la aeronave) con el propósito de obtener unos outputs cuyo resultado nos será de gran ayuda en la toma de decisiones y en la creación de afirmaciones predictivas.

Otro claro ejemplo sería la predicción del punto álgido de movimiento de pasajeros en un aeropuerto en temporadas críticas, como la época estival. Para esta predicción necesitaremos otro tipo de datos como inputs, como el volumen de pasajeros de los años anteriores, la situación de la economía actual, el número de reservas o el porcentaje de ocupación hotelera. Con el análisis de dichos datos se obtendrán los resultados que nos proporcionarán información vital para el desarrollo de predicciones fiables.

Dichas predicciones forman una de las partes más importantes de la planificación de operaciones aeronáuticas. Una predicción fiable permite crear modelos de planificación más precisos y eficaces, ahorrando una considerable cantidad de tiempo y recursos a todas las empresas e instituciones que forman parte de la operativa diaria aeronáutica.

Una buena predicción en la llegada de una aeronave, como en el ejemplo anterior, permite organizar las tareas de embarque y escala de forma precisa y continuada, evitando los “standby moments” y reduciendo tiempo y recursos. Así, una buena predicción también en el volumen de pasajeros en un intervalo de tiempo específico, permite asignar los recursos justos y necesarios a dicha demanda, preveer los colapsos y congestiones y mejorar la atención al cliente.

Las predicciones permiten en general, mejorar la eficacia y la eficiencia de los sistemas actuales, crear nuevas operativas o modificaciones adaptadas a dichas predicciones y optimizar los procesos aeronáuticos, mejorando en tiempo, recursos e incluso en seguridad.

2.2 Métodos de análisis de datos

El análisis de datos es un proceso de inspeccionar, limpiar y transformar datos con el objetivo de resaltar información útil, lo que sugiere conclusiones, y apoyo a la toma de decisiones. El análisis de datos tiene múltiples facetas y enfoques, que abarca diversas técnicas en una variedad de nombres, en diferentes negocios, la ciencia, y los dominios de las ciencias sociales.

La minería de datos o exploración de datos (es la etapa de análisis de "Knowledge Discovery in Databases" o KDD) es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos. El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior. Además de la etapa de análisis en bruto, que involucra aspectos de bases de datos y de gestión de datos, de procesamiento de datos, del modelo y de las consideraciones de inferencia, de métricas de Intereses, de consideraciones de la Teoría de la complejidad computacional, de post-procesamiento de las estructuras descubiertas, de la visualización y de la actualización en línea.

El término es una palabra de moda, y es frecuentemente mal utilizado para referirse a cualquier forma de datos a gran escala o procesamiento de la información (recolección, extracción, almacenamiento, análisis y estadísticas), pero también se ha generalizado a cualquier tipo de sistema de apoyo informático decisión, incluyendo la inteligencia artificial, aprendizaje automático y la inteligencia empresarial.

2.2.1 Regresión Lineal

El análisis de regresión lineal[8] es una técnica estadística utilizada para estudiar la relación entre variables. En la investigación social, el análisis de regresión se utiliza para predecir un amplio rango de fenómenos, desde medidas económicas, pasando por los aspectos del comportamiento humano hasta los retrasos de un aeropuerto. Tanto en el caso de dos variables (regresión simple) como en el de más de dos variables (regresión múltiple), el análisis de regresión lineal puede utilizarse para explorar y cuantificar la relación entre una variable dependiente o criterio (Y) y una o más variables llamadas independientes o predictoras, así como para desarrollar una ecuación lineal con fines predictivos. Además, el análisis de regresión lleva asociados una serie de procedimientos de diagnóstico (análisis de residuos, puntos de influencia) que informan sobre la estabilidad e idoneidad del análisis y que proporcionan pistas sobre cómo perfeccionarlo.

La recta de regresión

Las rectas de regresión son las rectas que mejor se ajustan a la nube de puntos (o también llamado diagrama de dispersión) generada por una distribución binomial.

La correlación ("r") de las rectas determinará la calidad del ajuste. Si r es cercano o igual a 1, el ajuste será bueno y las predicciones realizadas a partir del modelo obtenido serán muy fiables (el modelo obtenido resulta verdaderamente representativo); si r es cercano o igual a 0, se tratará de un ajuste malo en el que las predicciones que se realicen a partir del modelo obtenido no serán fiables (el modelo obtenido no resulta representativo de la realidad). Ambas rectas de regresión se intersecan en un punto llamado centro de gravedad de la distribución.

Bondad de Ajuste

Además de acompañar la recta con su fórmula, resulta útil disponer de alguna indicación precisa del grado en que la recta se ajusta a la nube de puntos. De hecho, la mejor recta posible no tiene por qué ser buena.

¿Cómo se puede cuantificar ese mejor o peor ajuste de la recta? Hay muchas formas de resumir el grado en que una recta se ajusta a una nube de puntos. Podríamos utilizar la media de los residuos, o la media de los residuos en valor absoluto, o las medianas de alguna de esas medidas, etc.

La medida que ha recibido mayor aceptación en el contexto del análisis de regresión es el coeficiente de determinación R^2: el cuadrado del coeficiente de correlación múltiple. Se trata de una medida

estandarizada que toma valores entre 0 y 1 (0 si las variables son independientes y 1 cuando entre ellas existe relación máxima).

Este coeficiente posee una interpretación muy intuitiva, representa el grado de ganancia que podemos obtener al predecir una variable basándonos en el conocimiento que tenemos de una u otras variables.

2.2.2 Suavizado de datos

En estadística y en procesamiento de imágenes, para suavizar[9] un conjunto de datos es necesario crear una función de aproximación que intente capturar patrones importantes en los datos, y dejando fuera el ruido u otras estructuras a escala fina / fenómenos rápidos. En el suavizado o smoothing, los puntos de los datos de una señal se modifican, así los puntos individuales (presumiblemente debido al ruido) se reducen, y los puntos que son inferiores a los puntos adyacentes se incrementan conduciendo a una señal más suave. El suavizado se puede usarse de dos maneras importantes ayudando en el análisis de datos:

- por ser capaz de extraer más información de los datos, siempre y cuando la asunción de suavizado es razonable
- por ser capaz de proporcionar análisis flexibles y robustos.

El suavizado de datos puede distinguirse del concepto relacionado y se superponen parcialmente de ajuste de la curva de las siguientes maneras:

- ajuste de la curva a menudo implica el uso de un formulario de función explícita para el resultado, mientras que los resultados inmediatos de suavizado son los valores "suavizadas" sin uso posterior hecha de una forma funcional si hay uno;
- el objetivo de suavizado es dar una idea general de los cambios relativamente lentos de valor con poca atención a la estrecha coincidencia de valores de datos, mientras que la curva de ajuste se concentra en el logro de tan cerca un partido como sea posible.
- métodos de suavizado a menudo tienen un parámetro de ajuste asociado que se utiliza para controlar el grado de suavizado. El ajuste de curvas se ajustará cualquier número de parámetros de la función de obtener la "mejor" forma.

Sin embargo, la terminología utilizada en las aplicaciones es mixto. Por ejemplo, el uso de un spline de interpolación se ajusta a una curva suave con exactitud a través de los puntos de datos dados ya veces se llama "suavizado".

2.2.3 Detección de anomalías

En la minería de datos o data mining, la detección de anomalías[10] (o detección de outliers) es la identificación de datos, eventos u observaciones que no se ajustan a un patrón esperado u otros elementos en un conjunto de datos.

Por lo general los elementos anómalos se traducirán a algún tipo de problema como fraude bancario, un defecto estructural, problemas médicos o la búsqueda de errores en el texto. Los outliers también se conocen como valores atípicos, ruido, desviaciones y excepciones.

En particular, en el contexto de la detección de intrusiones en la red de datos, los objetos interesantes a menudo no son valores raros, pero si inesperados estallidos de actividad. Este patrón no se adhiere a la definición estadística común de un valor atípico como un objeto raro, y muchos métodos de detección de valores atípicos (en particular, los métodos no supervisados) fallarán en dichos datos, a no ser que se

hayan agregado correctamente. En lugar de ello, un análisis de conglomerados puede ser capaz de detectar los grupos de micro formadas por estos patrones.

Existen tres grandes categorías de técnicas de detección de anomalías[11].

Detección de anomalías no supervisadas: técnicas que detectan anomalías en un conjunto de datos de prueba no marcados establecidos bajo el supuesto de que la mayoría de los casos en el conjunto de datos son normales mediante la búsqueda de casos que parecen encajar menos en el resto de la conjunto de datos.

Detección de anomalías supervisada: técnicas que requieren un conjunto de datos que se ha etiquetado como "normal" y "anormal" y consiste en la capacitación de un clasificador (la diferencia clave para muchos otros de clasificación estadística problemas es el desequilibrio inherente de detección de las demás).

Semi- detección de anomalías supervisadas: técnicas que construyen un modelo que representa el comportamiento normal de un dado normal de formación conjunto de datos, y luego probar la probabilidad de una instancia de prueba que se generen por el modelo aprendido.

Las técnicas de pre-procesamiento de datos por lo general se refieren a la adición, eliminación o transformación de los datos del conjunto de entrenamiento. Las transformaciones de datos o data transformation se usan para reducir el impacto de la asimetría de datos o valores extremos que puede conducir a mejoras significativas en el rendimiento.

La extracción de características, es una técnica empírica para crear variables sustitutas que son combinaciones de varios predictores. Además, las estrategias más simples, tales como la eliminación de factores predictivos basados en la falta de contenido de información también pueden ser eficaz. La necesidad de pre-procesamiento de datos se determina por el tipo de modelo que se utilice.

Algunos procedimientos, como los modelos basados en los árboles, son notablemente insensibles a las características de los datos de predicción. Otros, como la regresión lineal, no lo son.

2.2.4 Reducción de Dimensiones

En *machine learning* y en estadística, *dimensionality reduction*[12] o reducción de dimensiones es el proceso de reducir el número de variables aleatorias bajo consideración, pudiendo ser divididas en selección de características o *feature selection* y extracción de características o *feature extraction*[7].

- *Feature selection:* La aproximación por selección de características intenta buscar un subconjunto de las variables originales (también llamadas atributos). Las dos estrategias de aproximación de *feature selection* son el filtrado y la envoltura.

- *Feature extraction:* La aproximación por extracción de características transforma los datos de un espacio de muchas dimensiones en un espacio de las mínimas posibles. LA transformación de datos puede ser lineal, pero algunas reducciones no lineares pueden usar esta técnica.

Las técnicas de reducción de datos son otra clase de transformaciones de datos para la predicción. Estos métodos reducen los datos mediante la generación de un conjunto más pequeño de predictores que tratan de capturar la mayoría de la información en las variables originales.

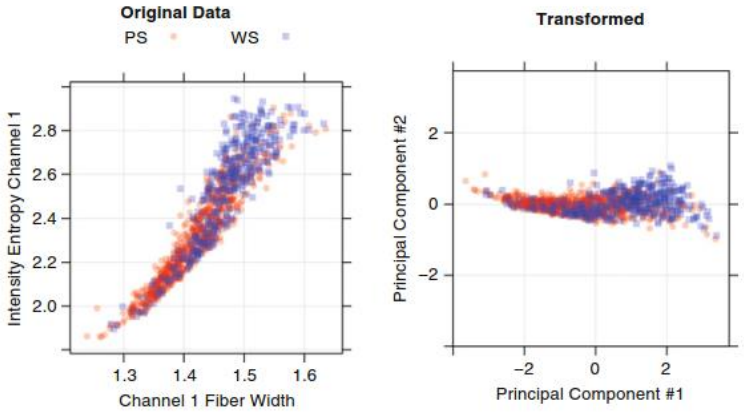


Figura 11, Reducción de dimensiones, Original vs Transformada

De esta manera, un menor número de variables utilizadas proporcionan una fidelidad razonable a los datos originales. Para la mayoría de las técnicas de reducción de datos, los nuevos predictores son funciones de los predictores originales; por lo tanto, aún se necesitan todos los predictores originales para crear las variables deseadas. Esta clase de métodos a menudo se llama extracción de señales o técnicas de extracción de características.

PCA es una técnica de reducción de datos muy utilizada creada por Abdi y Williams en 2010. Este método busca combinaciones lineales de predictores, conocidos como componentes principales (PC), que captan la mayor parte posible de la varianza. El PC primero se define como la combinación lineal de los predictores que captura la mayor variabilidad de todas las posibles combinaciones lineales. Entonces, los PCs posteriores se derivan de tal manera que estas combinaciones lineales capturan la variabilidad restante al mismo tiempo que son correlacionados con todos los PCs anteriores.

Matemáticamente, el PC j se puede escribir como:

$$PC_j = (a_{j1}\ x\ Predictor\ 1) + (a_{j2}\ x\ Predictor\ 2) + \cdots + (a_{jP}\ x\ Predictor\ P)$$

P es el número de predictores. Los coeficientes AJ1, AJ2,..., AJP se llaman pesos de los componentes y nos ayudan a entender que predictores son más importantes para cada PC.

Además de enseñarnos acerca de los datos, la reducción de dimensión puede conducirnos a mejores modelos para la inferencia estadística.

2.2.5 Ingeniería de Características

La ingeniería de características o *feature engineering*[13], trata de cómo se codifican los predictores, pudiendo tener un impacto significativo en el rendimiento del modelo. Por ejemplo, usar combinaciones de predictores puede ser más efectivo que usar valores individuales: la relación de dos predictores pueden ser más eficaz que el uso de dos predictores independientes.

> *Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.*
>
> —Andrew Ng, *Machine Learning and AI via Brain simulations*

La "correcta" ingeniería de características depende de varios factores. En primer lugar, algunas codificaciones pueden ser óptimas para algunos modelos y mala para otros. Por ejemplo, los modelos basados en árboles van a dividir los datos en dos o más bandejas. Teóricamente, si la variable "*Month*" o mes fuera importante, el árbol dividiría el día numérico del año correspondiente. También, en algunos modelos, las múltiples codificaciones de los mismos datos pueden causar problemas.

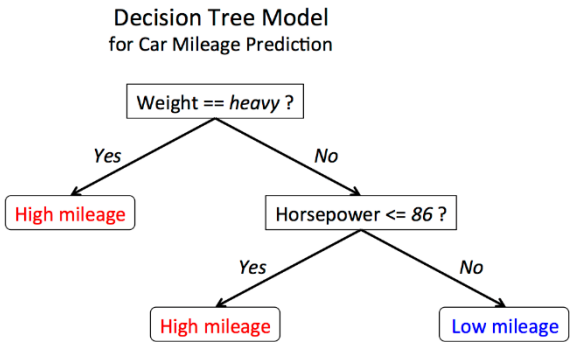


Figura 12, Ejemplo de árbol de decisión

Algunos modelos incluyen directamente la selección de características, lo que significa que el modelo sólo incluirá predictores que ayudan a maximizar la precisión. En estos casos, el modelo puede escoger y elegir los que la representación de los datos es mejor.

La relación entre el predictor y el resultado es un segundo factor. Por ejemplo, si hubiera un componente estacional de los datos, entonces el día numérico del año sería mejor. Además, si algunos meses muestran mayores tasas de éxito de otros, entonces la codificación basada en el mes es preferible. Al igual que con muchas preguntas de estadísticas, la respuesta a "¿qué métodos de ingeniería son los mejores?" Es depende. Específicamente, se depende del modelo que se utiliza y la verdadera relación con el resultado.

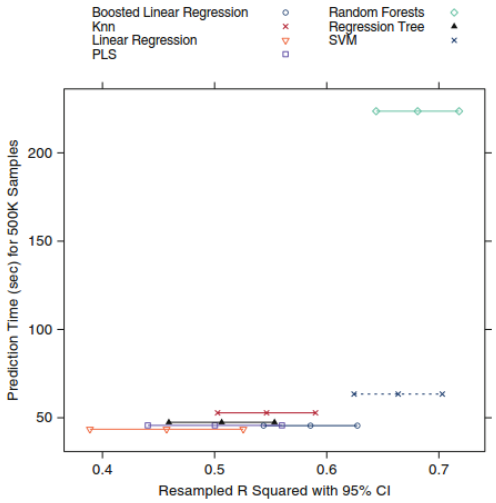


Figura 13, Tiempo y precisión de cada modelo de predicción

Importancia de la Ingeniería de características

Las características de los datos influirán directamente los modelos de predicción a utilizar y sobre los resultados que se pretenden.

Se puede decir que: cuanto mejor son las características que se deciden y preparan, mejores serán los resultados que pueden lograrse. Esto es cierto, pero también es engañoso.

Los resultados a alcanzar son un factor del modelo que se elige, los datos que se tienen disponibles y las características que se preparan. Incluso el encuadre del problema y las medidas objetivas que se estén utilizando para estimar la precisión juegan un papel esencial. Los resultados dependen de muchas propiedades interdependientes.

Se necesitan grandes características que describan las estructuras inherentes a los datos.

Mejores características significa flexibilidad

Se pueden elegir los modelos "equivocados" (menores al óptimo) y aun así obtener buenos resultados. La mayoría de los modelos se pueden recoger en una buena estructura de datos. La flexibilidad de las buenas características permitirá utilizar modelos menos complejos, que a la vez son más rápidos de procesar, más fácil de entender y fáciles de mantener. Esto es muy muy deseable.

Mejores características significa modelos más simples.

Con las características bien diseñadas, se puede elegir "los parámetros equivocados" (menos de óptimo) y aun así obtener buenos resultados, por las mismas razones. No se tiene que trabajar tan duro para recoger los modelos adecuados y los parámetros más optimizados.

Con unas buenas características, se está más cerca del problema de fondo y una representación de todos los datos que se tienen disponible y se podrían utilizar para caracterizar mejor el problema.

Mejores características significa mejores resultados.

2.3 Revisión de los métodos de modelado predictivo

2.3.1 Regresión Logística

En estadística, la regresión logística (logistic regression[14]) es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores.

En general, la regresión logística es adecuada cuando la variable de respuesta Y es politómica (admite varias categorías de respuesta, tales como mejora mucho, empeora, se mantiene, mejora, mejora mucho), pero es especialmente útil en particular cuando solo hay dos posibles respuestas (cuando la variable de respuesta es dicotómica), que es el caso más común.

La RL es una de las técnicas estadístico-inferenciales más empleadas en la producción científica contemporánea.

La identificación del mejor modelo de regresión logística se realiza mediante la comparación de modelos utilizando el cociente de verosimilitud, que indica a partir de los datos de la muestra cuanto más probable es un modelo frente al otro.

La diferencia de los cocientes de verosimilitud entre dos modelos se distribuye según la ley de la Chi-cuadrado con los grados de libertad correspondientes a la diferencia en el número de variables entre ambos modelos. Si a partir de este coeficiente no se puede demostrar que un modelo resulta mejor que el otro, se considerará como el más adecuado, el más sencillo.

Límite de decisión (Decision Boundary):

En un problema estadístico con clasificación de dos clases, un límite de decisión o de la superficie de decisiones es una frontera que divide el subyacente espacio vectorial en dos conjuntos, uno para cada clase.

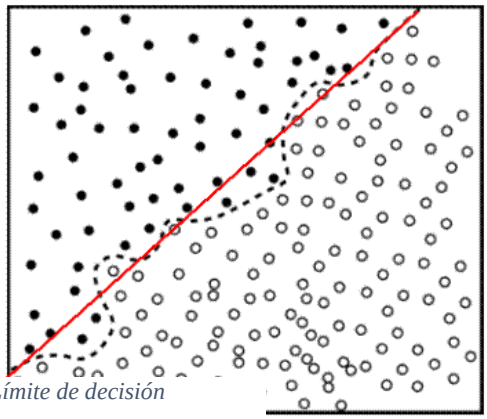


Figura 14, Límite de decisión

El clasificador clasificará todos los puntos en un lado del límite de decisión como pertenecientes a una clase y todos aquellos en el otro lado como perteneciente a la otra clase.

El límite de decisión es la región de un espacio del problema en el que la etiqueta de salida de un clasificador es ambigua.

Si la superficie de decisión es un hiperplano, a continuación, el problema de clasificación es lineal, y las clases son linealmente separables.

Los límites de decisión no siempre son claros o precisos. Es decir, la transición de una clase en el espacio de características a otro no es discontinua, pero si gradual. Este efecto es común en los algoritmos de clasificación lógica basada en difusos, donde la pertenencia a una clase u otra es ambigua.

Función Logística o Función Sigmoide *(Sigmoid Function)*

La **función logística**, **curva logística** o **curva en forma de S** es una función matemática que aparece en diversos modelos de crecimiento de poblaciones, propagación de enfermedades epidémicas y difusión en redes sociales. Dicha función constituye un refinamiento del modelo exponencial para el crecimiento de una magnitud. Modela la función sigmoidea de crecimiento de un conjunto *P*.

El estudio inicial de crecimiento es aproximadamente exponencial; al cabo de un tiempo, aparece la competencia entre algunos miembros de *P* por algún recurso crítico *K* ("cuello de botella") y la tasa de crecimiento disminuye; finalmente, en la madurez, el crecimiento se detiene.

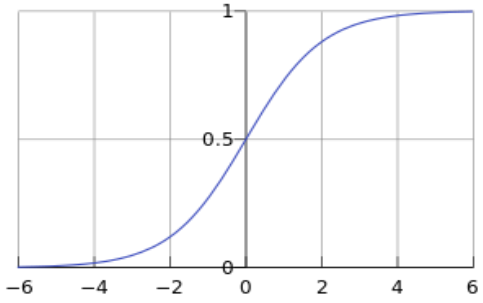


Figura 15, Función sigmoide

La función logística simple se define mediante la expresión matemática:

$$P(t) = \frac{1}{1 + e^{-t}}$$

donde la variable *P* puede ser considerada o denotada como *población*, donde *e* es la constante de Euler y la variable *t* puede ser considerada el *tiempo*.[1] Para valores de *t* en el rango de los números reales desde −∞ a +∞, la curva S se puede obtener. En la práctica, dada la naturaleza de la función exponencial, e^{-t}, es suficiente con computar *t* para un pequeño rango de números reales como pueden ser [−6, +6].

En su forma más general, la función logística se define por la fórmula matemática:

$$P(t; a, m, n, r) = a\frac{1 + me^{-t/r}}{1 + me^{-t/r}}$$

para parámetros reales *a*, *m*, *n*, y r. Estas funciones tienen un campo de aplicación muy amplio, desde la biología a la economía.

En estadística se emplea la función logística en el llamado análisis de regresión logística. Dicho análisis pretende estimar la probabilidad de un determinado evento, medible por variables categóricas (y no numéricas), que se sabe está correlacionado con ciertas variables cuantitativas. Por ejemplo en la

epidemiología y en la investigación de mecanismos lesionales es frecuente correlacionar la probabilidad de muerte o lesión con ciertos valores numéricos mediante una ecuación del tipo:

$$Prob_L = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1, i + \cdots + \beta_k x_{k,i})}}$$

Los datos empíricos constan de una lista de casos de los cuales se conocen una serie de indicadores numéricos para los cuales se examinó si presentaban lesión (o muerte), estos datos se representan usualmente como 0 (no-lesión) y 1 (lesión) y se estiman los parámetros β_i. El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados (*GLM*) que usa como función de enlace la función logit.

Factores de confusión

Durante el proceso de selección del modelo de regresión más adecuado, el que mejor se ajusta a los datos disponibles, hay que considerar un último aspecto adicional, especialmente si el proceso de selección de variables se hace mediante el método manual de obligar a que todas las variables entren en el modelo y es el propio investigador el que paso a paso va construyendo el modelo de regresión más conveniente.

Durante el proceso de incorporación de variables, al eliminar una variable de uno de los modelos de regresión estimados, hay que observar si en el modelo de regresión resultante al excluir esa variable, los coeficientes asociados al resto de variables introducidas en el modelo varían significativamente respecto al modelo de regresión que sí incluía dicha variable.

Si así sucede, significa que dicha variable podría ser un factor de confusión, al no mostrar una relación significativa con la variable que estamos estudiando directamente, pero sí indirectamente, al relacionarse con otras variables, que en sí mismas pueden estar significativamente relacionadas con la variable de estudio.

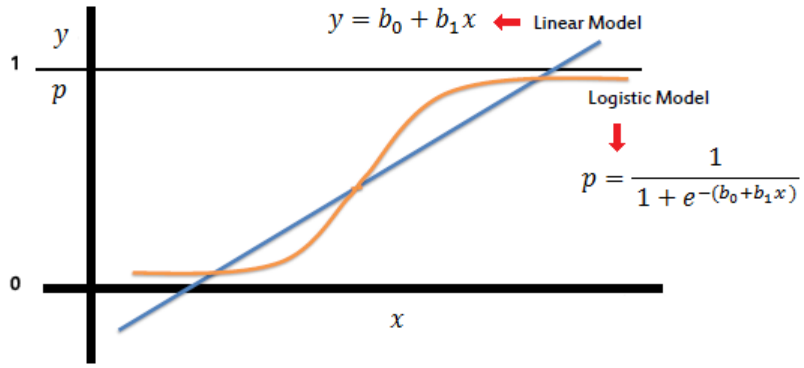


Figura 16, Modelo lineal vs Modelo logístico

En dicho caso, es conveniente no excluir la variable en cuestión del modelo de regresión, aunque no cumpla los requisitos para permanecer en él, obligando a que permanezca, de modo que aunque no se incluya su interpretación al evaluar los resultados del modelo, se ajusta el resultado del resto de variables seleccionadas por su posible efecto.

En la práctica, para incluir o no en la ecuación de regresión una variable de confusión, se utiliza el criterio (incorrectamente) de comprobar si su coeficiente correspondiente es significativamente diferente de cero, por lo que se mira sólo el valor de la probabilidad asociado a ese contraste. Sin embargo, no debe de ser la única razón, hay que considerar si su introducción en la ecuación modifica apreciablemente o no la relación entre la variable dependiente y el otro factor o factores estudiados.

En definitiva, la cuestión debe tratarse con enfoque clínico, puesto que hay que determinar desde ese punto de vista qué se considera como cambio apreciable en el coeficiente de la ecuación de regresión.

En esta línea, hay que tener cuidado con los términos relación, correlación o significación y causalidad. Que dos factores estén relacionados no implica de ninguna manera que uno sea causa del otro. Es muy frecuente que una alta dependencia indique que las dos variables dependen de una tercera que no ha sido medida (factor de confusión).

Concepto de interacción

Un concepto importante al construir un modelo de regresión es que pueden introducirse términos independientes únicos (una sola variable, por ejemplo efecto del tabaco) y además las interacciones entre variables de cualquier orden (efecto del tabaco según género), si se considera que pueden ser de interés o afectar a los resultados.

Al introducir los términos de interacción en un modelo de regresión es importante para la correcta estimación del modelo respetar un orden jerárquico, es decir siempre que se introduzca un término de interacción de orden superior (x•y•z), deben introducirse en el modelo los términos de interacción de orden inferior (x•y, x•z, y•z) y por supuesto los términos independientes de las variables que participan en la interacción (x, y, z).

Si se introducen en un modelo de regresión términos de interacción y resultan estadísticamente significativos, no se podrán eliminar del modelo los términos de interacción de orden inferiores ni los términos independientes de las variables que participan en la interacción para simplificarlo, deben mantenerse, aunque no resulten estadísticamente significativos.

El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados (GLM por sus siglas en inglés) que usa como función de enlace la función logit. Las probabilidades que describen el posible resultado de un único ensayo se modelan, como una función de variables explicativas, utilizando una función logística.

La regresión logística unidimensional puede usarse para tratar de correlacionar la probabilidad de una variable cualitativa binaria (asumiremos que puede tomar los valores reales "0" y "1") con una variable escalar x. La idea es que la regresión logística aproxime la probabilidad de obtener "0" (no ocurre cierto suceso) o "1" (ocurre el suceso) con el valor de la variable explicativa x. En esas condiciones, la probabilidad aproximada del suceso se aproximará mediante una función logística del tipo 1.

2.3.2 Máquinas de soporte vectorial

Las máquinas de soporte vectorial (vector support machines[15]) o máquinas de vectores de soporte (Support Vector Machines, SVMs) son un conjunto de algoritmos de aprendizaje supervisados y desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T.

Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra.

Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas a una u otra clase.

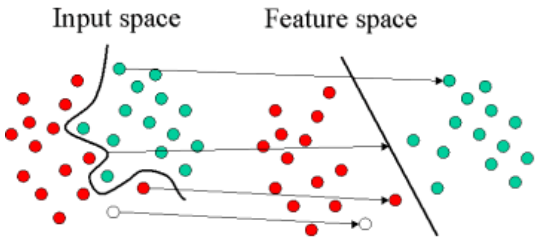


Figura 17, MSV, Espacio vectorial inicial y final

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector p-dimensional (una lista de p números).

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

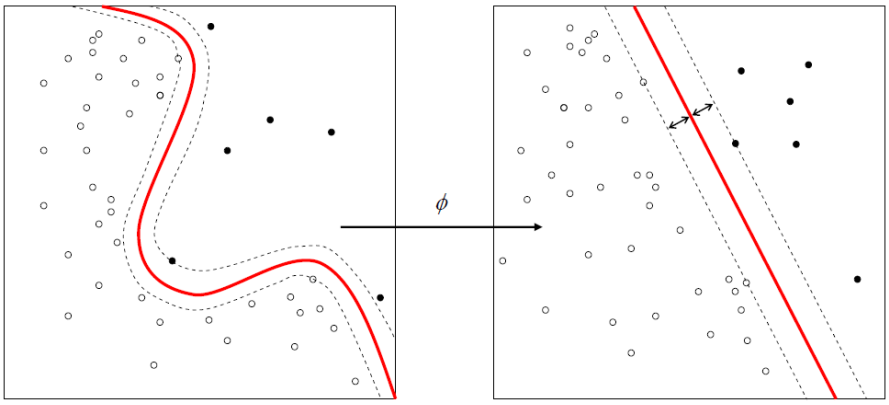


Figura 18, MSV, Dimensionalidad

En ese concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Los algoritmos SVM pertenecen a la familia de los clasificadores lineales. También pueden ser considerados un caso especial de la regularización de Tikhonov.

En la literatura de los SVMs, se llama atributo a la variable predictora y característica a un atributo transformado que es usado para definir el hiperplano. La elección de la representación más adecuada del universo estudiado, se realiza mediante un proceso denominado selección de características.

Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte.

Los modelos basados en SVMs están estrechamente relacionados con las redes neuronales. Usando una función kernel, resultan un método de entrenamiento alternativo para clasificadores polinomiales, funciones de base radial y perceptrón multicapa.

En el siguiente ejemplo idealizado para 2-dimensiones, la representación de los datos a clasificar se realiza en el plano x-y. El algoritmo SVM trata de encontrar un hiperplano 1-dimensional (en el ejemplo que nos ocupa es una línea) que une a las variables predictoras y constituye el límite que define si un elemento de entrada pertenece a una categoría o a la otra.

Existe un número infinito de posibles hiperplanos (líneas) que realicen la clasificación pero, ¿cuál es la mejor y cómo la definimos?

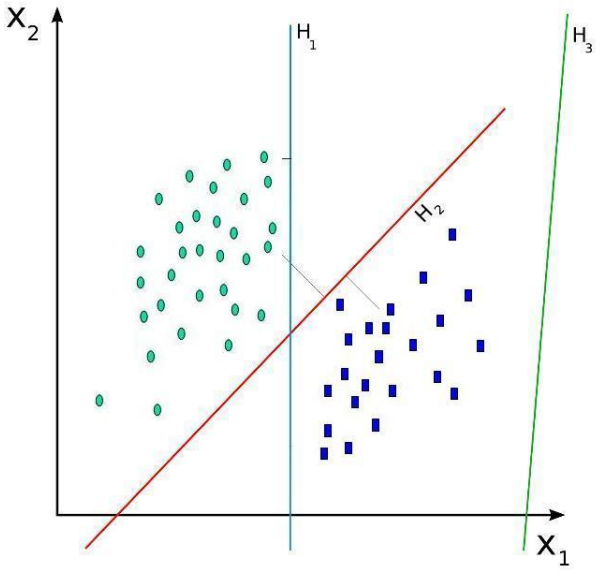


Figura 19, MSV, infinitos hiperplanos

Se denominan vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia entre ellas (margen) la mayor posible

2.3.3 Random forest

Random forest[16] es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging que construye una larga colección de árboles no correlacionados y luego los promedia.

En muchos problemas el rendimiento del algoritmo random forest es muy similar a la del boosting, y es más simple de entrenar y ajustar. Como consecuencia el random forests es popular y es ampliamente utilizado.

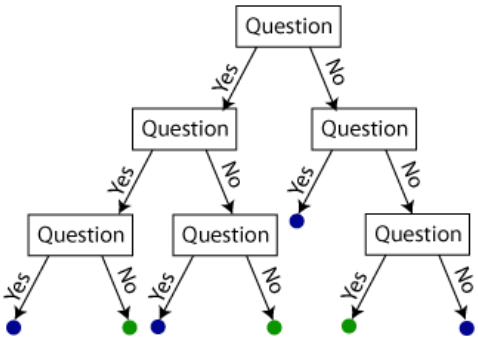


Figura 20, Árbol de Random Forests

La idea esencial del bagging es promediar muchos modelos ruidosos pero aproximadamente imparciales, y por tanto reducir la variación. Los árboles son los candidatos ideales para el bagging, dado que ellos pueden registrar estructuras de interacción compleja en los datos, y si crecen suficientemente profundo, tienen relativamente baja parcialidad. Producto de que los árboles son notoriamente ruidosos, ellos se benefician grandemente al promediar.

Cada árbol es construido usando el siguiente algoritmo:

1. Sea N el número de casos de prueba, M es el número de variables en el clasificador.
2. Sea m el número de variables de entrada a ser usado para determinar la decisión en un nodo dado; m debe ser mucho menor que M
3. Elegir un conjunto de entrenamiento para este árbol y usar el resto de los casos de prueba para estimar el error.
4. Para cada nodo del árbol, elegir aleatoriamente m variables en las cuales basar la decisión. Calcular la mejor partición a partir de las m variables del conjunto de entrenamiento.

Para la predicción un nuevo caso es empujado hacia abajo por el árbol. Luego se le asigna la etiqueta del nodo terminal donde termina. Este proceso es iterado por todos los árboles en el ensamblado, y la etiqueta que obtenga la mayor cantidad de incidencias es reportada como la predicción.

```
Select the number of models to build, m
for i = 1 to m do
    Generate a bootstrap sample of the original data
    Train a tree model on this sample
    for each split do
        Randomly select k (< P) of the original predictors
        Select the best predictor among the k predictors and
        partition the data
    end
    Use typical tree model stopping criteria to determine when a
    tree is complete (but do not prune)
end
```

Figura 21, Algoritmo básico de Random Forests

Características (o rasgos) y Ventajas

Las ventajas del random forests son:

- Es uno de los algoritmos de aprendizaje más certeros que hay disponible. Para un set de datos lo suficientemente grande produce un clasificador muy certero.
- Corre eficientemente en bases de datos grandes.
- Puede manejar cientos de variables de entrada sin excluir ninguna.
- Da estimados de qué variables son importantes en la clasificación.
- Tiene un método eficaz para estimar datos perdidos y mantener la exactitud cuando una gran proporción de los datos está perdida.
- Computa los prototipos que dan información sobre la relación entre las variables y la clasificación.
- Computa las proximidades entre los pares de casos que pueden usarse en los grupos, localizando valores atípicos, o (ascendiendo) dando vistas interesantes de los datos.
- Ofrece un método experimental para detectar las interacciones de las variables.

Desventajas de Random Forest

- Se ha observado que Random forests sobreajusta en ciertos grupos de datos con tareas de clasificación/regresión ruidosas.

- A diferencia de los árboles de decisión, la clasificación hecha por random forests es difícil de interpretar por el hombre.

- Para los datos que incluyen variables categóricas con diferente número de niveles, el random forests se parcializa a favor de esos atributos con más niveles. Por consiguiente, la posición que marca la variable no es fiable para este tipo de datos. Métodos como las permutaciones parciales se han usado para resolver el problema.

- Si los datos contienen grupos de atributos correlacionados con similar relevancia para el rendimiento, entonces los grupos más pequeños están favorecidos sobre los grupos más grandes.

2.3.4 Árboles de decisión

Un árbol de decisión (decision tree[17]) es un modelo de predicción utilizado en el ámbito de la inteligencia artificial. Dada una base de datos se construyen diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

Un árbol de decisión tiene unas entradas las cuales pueden ser un objeto o una situación descrita por medio de un conjunto de atributos y a partir de esto devuelve una respuesta la cual en últimas es una decisión que es tomada a partir de las entradas. Los valores que pueden tomar las entradas y las salidas pueden ser valores discretos o continuos.

Se utilizan más los valores discretos por simplicidad, cuando se utilizan valores discretos en las funciones de una aplicación se denomina clasificación y cuando se utilizan los continuos se denomina regresión.

Un árbol de decisión lleva a cabo un test a medida que este se recorre hacia las hojas para alcanzar así una decisión. El árbol de decisión suele contener nodos internos, nodos de probabilidad, nodos hojas y arcos. Un nodo interno contiene un test sobre algún valor de una de las propiedades.

Un nodo de probabilidad indica que debe ocurrir un evento aleatorio de acuerdo a la naturaleza del problema, este tipo de nodos es redondo, los demás son cuadrados.

Un nodo hoja representa el valor que devolverá el árbol de decisión y finalmente las ramas brindan los posibles caminos que se tienen de acuerdo a la decisión tomada.

En el diseño de aplicaciones informáticas, un árbol de decisión indica las acciones a realizar en función del valor de una o varias variables. Es una representación en forma de árbol cuyas ramas se bifurcan en función de los valores tomados por las variables y que terminan en una acción concreta. Se suele utilizar cuando el número de condiciones no es muy grande (en tal caso, es mejor utilizar una tabla de decisión).

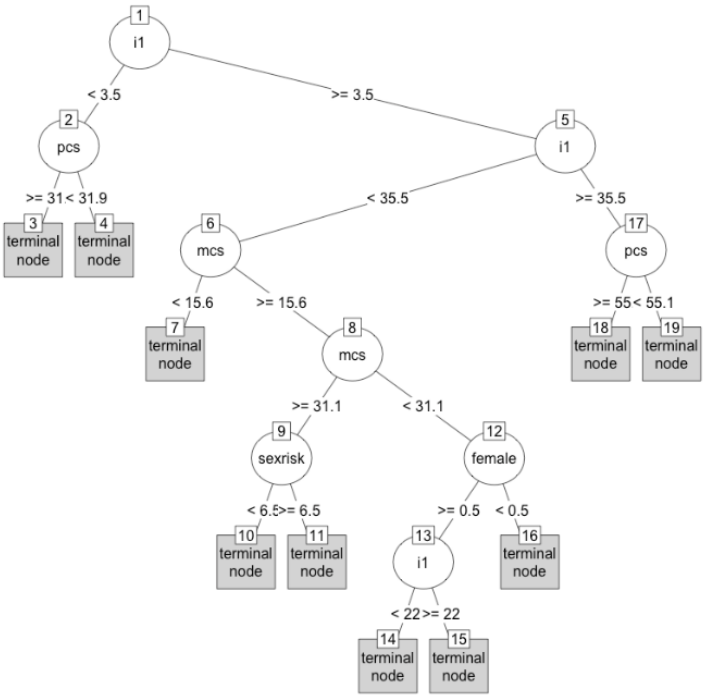


Figura 22, Ejemplo de árbol de decisión

De forma más concreta, refiriéndonos al ámbito empresarial, podemos decir que los árboles de decisión son diagramas de decisiones secuenciales nos muestran sus posibles resultados.

Éstos ayudan a las empresas a determinar cuáles son sus opciones al mostrarles las distintas decisiones y sus resultados. La opción que evita una pérdida o produce un beneficio extra tiene un valor. La habilidad de crear una opción, por lo tanto, tiene un valor que puede ser comprado o vendido.

2.3.5 Redes neuronales

Como se ha visto con otros métodos de clasificación, las clases se pueden codificar en columnas binarias de variables ficticias y luego ser utilizadas como resultados para el modelo.

Aunque la discusión previa en redes neuronales[18] para la regresión utiliza una única respuesta, el modelo se puede manejar fácilmente con múltiples salidas, tanto para la regresión como para la clasificación.

La figura 23[7] muestra un diagrama de la arquitectura de modelo para la clasificación. En lugar de una sola salida la capa inferior tiene múltiples nodos para cada clase. Debe tenerse en cuenta que, a diferencia de las redes neuronales para la regresión, una transformación no lineal adicional puede utilizarse en la combinación de unidades ocultas.

Cada clase es predicha por una combinación lineal de las unidades ocultas que han sido transformadas para estar entre cero y uno (por lo general por una función sigmoidal). Sin embargo, a pesar de que las predicciones son entre cero y uno (debido a la función sigmoidal extra[19]), no son "probabilidad " ya que no se suman a uno.

¿Qué debe optimizar la red neuronal a fin de encontrar las estimaciones de los parámetros adecuados? Para la regresión, la suma de los errores cuadráticos y, para este caso, sería alterado manejar múltiples salidas por la acumulación de los errores a través de muestras y de las clases.

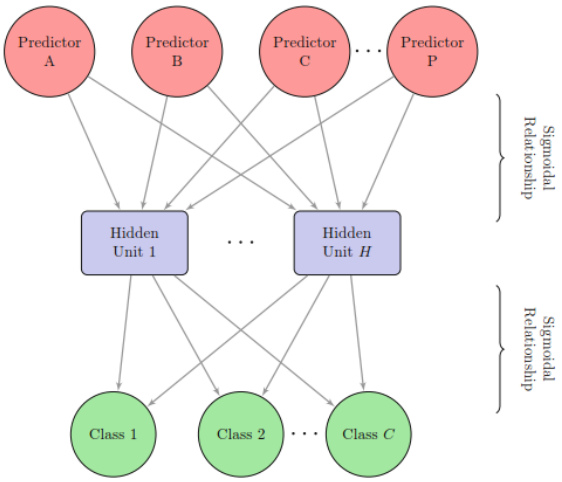


Figura 23, Redes Neuronales

Al igual que sus homólogos de regresión, las redes neuronales de clasificación tienen un potencial significativo. Al optimizar las sumas de cuadrados de error o entropía, la decadencia de peso atenúa el tamaño de las estimaciones de los parámetros[20]. Esto puede conducir a clasificaciones mucho más suaves.

También, como se mencionó anteriormente, el modelo promediado ayuda a reducir el exceso de overfitting.

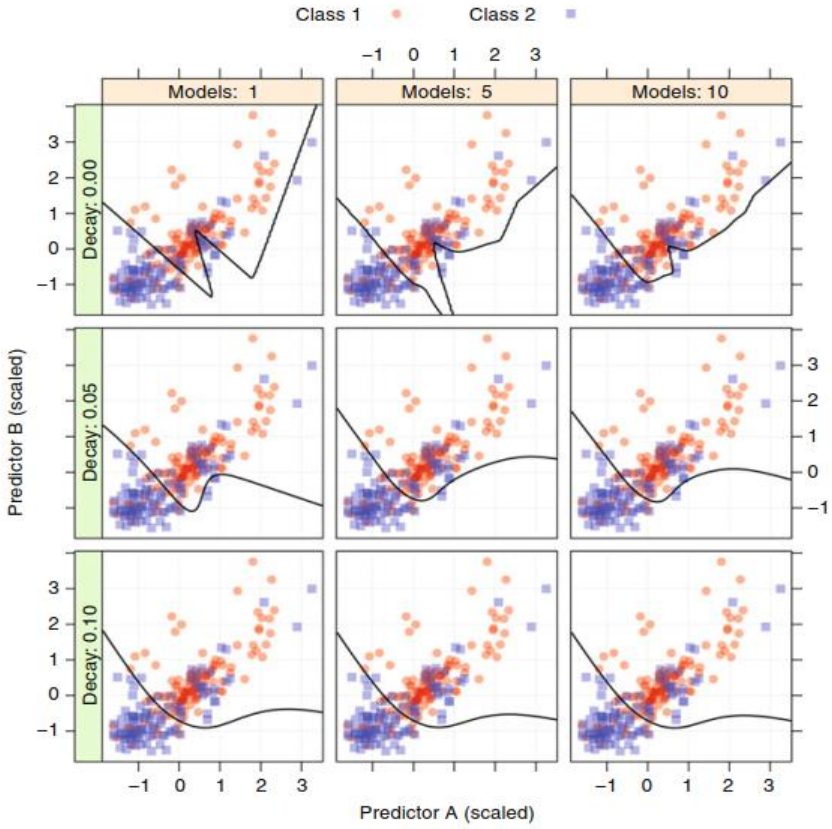


Figura 24, Ejemplo de modelos ficticios, redes neuronales

La figura anterior muestra ejemplos de modelos ficticios con cantidades diferentes de la decadencia de peso y modelo de promedio. Cada modelo se inició con la misma semilla aleatoria, utiliza tres unidades ocultas, y se ha optimizado para las sumas de cuadrado.

La primera fila de los modelos sin deterioro de peso muestra un significativo overfitting y, en estos casos, el modelo de promedio tiene un impacto marginal. La pequeña cantidad de descomposición en la segunda fila muestra una mejora (como lo hace el modelo de promedio) pero es todavía una sobre-adaptación a los datos de entrenamiento cuando se utiliza una única red.

La cantidad más alta de la decadencia de peso mostró los mejores resultados con prácticamente ningún impacto sobre el modelo de promedio. Para estos datos, un modelo único con la decadencia de peso es probablemente la mejor opción, ya que es computacionalmente menos costosa.

Muchos otros aspectos de los modelos de clasificación de la red neuronal los reflejan sus homólogos de regresión. Aumentar el número de predictores o unidades ocultas siguen dando lugar a un gran número de parámetros del modelo y las mismas rutinas numéricas. La colinariedad y los predictores no informativos tendrán un impacto notable sobre el rendimiento del modelo.

2.4 Aplicación de los métodos de predicción

2.4.1 Overfitting & Undefitting

Overfitting[21] o sobreajuste ocurre cuando un modelo estadístico o algoritmo de máquina de aprendizaje (Learning Machine) capta el ruido de los datos. Intuitivamente, el sobreajuste se produce cuando el modelo o el algoritmo ajustan los datos demasiado bien. Específicamente, el sobreajuste ocurre si el modelo o algoritmo muestra sesgo bajo pero alta varianza.

El sobreajuste es a menudo el resultado de un modelo excesivamente complicado, y se puede prevenir mediante el ajuste de múltiples modelos y el uso de la validación o la validación cruzada (Crossvalidation) para comparar su precisión de predicción en los datos de prueba.

Underfitting ocurre cuando un modelo estadístico o algoritmo de aprendizaje automático no puede capturar la tendencia subyacente de los datos. Intuitivamente, se produce underfitting cuando el modelo o algoritmo no se ajusta a los datos bastante bien. Específicamente, underfitting ocurre si el modelo o algoritmo muestra baja varianza pero alta sesgo. El underfitting es a menudo el resultado de un modelo excesivamente simple.

Ambos, tanto Overfitting como Underfitting conducen a predicciones pobres en nuevos conjuntos de datos.

En estadística y learning machine, no suele encontrarse underfitting muy a menudo. Los conjuntos de datos que se utilizan para el modelado predictivo hoy en día a menudo vienen con demasiados predictores, no demasiado pocos. No obstante, cuando la construcción de cualquier modelo de máquina de aprendizaje para el modelado predictivo, es conveniente usar la validación de uso o la validación cruzada para evaluar la exactitud de predicción, sobre todo si se está tratando de evitar el sobreajuste o underfitting.

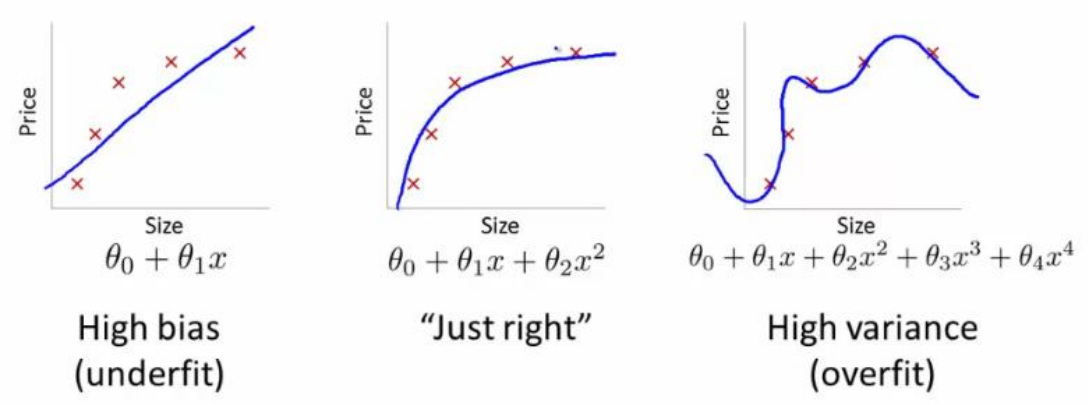


Figura 25, Underfitting & Overfitting

2.4.2 Evaluation and Crossvalidation

En las estadísticas, la **evaluación** y validación de modelos de regresión es el proceso de decidir si los resultados numéricos que cuantifican las relaciones hipotéticas entre variables, obtenidos a partir de un análisis de regresión, son aceptables como descripciones de los datos.

El proceso de validación puede implicar el análisis de la bondad de ajuste de la regresión, el análisis de si los residuos de la regresión son aleatorios, y comprobar si el rendimiento predictivo del modelo se deteriora sustancialmente cuando se aplica a datos que no fueron utilizados en la estimación del modelo.

La validación cruzada[25] o **crossvalidation**, a veces llamada la estimación de rotación, es un modelo de validación técnica para evaluar cómo los resultados de una estadística de análisis se generalizan a un conjunto de datos independientes. Se utiliza principalmente en entornos en los que el objetivo es la predicción, y uno quiere estimar con precisión un modelo predictivo se presentará en la práctica.

El objetivo de la validación cruzada es definir un conjunto de datos para "probar" el modelo en la fase de formación (es decir, el conjunto de datos de validación), con el fin de limitar los problemas como sobreajuste (overfitting), proporcionando una idea de cómo el modelo se generalizará a un conjunto de datos independiente (es decir, un conjunto de datos desconocido, por ejemplo de un problema real), etc.

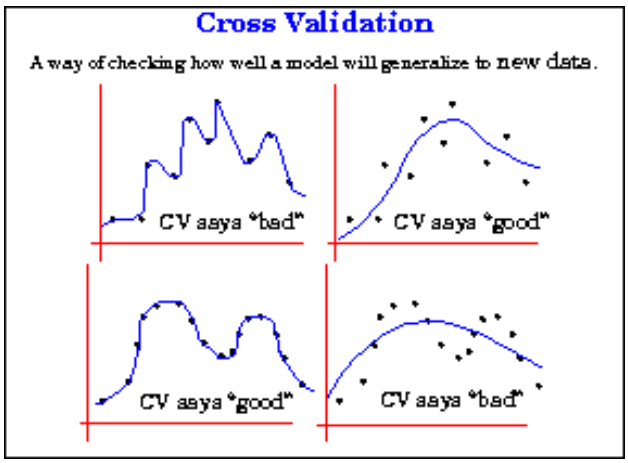


Figura 26, Validación cruzada

La validación cruzada[26] es importante en la protección contra la comprobación de hipótesis sugeridas por los datos (llamados " errores de tipo III "), especialmente cuando otras muestras son peligrosas, costosas o imposibles de conseguir.

Además, una de las principales razones para el uso de validación cruzada en lugar de utilizar la validación convencional (por ejemplo particionar el conjunto de datos en dos conjuntos de 70% para la formación y 30% para la prueba) es que el error (por ejemplo, Root Mean Square Error) en el conjunto de entrenamiento en la validación convencional no es un estimador útil del desempeño del modelo y por lo tanto el error en el conjunto de datos de prueba no representa adecuadamente la evaluación del desempeño del modelo.

En resumen, la validación cruzada combina (promedios) medidas de ajuste (error de predicción) para corregir la naturaleza optimista del error de entrenamiento y deriva en una estimación más precisa del rendimiento del modelo de predicción.

2.4.3 Embolsado y Boosting

El **embolsado**, también llamado bagging, es una máquina de aprendizaje diseñada para mejorar la estabilidad y la precisión de los algoritmos de aprendizaje automático utilizados en clasificación estadística y regresión. También reduce la varianza y ayuda a evitar overfitting.

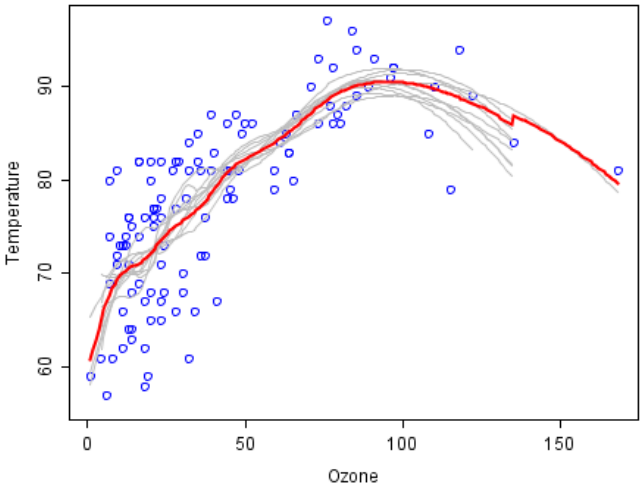


Figura 27, Ejemplo de embolsado

A pesar de que se aplica generalmente a los métodos de árboles de decisión, puede ser utilizado con cualquier tipo de método. El embolsado es un caso especial del enfoque de modelo de promedio.

El **Impulso** o Boosting es una máquina de aprendizaje utilizada para reducir la oblicuidad y la varianza en el aprendizaje, y una familia de algoritmos de aprendizaje automático que convierten los valores débiles en fuertes. Al algoritmo se basa en la siguiente pregunta:

¿Puede un grupo de datos débiles crear un solo dato fuerte?

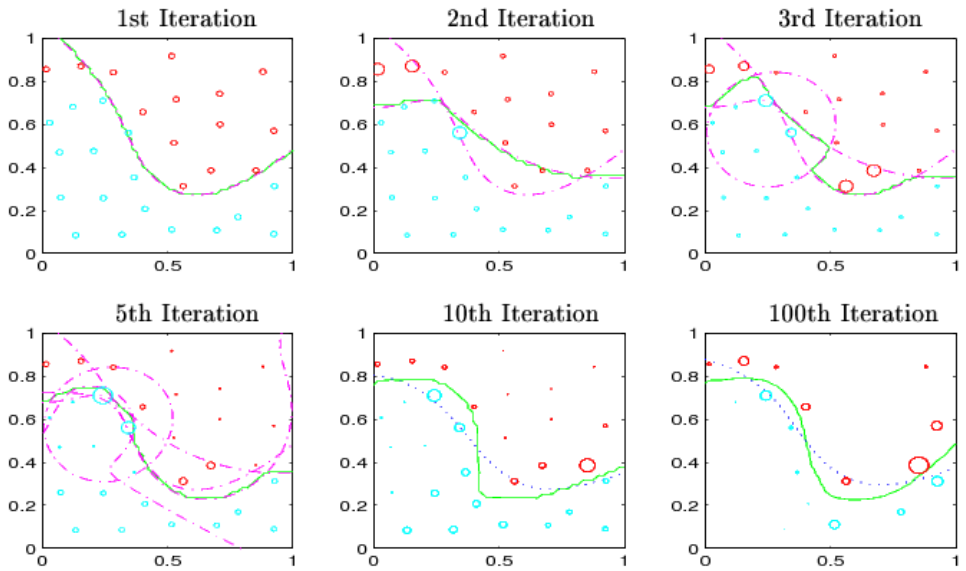


Figura 28, Iteraciones de boosting

3. CASO DE ESTUDIO, ANÁLISIS DE RETRASOS EN EL AEROPUERTO DE SEATTLE-TACOMA

3.1 Análisis Situacional

Para el caso de estudio de este Proyecto, se ha seleccionado el aeropuerto estadounidense de Seattle-Tacoma Internacional (Washington) gracias a la libre publicación estadística de sus datos, de vital importancia para la creación de los experimentos de modelado predictivo realizados a continuación. Los datos utilizados corresponden al mes de Enero de 2014 en dicho aeropuerto.

El Aeropuerto Internacional de Seattle-Tacoma (IATA: SEA, OACI: KSEA) es un aeropuerto situado entre las ciudades de Seattle y Tacoma en la ciudad de SeaTac en el estado de Washington. Este aeropuerto es uno de los aeropuertos más grandes de Estados Unidos con un tráfico de más de 30 millones de pasajeros cada año.

En 2010, sirvió a más de 31.5 millones de pasajeros, lo que lo ubica como el décimo-octavo aeropuerto más ocupado en los Estados Unidos.

Estadísticas (2010):

Operaciones aéreas	313,954
Movimiento de pasajeros	31,553,166

En Washington, las aerolíneas más grandes de la nación registraron una tasa de puntualidad de un 76.9 por ciento en mayo, por debajo del 79,4 por ciento tanto en el índice de puntualidad en mayo de 2013 y el 79,6 por ciento índice de puntualidad publicado en abril de 2014.

Además, las compañías han cancelado un 1,9 por ciento de sus vuelos nacionales regulares en mayo, un aumento de la tasa de cancelación de un 1,1 por ciento publicado tanto en mayo de 2013 como en abril de 2014.

Además, en mayo, cuatro aerolíneas reportaron demoras en pista de más de tres horas en vuelos nacionales y un retraso en asfalto de más de cuatro horas en un vuelo internacional, según el *U.S. Department of Transportation's Air Travel Consumer Report*.

Todos los Estados y las compañías aéreas extranjeras que operan por lo menos una aeronave con 30 o más asientos para pasajeros deben informar de los largos retrasos en Tarmac en los aeropuertos de Estados Unidos.

Todas las demoras en la pista reportadas son objeto de investigación por el Departamento. El informe del consumidor también incluye datos sobre retrasos de vuelos y las causas de los retrasos presentadas con la *Department's Bureau of Transportation Statistics* por las compañías de informes.

Además, el informe del consumidor contiene información sobre los informes de mal manejo de equipaje presentados por los consumidores con los transportistas y las quejas de servicios aerolínea recibidas por la *Department's Aviation Consumer Protection Division*.

El informe del consumidor también incluye informes de incidentes relacionados con la pérdida, la muerte o lesiones de las mascotas que viajan en avión, según sea necesario para ser presentada por las compañías estadounidenses.

Vuelos con retraso crónico

A finales de mayo, hubo 53 vuelos que se retrasaron crónicamente - más de 30 minutos tarde más de 50 por ciento del tiempo - por dos meses consecutivos. No hubo vuelos retrasados crónicamente durante tres meses consecutivos o más. Una lista de los vuelos que se retrasó crónicamente para un solo mes está disponible desde Bureau of Transportation Statistics.

Las causas de los retrasos de vuelos

En mayo, las compañías que presentaron los datos de puntualidad informaron que:

- Un 6,04 por ciento de sus vuelos se retrasaron por los retrasos del sistema aeronáutico general, en comparación con 5,75 en abril.
- Un 8,18 por ciento en los aviones tardó en llegar, en comparación con el 7,47 por ciento en abril.
- Un 6,04 por ciento por factores dentro del control de la aerolínea, como de mantenimiento o de problemas de tripulación, en comparación con 5,34 por ciento en abril.
- Un 0,57 por ciento por el clima extremo, en comparación con 0,40 por ciento en abril.
- Un 0,03 por ciento por razones de seguridad, en comparación con el 0,02 por ciento en abril.

El tiempo es un factor tanto en la categoría de tiempo-extremo y la categoría de sistema de aviación. Esto incluye los retrasos debidos al re-enrutamiento de los vuelos por la *DOT's Federal Aviation Administration* en consulta con las compañías involucradas. El tiempo es también un factor en los retrasos atribuidos a la llegada tardía de las aeronaves, aunque las aerolíneas no informan de las causas específicas de dicha categoría.

Los datos recogidos por la BTS también muestra el porcentaje final de los vuelos retrasados por el clima, incluidos los reportados tanto en la categoría de clima extremo tanto en los retrasos del *National Aviation System.*

- En mayo, el 33,49 por ciento de finales de los vuelos se retrasaron por el clima, por encima del 26,62 por ciento en abril y por debajo del 39,01 por ciento en mayo de 2013.

Las quejas sobre el servicio de las aerolíneas

En mayo, el Departamento recibió 1.280 quejas sobre el servicio de las aerolíneas por los consumidores, hasta el 31,3 por ciento de las 975 quejas presentadas en mayo de 2013, y un 1,7 por ciento de las 1259 recibidas en abril de 2014.

Extrayendo datos del Air Travel Consumer report de Julio de 2014, se describen a continuación las estadísticas clave en tiempo de rendimiento y cancelaciones de vuelo, basadas en datos presentados ante la Oficina de Estadísticas de Transporte de los 14 informes de compañías aéreas y datos de Tarmac.

Total

76.9 por ciento de puntualidad

Puntualidad más alta

1. Hawaiian Airlines - 93,2 por ciento
2. Alaska Airlines - 89,7 por ciento
3. Delta Air Lines - 84,4 por ciento

Puntualidad más baja

1. ExpressJet Airlines - 70,3 por ciento
2. Enviado Aire - 71,4 por ciento
3. Southwest Airlines - 71,8 por ciento

Vuelos nacionales con retrasos Tarmac superiores a tres horas

1. United Airlines 1426 de Los Ángeles a Houston, 09/05/14 - diferido en asfalto 222 minutos
2. United Airlines 1631 de Newark a Tampa, 23/05/14 - diferido en asfalto 217 minutos
3. United Airlines 1435 de Chicago O'Hare a Santa Ana, California, 12/05/14 -. Retrasó en asfalto 193 minutos
4. United Airlines vuelo 687 de Chicago O'Hare a Portland, Oregon, 12.05.14 -. Retrasó en asfalto 183 minutos

Vuelos internacionales con retrasos Tarmac superiores a cuatro horas

1. Vuelo ExpressJet Aerolíneas 4475 de Monterrey, México a Houston, 09/05/14 - diferido en asfalto 261 minutos

Tasas más altas de Vuelos cancelados

1. ExpressJet Airlines - 5.8 por ciento
2. Enviado Aire - 4.3 por ciento
3. JetBlue Airways - 2.4 por ciento

Caso de estudio real.

El caso de estudio descrito a continuación consta de la siguiente estructura:

En primer lugar se procede a crear una base de datos con los datos extraídos de la agencia de estadística de estados unidos para el aeropuerto de Seatte-Tacoma y se pre-procesan los datos con el fin de suavizarlos y prepararlos para su uso. A continuación se realiza una selección de características para seleccionar aquellos datos y variables que nos aporten la información necesaria para el modelado predictivo. Después de esto, se crea un algoritmo de modelado predictivo de regresión logística y otro de Gradient Boosting Machine, además de un modelo aplicado a la vida real con datos censurados. Por último, se analizan los resultados de los modelos creados mediante ROC y AUC.

3.2 Creación de la Base de Datos

Para el correcto análisis de los datos esenciales para el estudio y predicción de retrasos, es necesario crear e importar una batería de datos recopilados en el tiempo. En este caso, se va a proceder a crear una base de datos con los parámetros recogidos en el aeropuerto de Washington durante Enero de 2014.

A continuación se describen los parámetros incluidos en la base de datos:

YEAR: Año en el que se ha recopilado cada una de las muestras del estudio. En este caso todas las muestras coinciden en el año 2014, sin embargo para estudios de mayor magnitud y mayor amplitud temporal es necesario para acotar la variable temporal de los datos.

MONTH: Mes en el que se han recopilado cada una de las muestras del estudio. Este parámetro varía en una escala del 1 al 12, siendo el 1 el mes de Enero y 12 el mes de Diciembre. Como en el caso anterior todas las muestras coinciden en el mes de Enero, sin embargo para estudios de mayor magnitud y mayor amplitud temporal es necesario para acotar la variable temporal de los datos.

DAY_OF_MONTH: Día del mes en el que se ha recopilado cada muestra del estudio. Este parámetro varía en una escala del 1 al 31, coincidiendo con la numeración del día mensual del dato.

DAY_OF_WEEK: Día de la semana en el que se ha recopilado cada muestra del estudio. Este parámetro varía en una escala del 1 al 7, siendo el 1 el número equivalente al Lunes y el 7 el equivalente al Domingo. Tanto este dato como el anterior pueden ser necesarios para estimar que día de la semana (p.ej. el inicio del fin de semana) o que día del mes (p.ej. principios de mes) se estima que puede haber más congestión y por ende más retrasos en los vuelos.

DEP_TIME: Departure time. Hora de salida del vuelo en hora local real (LMT) expresado como un conjunto de cuatro dígitos, siendo los dos primeros la hora y los dos últimos los minutos. Tiempo computado en el momento en el que la aeronave abandona la puerta o gate. Dato esencial para calcular y prever los colapsos en función de la hora del día, horas punta y picos de trabajo.

CRS_DEP_TIME: Computerized Reservations Systems (CRS) departure time. Tiempo que cada aerolínea tiene computado como tiempo de salida. Puede definirse como la hora de salida que la compañía estima a la que va a salir cada vuelo. Dato muy útil para comprobar la diferencia entre la hora real de salida y la hora estimada por la compañía. Indicador de la "on-time performace" de cada aerolínea.

ARR_TIME: Arrival time. Hora de llegada del vuelo en hora local real (LMT) expresado como un conjunto de cuatro dígitos, siendo los dos primeros la hora y los dos últimos los minutos. Tiempo computado en el momento en el que la rueda de morro de la aeronave golpea el Tarmac. Dato esencial para calcular y prever los colapsos en función de la hora del día, horas punta y picos de trabajo.

CRS_ARR_TIME: Computerized Reservations Systems (CRS) arrival time. Tiempo que cada aerolínea tiene computado como tiempo de llegada. Puede definirse como la hora de salida que la compañía estima a la que va a llegar cada vuelo. Dato muy útil para comprobar la diferencia entre la hora real de llegada y la hora estimada por la compañía. Indicador de la "on-time performace" de cada aerolínea.

UNIQUE_CARRIER: Nombre en siglas de la compañía aérea de la que se recoge la muestra de datos. Dato necesario para asociar cada retraso con cada aerolínea y poder estimar y predecir que compañías producen más retrasos. Cadena de caracteres.

FL_NUM: Flight Number. Número de vuelo de cada muestra.

TAIL_NUM: Tail Number. Conjunto de números y letras que componen el registro de matrícula de cada aeronave, situados en la cola del avión. Este número es único para cada aeronave y su formato depende del país de matriculación (p.ej. en el caso de EEUU está compuesto de la letra November más un conjunto de dígitos). Puede encontrarse tanto en la parte posterior y anterior del avión como bajo los planos.

ACTUAL_ELAPSED_TIME: Tiempo real de vuelo, desde la salida en origen hasta la llegada en destino, expresado en minutos como un conjunto de enteros. Dato útil para la estimación e identificación de comportamientos o retrasos en ruta.

CRS_ELAPSED_TIME: Computerized Reservations Systems (CRS) elapsed time. Tiempo que cada aerolínea tiene computado en su sistema como tiempo de vuelo, desde la salida en origen hasta la llegada en destino, expresado en minutos como un conjunto de enteros. Dato útil para la estimación e identificación de comportamientos o retrasos en ruta y la diferencia con el tiempo real de vuelo para el cálculo de la on-time performance.

AIR_TIME: Tiempo real de vuelo en aire, desde la salida en origen hasta la llegada en destino, contabilizado desde que el avión abandona la pista hasta vuelve a entrar en contacto con ella, expresado en minutos como un numero entero. Dato útil para la estimación e identificación de comportamientos o retrasos en ruta.

ARR_DELAY: Diferencia entre la hora de llegada de CRS o estimada por la compañía y la hora real de llegada en destino, expresada en minutos como un numero entero. Información útil para identificar el origen de un problema o retraso.

DEP_DELAY: Diferencia entre la hora de salida de CRS o estimada por la compañía y la hora real de salida en origen, expresada en minutos como un numero entero. Información útil para identificar el origen de un problema o retraso.

ORIGIN: Código del aeropuerto de origen del vuelo, expresado como una cadena de caracteres de tres letras. Código único para cada aeropuerto estandarizado por la Asociación Internacional de Transporte Aéreo (IATA).

DEST: Código del aeropuerto de destino del vuelo, expresado como una cadena de caracteres de tres letras. Código único para cada aeropuerto estandarizado por la Asociación Internacional de Transporte Aéreo (IATA).

DISTANCE: Distancia entre los aeropuertos origen y destino, expresada en millas náuticas como un numero entero. Dato útil para estimas la relación entre los retrasos y la distancia de vuelo entre varios saltos.

TAXI_IN: Taxi in time. Tiempo de rodadura hasta puerta en el aeropuerto de destino. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos en las fases y operativas de rodadura.

TAXI_OUT: Taxi out time. Tiempo de rodadura desde la puerta hasta la pista en el aeropuerto de origen. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos en las fases y operativas de rodadura.

CANCELLED: Identificador de vuelo cancelado. Expresado como un valor binario, siendo cero si el vuelo se ha efectuado o uno si ha sido cancelado. Cuando este valor es uno se omiten el resto de valores temporales.

CANCELLATION_CODE: Código de cancelación. Expresado con un carácter alfabético en función de la causa de cancelación siguiendo la siguiente leyenda: A-"Carrier", B-"Weather", C-"National Air System", D-"Security".

DIVERTED: Indicador de vuelo desviado. Expresado como un valor binario, siendo cero si el vuelo se ha efectuado con normalidad o uno si ha sido desviado.

CARRIER_DELAY: Retraso causado por la portadora. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos causados por este factor.

WEATHER_DELAY: Retraso causado por inclemencias meteorológicas. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos causados por este factor.

NAS_DELAY: Retraso causado por la National Air System. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos causados por este factor.

SECURITY_DELAY: Retraso causado por los controles de seguridad o sus inspecciones. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos causados por este factor.

LATE_AIRCRAFT_DELAY: Retraso por llegada tarde de la aeronave. Expresado como un conjunto de enteros en minutos. Dato útil para identificar retrasos causados por este factor.

Una vez identificados los parámetros de la base de datos, se ha procedido a importarla a R. Para ello, se ha utilizado el archivo CSV (coma separated) Washington2014January.csv y se ha importado manteniendo el nombre de sus parámetros a sqlite manager.

3.3 Importación a R

Con el propósito de mejorar el manejo y la gestión de los datos, se ha decidido crear una Base de Datos en SQLite Manager de Firefox.

Para ello, se ha creado en dicho programa una nueva Base de datos llamada "ontime". A continuación, se ha creado una tabla con el mismo nombre cuyos campos coincidan con los campos del archivo Washington2014January.csv con la opción Ejecutar SQL.

```
create table ontime (
    Year int,
    Month int,
    DayofMonth int,
    DayofWeek int,
    DepTime int,
    CRSDepTime int,
    ArrTime int,
    UniqueCarrier varchart(5),
    FlightNum int,
    TailNum varchart(8),
    ActualElapsedTime int,
    CRSElapsedTime int,
    AirTime int,
    ArrDelay int,
    DepDelay int,
    Origin varchart(3),
    Dest varchart(3),
    Distance int,
    TaxiIn int,
    TaxiOut int,
    Cancelled int,
    CancellationCode varchart(1),
    Diverted varchart(1),
    CarrierDelay int,
    WeatherDelay int,
    NASDelay int,
    SecurityDelay int,
    LateAircraftDelay int);
```

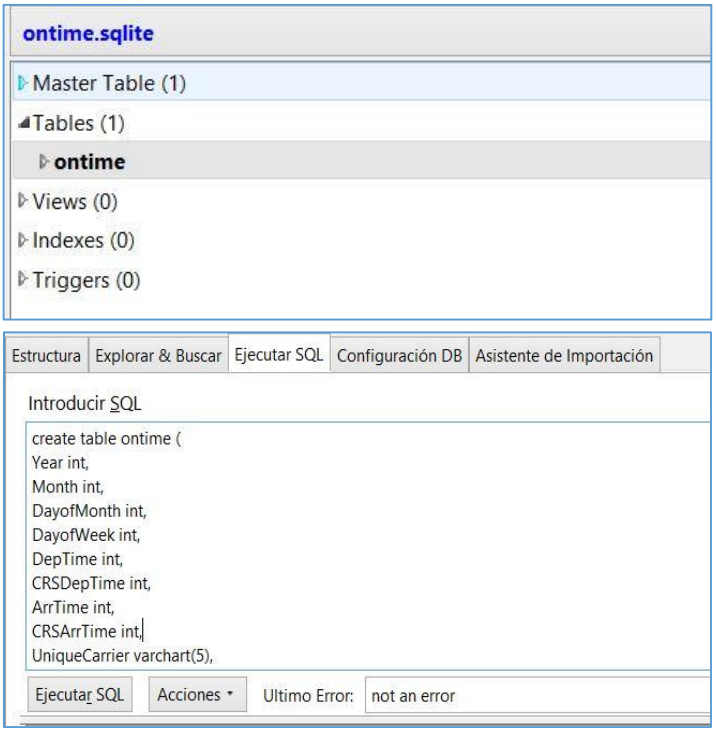
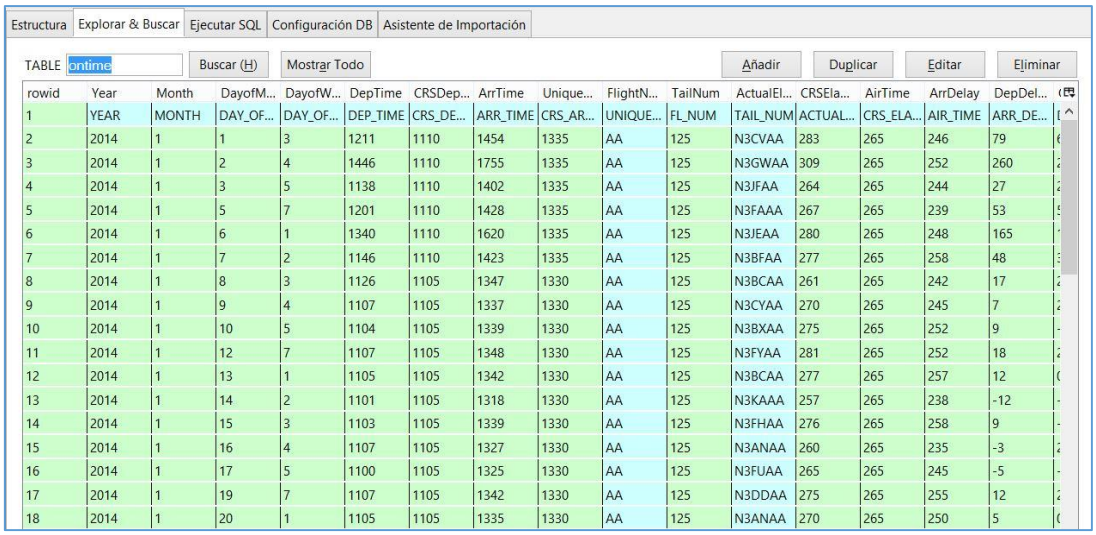


Figura 29, Creación de la BBDD

Se importan los datos del archivo Washington2014January.csv dentro de la tabla ontime utilizando la opción Base de Datos -> Importar. También se ha suprimido la primera fila que contenía datos irrelevantes de parámetros de índices.

rowid	Year	Month	DayofM...	DayofW...	DepTime	CRSDep...	ArrTime	Unique...	FlightN...	TailNum	ActualEl...	CRSEla...	AirTime	ArrDelay	DepDel...
1	YEAR	MONTH	DAY_O...	DAY_O...	DEP_TI...	CRS_DE...	ARR_TI...	CRS_AR...	UNIQU...	FL_NUM	TAIL_N...	ACTUA...	CRS_EL...	AIR_TIME	ARR_D...
2	2014	1	1	3	1211	1110	1454	1335	AA	125	N3CVAA	283	265	246	79



rowid	Year	Month	DayofM...	DayofW...	DepTime	CRSDep...	ArrTime	Unique...	FlightN...	TailNum	ActualEl...	CRSEla...	AirTime	ArrDelay	DepDel...
1	YEAR	MONTH	DAY_OF...	DAY_OF...	DEP_TIME	CRS_DE...	ARR_TIME	CRS_AR...	UNIQUE...	FL_NUM	TAIL_NUM	ACTUAL...	CRS_ELA...	AIR_TIME	ARR_DE...
2	2014	1	1	3	1211	1110	1454	1335	AA	125	N3CVAA	283	265	246	79
3	2014	1	2	4	1446	1110	1755	1335	AA	125	N3GWAA	309	265	252	260
4	2014	1	3	5	1138	1110	1402	1335	AA	125	N3JFAA	264	265	244	27
5	2014	1	5	7	1201	1110	1428	1335	AA	125	N3FAAA	267	265	239	53
6	2014	1	6	1	1340	1110	1620	1335	AA	125	N3JEAA	280	265	248	165
7	2014	1	7	2	1146	1110	1423	1335	AA	125	N3BFAA	277	265	258	48
8	2014	1	8	3	1126	1105	1347	1330	AA	125	N3BCAA	261	265	242	17
9	2014	1	9	4	1107	1105	1337	1330	AA	125	N3CYAA	270	265	245	7
10	2014	1	10	5	1104	1105	1339	1330	AA	125	N3BXAA	275	265	252	9
11	2014	1	12	7	1107	1105	1348	1330	AA	125	N3FYAA	281	265	252	18
12	2014	1	13	1	1105	1105	1342	1330	AA	125	N3BCAA	277	265	257	12
13	2014	1	14	2	1101	1105	1318	1330	AA	125	N3KAAA	257	265	238	-12
14	2014	1	15	3	1103	1105	1339	1330	AA	125	N3FHAA	276	265	258	9
15	2014	1	16	4	1107	1105	1327	1330	AA	125	N3ANAA	260	265	235	-3
16	2014	1	17	5	1100	1105	1325	1330	AA	125	N3FUAA	265	265	245	-5
17	2014	1	19	7	1107	1105	1342	1330	AA	125	N3DDAA	275	265	255	12
18	2014	1	20	1	1105	1105	1335	1330	AA	125	N3ANAA	270	265	250	5

Figura 30, Vista de la BBDD

Por último, se han añadido índices para acelerar y facilitar el acceso a los datos con los siguientes comandos:

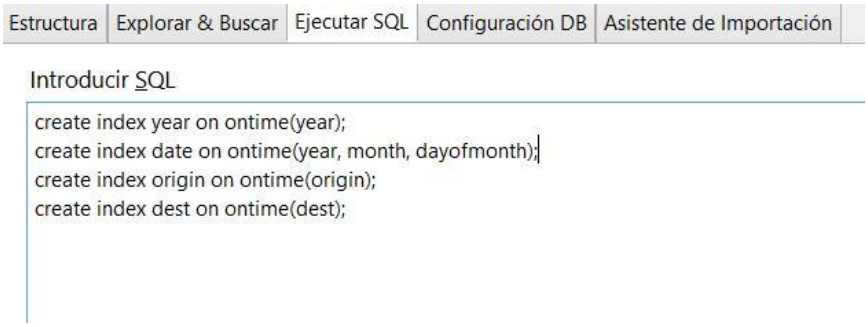


Figura 31, Índices de la BBDD

Para su inclusión en el programa RStudio, primero se han instalado los directorios necesarios mediante el comando install.packages("RSQLite")

```
> install.packages("RSQLite")
Installing package into 'C:/Users/Raul/Documents/R/win-library/3.1'
(as 'lib' is unspecified)
also installing the dependency 'DBI'

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/DBI_0.3.1.zip'
Content type 'application/zip' length 153313 bytes (149 KB)
opened URL
downloaded 149 KB

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/RSQLite_1.0.0.zip'
Content type 'application/zip' length 1211280 bytes (1.2 MB)
opened URL
downloaded 1.2 MB

package 'DBI' successfully unpacked and MD5 sums checked
package 'RSQLite' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
 C:\Users\Raul\AppData\Local\Temp\RtmpEX6a2G\downloaded_packages
>
```

Figura 32, Instalación de paquetes

A continuación, se ha creado el dataframe con la siguiente función, indicando el directorio donde se encuentra el archivo .sqlite:

```
Source on Save
1  library(RSQLite)
2  library(DBI)
3
4  sqLiteConnect <- function(database, table){
5    con <- dbConnect(RSQLite::SQLite(), dbname =database)
6    result <- dbGetQuery(con, paste("select * from ontime"));
7    dbDisconnect(con)
8    return(result)
9  }
10
11 result <-sqLiteConnect("C:/Users/Raul/Desktop/TFG/ontime.sqlite","ontime")
12
```

Figura 33, Creación del dataframe

```
Data
result                    17339 obs. of 29 variables
  Year : int 2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
  Month : int 1 1 1 1 1 1 1 1 1 1 ...
  DayofMonth : int 1 2 3 5 6 7 8 9 10 12 ...
  DayofWeek : int 3 4 5 7 1 2 3 4 5 7 ...
  DepTime : int 1211 1446 1138 1201 1340 1146 1126 1107 1104 1107 ...
  CRSDepTime : int 1110 1110 1110 1110 1110 1110 1105 1105 1105 1105 ...
  ArrTime : int 1454 1755 1402 1428 1620 1423 1347 1337 1339 1348 ...
  CRSArrTime : int 1335 1335 1335 1335 1335 1335 1330 1330 1330 1330 ...
  UniqueCarrier : chr "AA" "AA" "AA" "AA" ...
  FlightNum : int 125 125 125 125 125 125 125 125 125 125 ...
  TailNum : chr "N3CVAA" "N3GWAA" "N3JFAA" "N3FAAA" ...
  ActualElapsedTime: int 283 309 264 267 280 277 261 270 275 281 ...
  CRSElapsedTime : int 265 265 265 265 265 265 265 265 265 265 ...
```

Figura 34, Vista del dataset

Con el propósito de mejorar la información recopilada, se procede a realizar un pre-procesamiento de los datos de la base de datos importada en R, mediante algunas de las acciones descritas a continuación.

Campos vacíos

En primer lugar, se va a proceder a transformar los campos "NA" (vacíos) de las columnas "CarrierDelay", "WeatherDelay", "NASDelay", "SecurityDelay" y "LateAircraftDelay" a 0, indicador de que el retraso equivale a 0 minutos. Para ello se introduce en R la siguiente función:

```
bad<-is.na(result[,"CarrierDelay"])
result[bad,"CarrierDelay"]<-0
```

Y la repetimos para cada uno de los campos anteriores:

```
bad<-is.na(result[,"WeatherDelay"])
result[bad,"WeatherDelay"]<-0

bad<-is.na(result[,"NASDelay"])
result[bad,"NASDelay"]<-0

bad<-is.na(result[,"SecurityDelay"])
result[bad,"SecurityDelay"]<-0

bad<-is.na(result[,"LateAircraftDelay"])
result[bad," LateAircraftDelay "]<-0
```

El resultado obtenido es la sustitución de todas las entradas con NA o datos vacíos por 0.

A continuación, analizamos la columna "CancellationCode", que contiene la mayoría de parámetros NA. Para ver los posibles valores de esta columna, se escribe la siguiente función:

```
table(result[,"CancellationCode"])
```

Devolviendo este res

A	B	C
84	197	4

Analizando este resultado, se procede a sustituir todas las entradas NA de este parámetro por el valor "No code", que significa que no existe código a causa de que el vuelo no ha sido cancelado. Para ello se introduce el siguiente código:

```
Bad<-is.na(result[,"CancellationCode"])
result[bad,"CancellationCode"]<-"No code"
```

A continuación, se procede a eliminar todas las filas que contengan el resto de campos con datos vacíos "NA" para no contaminar el conjunto de datos. Esto se consigue seleccionando las filas correctas e incluyéndolas en un nuevo dataframe llamado "dataset" con el siguiente código:

```
good<-complete.cases(result)
dataset<-result[good,]
```

Como se ha comentado, despues de este proceso se ha conseguido un Nuevo dataframe llamado "data set", pasando de tener 17339 observaciones del dataframe "result" a 17003 observaciones con 29 variables.

3.4 Análisis Exploratorio

El principal propósito del análisis exploratorio es comprender e interpretar los datos y las dependencias entre las diferentes variables en cualquiera de sus permutaciones. En el caso descrito a continuación, las variables dependientes se trata de aquellas que están midiendo los retrasos de vuelo en minutos "CarrierDelay", "WeatherDelay", "NASDelay", "SecurityDelay" y "LateAircraftDelay".

Para todos los tipos de variables en el dataset, utilizaremos el comando descrito a continuación:

```
str(dataset)
```

Cuyo output es el siguiente:

```
$ Year              : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
 $ Month             : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DayofMonth        : int  1 2 3 5 6 7 8 9 10 12 ...
 $ DayofWeek         : int  3 4 5 7 1 2 3 4 5 7 ...
 $ DepTime           : int  1211 1446 1138 1201 1340 1146 1126 1107 1104 1107 ...
 $ CRSDepTime        : int  1110 1110 1110 1110 1110 1110 1105 1105 1105 1105 ...
 $ ArrTime           : int  1454 1755 1402 1428 1620 1423 1347 1337 1339 1348 ...
 $ CRSArrTime        : int  1335 1335 1335 1335 1335 1335 1330 1330 1330 1330 ...
 $ UniqueCarrier     : chr  "AA" "AA" "AA" "AA" ...
 $ FlightNum         : int  125 125 125 125 125 125 125 125 125 125 ...
 $ TailNum           : chr  "N3CVAA" "N3GWAA" "N3JFAA" "N3FAAA" ...
 $ ActualElapsedTime : int  283 309 264 267 280 277 261 270 275 281 ...
 $ CRSElapsedTime    : int  265 265 265 265 265 265 265 265 265 265 ...
 $ AirTime           : int  246 252 244 239 248 258 242 245 252 252 ...
 $ ArrDelay          : int  79 260 27 53 165 48 17 7 9 18 ...
 $ DepDelay          : int  61 216 28 51 150 36 21 2 -1 2 ...
 $ Origin            : chr  "ORD" "ORD" "ORD" "ORD" ...
 $ Dest              : chr  "SEA" "SEA" "SEA" "SEA" ...
 $ Distance          : int  1721 1721 1721 1721 1721 1721 1721 1721 1721 1721 ...
 $ TaxiIn            : int  9 10 7 6 7 7 5 7 6 9 ...
 $ TaxiOut           : int  28 47 13 22 25 12 14 18 17 20 ...
 $ Cancelled         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ CancellationCode  : chr  "No code" "No code" "No code" "No code" ...
 $ Diverted          : chr  "0" "0" "0" "0" ...
 $ CarrierDelay      : num  0 108 0 0 0 0 17 0 0 0 ...
 $ WeatherDelay      : num  9 76 0 37 102 36 0 0 0 0 ...
 $ NASDelay          : num  18 44 0 2 15 12 0 0 0 18 ...
 $ SecurityDelay     : num  0 0 0 0 0 0 0 0 0 0 ...
 $  LateAircraftDelay: num  0 0 0 0 0 0 0 0 0 0 ...
```

Como podemos observar en el output, con este comando permitimos a R mostrarnos por pantalla todas las variables incluidas en dataset, así como su tipo (entero, cadena de caracteres,etc.), y una pequeña muestra de los datos de cada variable.

Además de mostrar el número total de observaciones y el número total de variables del dataframe "dataset"

```
> str(dataset)
'data.frame': 17003 obs. of  29 variables:
 $ Year              : int  2014 2014 2014
 $ Month             : int  1 1 1 1 1 1 1
 $ DayofMonth        : int  1 2 3 5 6 7 8
 $ DayofWeek         : int  3 4 5 7 1 2 3
 $ DepTime           : int  1211 1446 1138
 $ CRSDepTime        : int  1110 1110 1110
 $ ArrTime           : int  1454 1755 1402
 $ CRSArrTime        : int  1335 1335 1335
 $ UniqueCarrier     : chr  "AA" "AA" "AA"
 $ FlightNum         : int  125 125 125 12
 $ TailNum           : chr  "N3CVAA" "N3GW
 $ ActualElapsedTime : int  283 309 264 26
 $ CRSElapsedTime    : int  265 265 265 26
 $ AirTime           : int  246 252 244 23
 $ ArrDelay          : int  79 260 27 53 1
 $ DepDelay          : int  61 216 28 51 1
 $ Origin            : chr  "ORD" "ORD" "O
 $ Dest              : chr  "SEA" "SEA" "S
```

En el caso de nuestro studio, este va a funcionar con una predicción binaria, es decir, simplemente va a predecir si el vuelo está retrasado o no. Esto va a funcionar con los retrasos "LateAircraftDelay".

Para crear una predicción binaria, es necesario crear una nueva columna llamada "IsDelayed", relacionada con los valores de la columna "LateAircraftDelay". Si este valor es menor a 15 minutos, entonces podemos asumir que el vuelo no va con retraso (por ejemplo, "IsDelayed" = 0). Si este valos es mayor al margen, entonces "IsDelayed" será igual a 1. Para ello, introducimos el siguiente código:

```
intall.packages("car")      #Esta línea solo se ejecuta una vez
library("car")
dataset$IsDelayed<-
factor(car::Recode(dataset$LateAircraftDelay,"0:15=0;else=1"),ordered=TRUE)
```

Se procede a comprobar en el dataset el output:

CancellationCode	Diverted	CarrierDelay	WeatherDelay	NASDelay	SecurityDelay	LateAircraftDelay	IsDelayed
No code	0	0	9	18	0	52	1
No code	0	108	76	44	0	32	1
No code	0	0	0	0	0	27	1
No code	0	0	37	2	0	14	0
No code	0	0	102	15	0	48	1
No code	0	0	36	12	0	0	0
No code	0	17	0	0	0	0	0
No code	0	0	0	0	0	0	0
No code	0	0	0	0	0	0	0
No code	0	0	0	18	0	0	0
No code	0	0	0	0	0	0	0
No code	0	0	0	0	0	0	0
No code	0	0	0	0	0	0	0

Figura 35, Output factor

Como podemos observar en la captura, se ha añadido una nueva columna al dataframe dataset llamada IsDelayed, cuyos campos toman un valor binario entre 0 y 1 en función de la columna LateAircraftDelay y de si el vuelo esta efectivamente retrasado o no.

Para obtener una visión general de una variable dependiente, se procede a crear un histograma de la variable "IsDelayed" y su frecuencia con el siguiente comando:

```
Plot(table(dataset$IsDelayed))
```

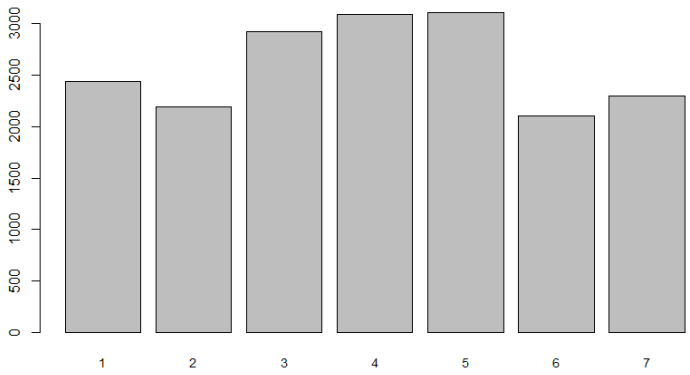
Cuyo resultado es el siguiente:

Figura 36, Gráfico variable IsDelayed

Como puede observarse, se interpretan los dos valores que puede tomar la variable IsDelayed, así como la predominancia de los vuelos retrasados, con casi 16000 de los vuelos en comparación con la ridícula cantidad de vuelos en hora.

El siguiente paso en el análisis exploratorio es el estudio de las relaciones entre la variable dependiente y el conjunto de variables independientes, es decir, en qué medida afectan las variables independientes al retraso del vuelo. Esto se conoce como colinaridad.

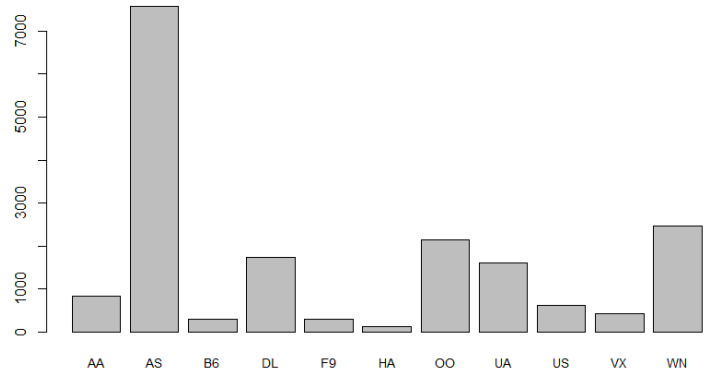
```
barplot(by(data=as.numeric(dataset$IsDelayed),
INDICES=dataset$DayOfWeek, sum ), axes=TRUE, cex.names=0.9)
```



Podemos observar cómo Miércoles, Jueves y Viernes tienen el mayor factor de influencia sobre la variable dependiente, siendo el inicio de fin de semana el valor más alto.

Figura 37, Gráfico día de la semana

```
barplot(by(data=as.numeric(dataset$IsDelayed),
INDICES=dataset$UniqueCarrier, sum ), axes=TRUE, cex.names=0.9)
```



Podemos observar como La compañía AS Alaska Airlines tiene el mayor factor de influencia sobre la variable dependiente, este valor es increíblemente alto.

Figura 38, Gráfico UniqueCarrier

```
barplot(by(data=as.numeric(dataset$IsDelayed),
INDICES=dataset$DepTime, sum ), axes=TRUE, cex.names=0.9)
```

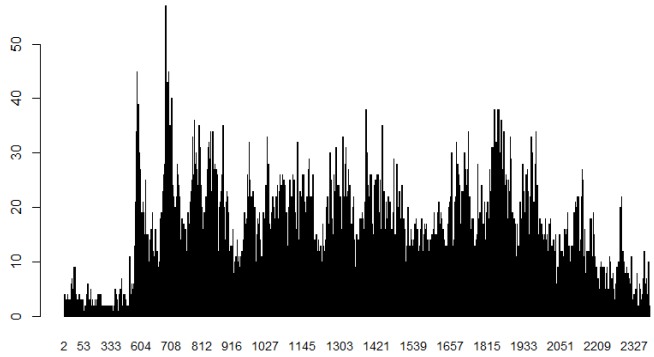


Figura 39, Gráfico DepartureTime

Podemos observar como el mayor número de retrasos, como la variable independiente DepTime afecta al retraso del vuelo, se suceden aproximadamente entre las 6 y las 7 de la mañana, momento de salida crítico.

3.5 Pre-Procesado de los datos

Data Transformation

En estadística, data transformation se refiere a la aplicación de una función matemática determinista a cada punto de un conjunto de datos, es decir, cada punto de datos z i se reemplaza con el valor transformado y i = f (z i), donde f es una función . Las transformaciones se aplican generalmente de forma que los datos parecen cumplir más de cerca los supuestos del procedimiento de inferencia estadística que se va a aplicar, o para mejorar la interpretabilidad o la apariencia de los gráficos.

Casi siempre, la función que se usa para transformar los datos es invertible, y generalmente es continua. La transformación se aplica por lo general a una colección de mediciones comparables. Por ejemplo, si estamos trabajando con los datos sobre los ingresos de la gente en alguna moneda unidad, sería común para transformar el valor de los ingresos de cada persona por el logaritmo función.

Centrado y Escalado (Centering and Scaling)

La transformación de datos más sencilla y común es centrar y escalar las variables predictor. Para centrar una variable predictor, el valor promedio predictor es restado de todos los valores. Como resultado de centrado, el predictor tiene una media cero. Del mismo modo, para escalar los datos, cada valor de la variable predictor es dividida por su desviación estándar. Escalar los datos coacciona los valores para tener una desviación estándar común de uno.

Estas manipulaciones son usadas generalmente para mejorar la estabilidad numérica de algunos cálculos. Algunos modelos, como PLS, se benefician que los predictors se encuentren en una escala común. El único inconveniente de estas transformaciones es una pérdida de la interpretación de los valores individuales ya que los datos ya no están en las unidades originales.

Cuando los predictors se encuentran en diferentes rangos, el centrado y escalado resulta útil al proporcionar más estabilidad y precisión de los algoritmos predictivos. Además, la transformación oblicua resulta muy útil si los datos se encuentran partidos.

Transformaciones para resolver la Skewness (Asimetría)

Otra razón común para las transformaciones es eliminar la asimetría distributiva. Una distribución sesgada es una que es más o menos simétrica. Esto significa que la probabilidad de caer a cada lado de la media de la distribución es aproximadamente igual.

Una distribución sesgada a derechas tiene un gran número de puntos en el lado izquierdo de la distribución (valores más pequeños) que en el lado derecho (valores más grandes). Por ejemplo, los datos de segmentación de células contienen un predictor que mide la desviación estándar de la intensidad de los píxeles en los filamentos de actina.

En las unidades naturales, la exposición de los datos de una fuerte asimetría derecha; hay una mayor concentración de puntos de datos en valores relativamente pequeños y pequeño número de grandes valores.

Una regla general a considerar es que los datos sesgados cuya relación entre el valor más alto en el menor valor es mayor que 20 tienen asimetría significativa. Además, la estadística de asimetría se puede utilizar como un diagnóstico. Si la distribución predictor es aproximadamente simétrica, los valores de asimetría estarán cerca de cero.

A medida que la distribución se desestabiliza, la estadística de asimetría se hace más grande. Del mismo modo, como la distribución se desestabiliza más a la izquierda, el valor se convierte en negativo. La fórmula para la estadística muestra asimetría es:

$$\text{skewness} = \frac{\sum(x_i - \overline{x})^3}{(n-1)v^{3/2}}$$
$$\text{where} \quad v = \frac{\sum(x_i - \overline{x})^2}{(n-1)},$$

Donde x es la variable predictora, n es el número de valores, y x es la media de la muestra del predictor.

Para los datos de la figura siguiente, el panel de la derecha muestra la distribución de los datos una vez que se ha aplicado una transformación logarítmica. Después de la transformación, la distribución no es del todo simétrica pero estos datos están mejor atendidos que cuando estaban en las unidades naturales.

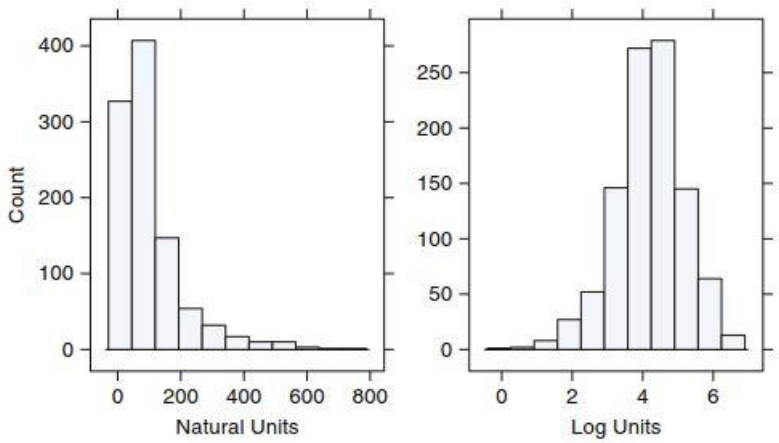


Figura 40, Tranformación logarítmica

En nuestro modelo, necesitamos escalar y centrar las siguientes variables:

- DepTime
- ArrTime

- CRSDepTime
- CRSArrTime
- ActualElapsedTime
- CRSElapsedTime
- AirTime
- Distance
- TaxiIn
- TaxiOut

Para ello, en primer lugar se procede a crear un nuevo dataset llamado processed_dataset, con la siguiente entrada:

```
processed_dataset<-NULL
```

De esta manera, se crea un dataset vacío en el que introducir los datos transformados, que más adelante importaremos a un nuevo data frame.

Con este nuevo dataset creado, procedemos a llenarlo con los predictors mencionados en la tabla anterior, a la vez que convertimos dicho predictor en un valor numérico con las siguientes entradas:

```
#CREATING THE DATASET

processed_dataset<-NULL

processed_dataset$DepTime<-as.numeric(dataset$DepTime)
processed_dataset$CRSDepTime<-as.numeric(dataset$CRSDepTime)
processed_dataset$ArrTime<-as.numeric(dataset$ArrTime)
processed_dataset$CRSArrTime<-as.numeric(dataset$CRSArrTime)
processed_dataset$ActualElapsedTime<-as.numeric(dataset$ActualElapsedTime)
processed_dataset$CRSElapsedTime<-as.numeric(dataset$CRSElapsedTime)
processed_dataset$AirTime<-as.numeric(dataset$AirTime)
processed_dataset$Distance<-as.numeric(dataset$Distance)
processed_dataset$TaxiIn<-as.numeric(dataset$TaxiIn)
processed_dataset$TaxiOut<-as.numeric(dataset$TaxiOut)
```

A continuación, se procede a instalar y cargar la libreria caret, así como a aplicar la función preprocess con la finalidad de centrar y escalar los predictors y aplicar la trasformación de oblicuidad (Box-Cox transformation).

```
# DATA TRANSFORMATION

processed_dataset<-as.data.frame(processed_dataset)
trans=preProcess(processed_dataset,c("BoxCox","center","scale"))
processed_dataset=data.frame(trans=predict(trans,processed_dataset))
```

La función anterior, prueba los diferentes métodos de transformación de datos para conseguir un output lo más limpio posible, proporcionando la siguiente salida:

	trans.DepTime	trans.CRSDepTime	trans.ArrTime	trans.CRSArrTime	trans.ActualElapsedTime	trans.CRSElapsedTime	trans.AirTime
1	-0.2356544101	-0.436201637	-0.151853998	-0.409943952	1.215104238	1.001348431	1.115449761
2	0.2216629674	-0.436201637	0.433429820	-0.409943952	1.421409703	1.001348431	1.171295742
3	-0.3777147018	-0.436201637	-0.250659096	-0.409943952	1.051979192	1.001348431	1.096592272
4	-0.2551147240	-0.436201637	-0.201346500	-0.409943952	1.078501479	1.001348431	1.048901705
5	0.0153836397	-0.436201637	0.168214751	-0.409943952	1.190089464	1.001348431	1.134184997
6	-0.3621464507	-0.436201637	-0.210843744	-0.409943952	1.164805225	1.001348431	1.226087917
7	-0.4010670785	-0.446138313	-0.354369234	-0.419523731	1.025153786	1.001348431	1.077610718
8	-0.4380416750	-0.446138313	-0.373135789	-0.419523731	1.104727422	1.001348431	1.106036410
9	-0.4438797692	-0.446138313	-0.369384716	-0.419523731	1.147796474	1.001348431	1.171295742
10	-0.4380416750	-0.446138313	-0.352491042	-0.419523731	1.198457378	1.001348431	1.171295742
11	-0.4419337378	-0.446138313	-0.363756007	-0.419523731	1.164805225	1.001348431	1.217027178
12	-0.4497178634	-0.446138313	-0.408714820	-0.419523731	0.988902911	1.001348431	1.039267969
13	-0.4458258006	-0.446138313	-0.369384716	-0.419523731	1.156316256	1.001348431	1.226087917
14	-0.4380416750	-0.446138313	-0.391874292	-0.419523731	1.016143420	1.001348431	1.010171147
15	-0.4516638948	-0.446138313	-0.395618611	-0.419523731	1.060853295	1.001348431	1.106036410
16	-0.4380416750	-0.446138313	-0.363756007	-0.419523731	1.147796474	1.001348431	1.198820756
17	-0.4419337378	-0.446138313	-0.376885739	-0.419523731	1.104727422	1.001348431	1.152799747
18	-0.4360956436	-0.446138313	-0.356247147	-0.419523731	1.173263604	1.001348431	1.189674671
19	-0.4283115180	-0.446138313	-0.378760293	-0.419523731	1.034129695	1.001348431	1.077610718
20	-0.4458258006	-0.446138313	-0.427399557	-0.419523731	0.876664918	1.001348431	0.880541669

Figura 41, Dataset Transformado

Se añaden ahora los predictors restantes a processed_dataset:

- Year
- Month
- DayOfMonth
- DayOfWeek
- UniqueCarrier
- FlightNum
- TailNum
- Origin
- Dest
- Cancelled
- CancellationCode
- Diverted

```
# ADDING PREDICTORS

processed_dataset$Year<-as.numeric(as.factor(dataset$Year))
processed_dataset$Month<-as.numeric(as.factor(dataset$Month))
processed_dataset$DayofMonth<-as.numeric(as.factor(dataset$DayofMonth))
processed_dataset$DayofWeek<-as.numeric(as.factor(dataset$DayofWeek))
processed_dataset$UniqueCarrier<-as.numeric(as.factor(dataset$UniqueCarrier))
processed_dataset$FlightNum<-as.numeric(as.factor(dataset$FlightNum))
processed_dataset$TailNum<-as.numeric(as.factor(dataset$TailNum))
processed_dataset$Origin<-as.numeric(as.factor(dataset$Origin))
processed_dataset$Dest<-as.numeric(as.factor(dataset$Dest))
processed_dataset$Cancelled<-as.numeric(as.factor(dataset$Cancelled))
processed_dataset$CancellationCode<-as.numeric(as.factor(dataset$CancellationCode))
processed_dataset$Diverted<-as.numeric(as.factor(dataset$Diverted))
```

Figura 42, Variables predictoras

Análisis de la oblicuidad (skewness)

Con la siguiente función, se procede a analizar la asimetría de cada una de las variables transformadas. Para ello primero es necesario instalar y cargar las librerías:

```
Install.packages("e1071")
Library("e1071")
```

Las medidas de asimetría o *skewness* son indicadores que permiten establecer el grado de simetría (o asimetría) que presenta una distribución de probabilidad de una variable aleatoria sin tener que hacer su representación gráfica.

Como eje de simetría consideramos una recta paralela al eje de ordenadas que pasa por la media de la distribución. Si una distribución es simétrica, existe el mismo número de valores a la derecha que a la izquierda de la media, por tanto, el mismo número de desviaciones con signo positivo que con signo negativo.

Decimos que hay asimetría positiva (o a la derecha) si la "cola" a la derecha de la media es más larga que la de la izquierda, es decir, si hay valores más separados de la media a la derecha. Diremos que hay asimetría negativa (o a la izquierda) si la "cola" a la izquierda de la media es más larga que la de la derecha, es decir, si hay valores más separados de la media a la izquierda.

En teoría de la probabilidad y estadística, la medida de asimetría más utilizada parte del uso del tercer momento estándar. La razón de esto es que nos interesa mantener el signo de las desviaciones con respecto a la media, para obtener si son mayores las que ocurren a la derecha de la media que las de la izquierda. Sin embargo, no es buena idea tomar el momento estándar con respecto a la media de orden 1.

Debido a que una simple suma de todas las desviaciones siempre es cero. En efecto, si por ejemplo, los datos están agrupados en k clases, se tiene que:

$$\sum_{i=1}^{k} f_i(x_i - \mu) = \sum_{i=1}^{k} f_i x_i - \mu \sum_{i=1}^{k} f_i = \mu - \mu = 0$$

en donde x_i representa la marca de la clase i-ésima y f_i denota la frecuencia relativa de dicha clase. Por ello, lo más sencillo es tomar las desviaciones al cubo.

El coeficiente de asimetría de Fisher, representado por \gamma_1, se define como:

$$\gamma_1 = \frac{\mu_3}{\sigma^3},$$

donde \mu_3 es el tercer momento en torno a la media y \sigma es la desviación estándar.

S i$\gamma_1 > 0$, la distribución es asimétrica positiva o a la derecha.

Si $\gamma_1 < 0$, la distribución es asimétrica negativa o a la izquierda.

Si la distribución es simétrica, entonces sabemos que $\gamma_1 = 0$. El recíproco no es cierto: es un error común asegurar que si $\gamma_1 = 0$ entonces la distribución es simétrica (lo cual es falso).

A continuación se pretende comparar la asimetría de la variable ArrTime en dataset y processed_dataset:

```
skewness(dataset$ArrTime)
skewness(processed_dataset$trans.ArrTime)
hist(dataset$Arrtime)
hist(processed_dataset$trans.ArrTime)
```

Para ello, las líneas anteriores de código nos muestran el coeficiente de asimetría de la variable ArrTime antes y después de la trasformación.

```
> skewness(dataset$ArrTime)
[1] -0.4647361
> skewness(processed_dataset$trans.ArrTime)
[1] -0.2566696
```

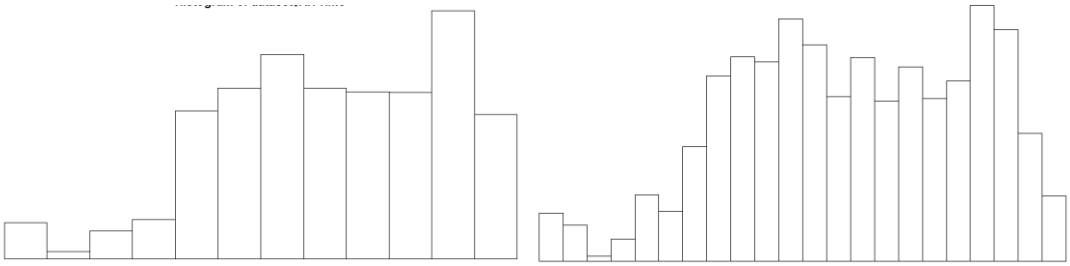
Podemos observar como el coeficiente de asimetría es negativo, además por lo que ha disminuido en la variable transformada, por lo que podemos suponer que la variable ha mejorado su gráfica, asemejándose más a una distribución normal.

Realizamos esto para todas las variables.

ARRTIME

skewness(dataset$ArrTime)
[1] -0.4647361

skewness(processed_dataset$trans.ArrTime)
[1]-0.256669

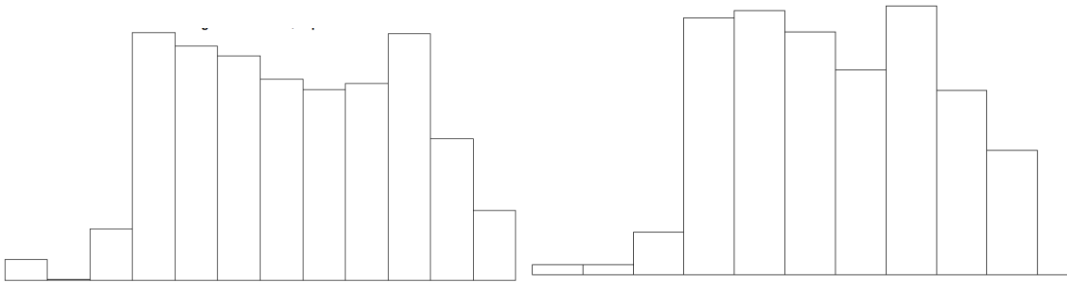


Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

DEPTIME

skewness(dataset$DepTime)
[1]0.01611072

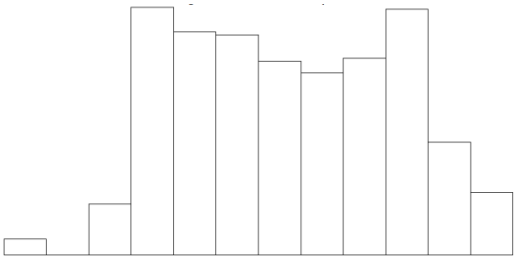
skewness(processed_dataset$trans.DepTime)
[1] 0.01611072



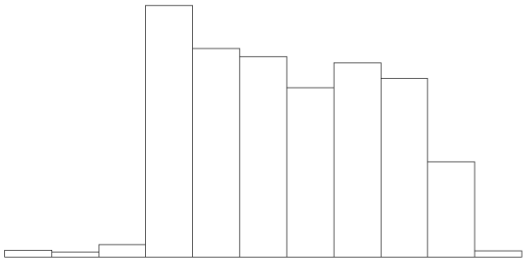
Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es la misma, por lo que podemos intuír en el histograma que no se ha realizado ningun cambio sobre la skewness.

CRSDEPTIME

skewness(dataset$CRSDepTime)
[1]0.06408598



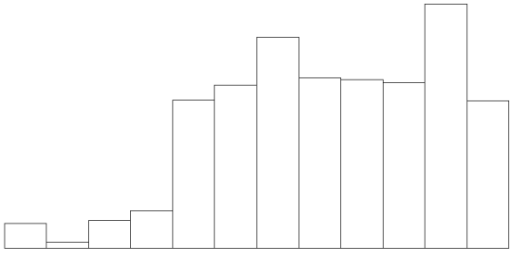
skewness(processed_dataset$trans.CRSDepTime)
[1] 0.06408598



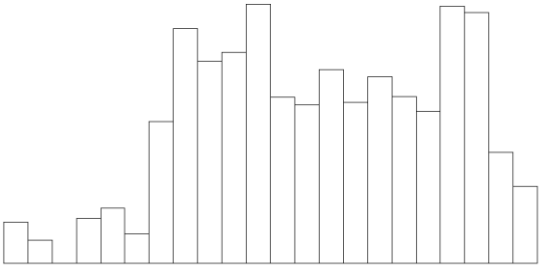
Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es la misma, por lo que podemos intuír en el histograma que no se ha realizado ningun cambio sobre la skewness.

CRSARRTIME

skewness(dataset$CRSArrTime)
[1]-0.4063495



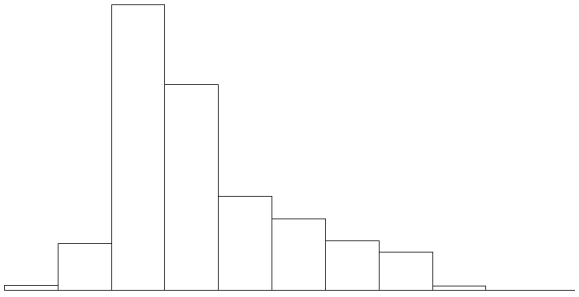
skewness(processed_dataset$trans.CRSArrTime)
[1] -0.215386



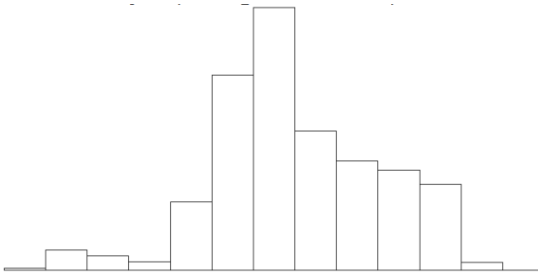
Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

ACTUALELAPSEDTIME

skewness(dataset$ActualElapsedTime)
[1]0.9218718



skewness(processed_dataset$trans.Actual ElapsedTime)
[1] -0.1205357

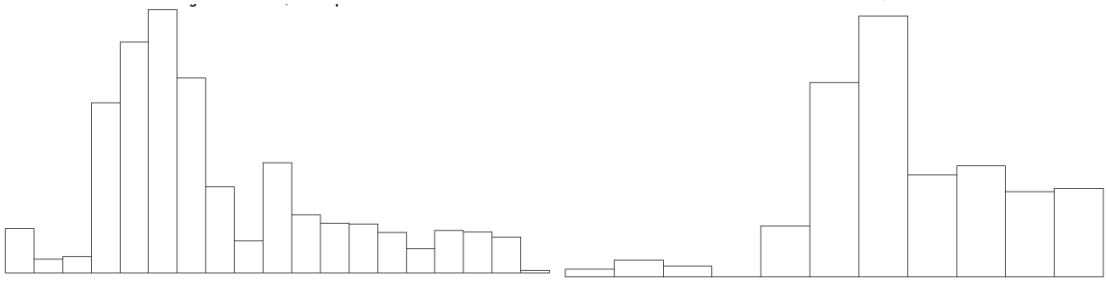


Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

CRSELAPSEDTIME

skewness(dataset$CRSElapsedTime)
[1]0.9079377

skewness(processed_dataset$trans.CRSElapsedTime)[1] -0.2074146

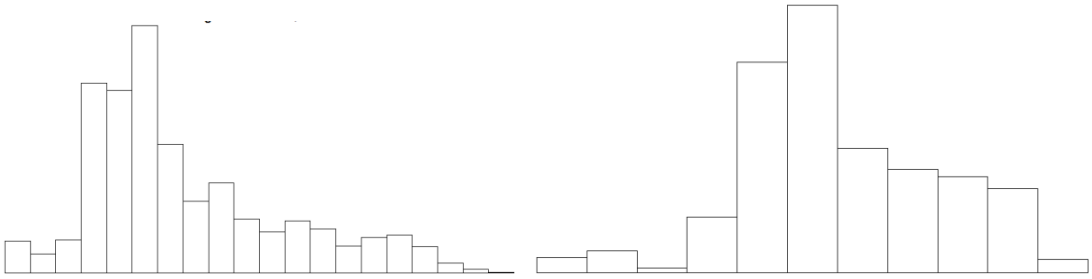


Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

AIRTIME

skewness(dataset$AirTime)
[1]0.9309248

skewness(processed_dataset$trans.AirTime)
[1] 0.02327244

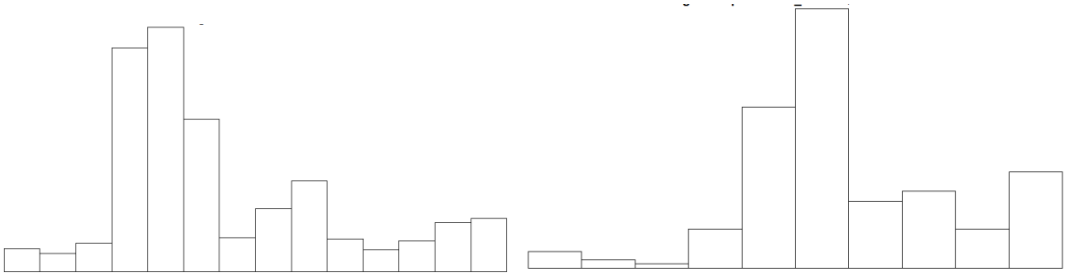


Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

DISTANCE

skewness(dataset$Distance)
[1]0.8604543

skewness(processed_dataset$trans.Distance)
[1] 0.09226696



Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

TAXIIN

skewness(dataset$TaxiIn)
[1]6.278846

skewness(processed_dataset$trans.TaxiIn)
[1] -0.1358623

Podemos observar como la tranformación ha actuado sobre la skewness. En el caso de la variable transformada la asímetria es mas cercana a 0, por lo que podemos intuír en el histograma como sigue una distribución mas semejante a la normal.

TAXIOUT

```
skewness(dataset$TaxiIn)
 [1]6.278846
skewness(processed_dataset$trans.TaxiIn)
 [1] -0.135
```

Feature Selection

En *learnign machine* y en estadística, la selección de características o *feature selection*, también conocido como la variable de selección, selección de atributos o una selección subconjunto variables, es el proceso de selección de un subconjunto de características pertinentes para su uso en la construcción de un modelo.

El supuesto central cuando se utiliza una técnica de selección característica es que los datos contienen muchas características redundantes o irrelevantes.

Características redundantes son los que proporcionan más información que las entidades seleccionadas en la actualidad, y las características irrelevantes proporcionar ninguna información útil en cualquier contexto.

Técnicas de feature selection es de distinguirse de extracción de características. Extracción de características crea nuevas características de las funciones de los elementos originales, mientras que la selección de funciones devuelve un subconjunto de las características.

Técnicas de selección de funciones se utilizan a menudo en ámbitos en los que hay muchas características y comparativamente pocas muestras (o puntos de datos). El caso paradigmático es el uso de la función de selección en el análisis de microarrays de ADN, donde hay muchos miles de funciones, y unas pocas decenas a cientos de muestras. Técnicas de selección de funciones proporcionan tres ventajas principales al construir modelos predictivos:

- mejora de la interpretabilidad modelo,
- tiempos de entrenamiento más cortos,
- generalización mejorada mediante la reducción de sobreajuste.

Característica de selección también es útil como parte del proceso de análisis de datos, ya que muestra qué características son importantes para la predicción, y cómo se relacionan estas características.

En probabilidad y estadística, la correlación indica la fuerza y la dirección de una relación lineal y proporcionalidad entre dos variables estadísticas.

Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra: si tenemos dos variables (A y B) existe correlación si al aumentar los valores de A lo hacen también los de B y viceversa. La correlación entre dos variables no implica, por sí misma, ninguna relación de causalidad

La correlación estadística constituye una técnica estadística que nos indica si dos variables están relacionadas o no.

Por ejemplo, considera que las variables son el ingreso familiar y el gasto familiar. Se sabe que los aumentos de ingresos y gastos disminuyen juntos. Por lo tanto, están relacionados en el sentido de que el cambio en cualquier variable estará acompañado por un cambio en la otra variable.

De la misma manera, los precios y la demanda de un producto son variables relacionadas; cuando los precios aumentan la demanda tenderá a disminuir y viceversa.

Si el cambio en una variable está acompañado de un cambio en la otra, entonces se dice que las variables están correlacionadas. Por lo tanto, podemos decir que el ingreso familiar y gastos familiares y el precio y la demanda están correlacionados.

La correlación puede decir algo acerca de la relación entre las variables. Se utiliza para entender:

- si la relación es positiva o negativa
- la fuerza de la relación.

La correlación es una herramienta poderosa que brinda piezas vitales de información.

En el caso del ingreso familiar y el gasto familiar, es fácil ver que ambos suben o bajan juntos en la misma dirección. Esto se denomina correlación positiva. En caso del precio y la demanda, el cambio se produce en la dirección opuesta, de modo que el aumento de uno está acompañado de un descenso en el otro. Esto se conoce como correlación negativa.

Coeficiente de Correlación

La correlación estadística es medida por lo que se denomina coeficiente de correlación (r). Su valor numérico varía de 1,0 a -1,0. Nos indica la fuerza de la relación.

En general, r> 0 indica una relación positiva y r <0 indica una relación negativa, mientras que r = 0 indica que no hay relación (o que las variables son independientes y no están relacionadas). Aquí, r = 1,0 describe una correlación positiva perfecta y r = -1,0 describe una correlación negativa perfecta.

Cuanto más cerca estén los coeficientes de +1,0 y -1,0, mayor será la fuerza de la relación entre las variables. Como norma general, las siguientes directrices sobre la fuerza de la relación son útiles (aunque muchos expertos podrían disentir con la elección de los límites).

Valor de r	Fuerza de relación
-1,0 A -0,5 o 1,0 a 0,5	Fuerte
-0,5 A -0,3 o 0,3 a 0,5	Moderada
-0,3 A -0,1 o 0,1 a 0,3	Débil
-0,1 A 0,1	Ninguna o muy débil

La correlación es solamente apropiada para examinar la relación entre datos cuantificables significativos (por ejemplo, la presión atmosférica o la temperatura) en vez de datos categóricos, tales como el sexo, el color favorito, etc.

Correlation Feature Selection

La selección de características en función de la correlación evalúa las características en función de su correlación. Es decir, predictors con alta correlación con otros predictors deben ser eliminados.

Procedimiento:
En primer lugar, es necesario filtrar los predictors con variáncia igual a cero. Para ello, se añade el paquete caret con la función NearZeroVar, que devolvera aquellas columnas de predictors cuya variáncia sea nula.

NearZeroVar(processed_dataset)

```
# ******* FEATURE SELECTION ********

nearZeroVar(processed_dataset)
```

Output:
```
> nearZeroVar(processed_dataset)
[1] 11 12 20 21 22
```

Esto nos indica, que las columnas del dataframe processed_dataset 11,12,20,21,22, correspondiendo a las variables Year, Month, Cancelled, CancellationCode, Diverted, tienen variáncia igual a cero.

A continuación, es necesario eliminar dichas variables del dataframe con el siguiente código:

```
# DELETING THE 0VAR VARIABLES

processed_dataset$Year<-NULL
processed_dataset$Month<-NULL
processed_dataset$Cancelled<-NULL
processed_dataset$CancellationCode<-NULL
processed_dataset$Diverted<-NULL
```

Con las variables anteriores eliminadas, con la ayuda de la función cor se calculan las correlaciones entre los predictors.

```
# CORRELATION ANALYSIS

correlations<-cor(processed_dataset)
```

	row.names	trans.DepTime	trans.CRSDepTime	trans.ArrTime	trans.CRSArrTime	trans.ActualElapsedTime	trans.CRSElapsedTime	trans.AirTime
1	trans.DepTime	1.000000000	0.9622763719	0.549992773	5.919868e-01	0.03501816	0.046758028	0.05997040
2	trans.CRSDepTime	0.962276372	1.0000000000	0.527740557	5.760949e-01	0.03941504	0.050906908	0.06493753
3	trans.ArrTime	0.549992773	0.5277405571	1.000000000	8.959812e-01	0.05501808	0.062074150	0.07380619
4	trans.CRSArrTime	0.591986781	0.5760949126	0.895981225	1.000000e+00	0.06707028	0.075742674	0.08714294
5	trans.ActualElapsedTime	0.035018157	0.0394150446	0.055018084	6.707028e-02	1.00000000	0.984005333	0.99158340
6	trans.CRSElapsedTime	0.046758028	0.0509069078	0.062074150	7.574267e-02	0.98400533	1.000000000	0.98745862
7	trans.AirTime	0.059970399	0.0649375270	0.073806186	8.714294e-02	0.99158340	0.987458618	1.00000000
8	trans.Distance	0.037590644	0.0442656864	0.059232620	7.254082e-02	0.97076889	0.983622529	0.97713842
9	trans.TaxiIn	-0.082679969	-0.0859951839	0.008908125	5.570967e-05	0.20525455	0.174065593	0.15267443
10	trans.TaxiOut	-0.146500173	-0.1485602039	-0.125787276	-1.312465e-01	0.27451315	0.197282466	0.17872976
11	DayofMonth	-0.022077792	-0.0198341139	0.014895993	7.083492e-03	-0.01291762	-0.009786284	-0.01012672
12	DayofWeek	0.017219758	0.0180775715	0.002294843	4.549915e-03	0.02496709	0.021232127	0.02015334
13	UniqueCarrier	-0.001536977	-0.0221626163	-0.003557472	-1.908032e-02	-0.25009413	-0.241355739	-0.24051660
14	FlightNum	0.029555099	0.0245295377	-0.003251848	-1.314128e-02	-0.22637228	-0.236590555	-0.23724122
15	TailNum	-0.001247401	-0.0010313165	-0.023360184	-2.526204e-02	-0.08299142	-0.082710052	-0.08668415
16	Origin	-0.025324524	-0.0174607784	-0.015613645	-2.058507e-02	-0.23446782	-0.224721766	-0.24236580
17	Dest	0.014190285	-0.0006749746	0.066649815	7.391012e-02	-0.11111041	-0.120952423	-0.10778608

En el output anterior podemos observar en la tabla la correlación cruzada entre todos los predictors.

Con el fin de examinar visualmente la estructura de dichos datos, es necesario instalar el paquete corrplot. El tamaño y el color de los puntos están directamente asociado con la fuerza de la correlación entre dos variables predictor.

```
Install.packages("corrplot")
Library("corrplot")
Corrplot(correlations, order = "hclust")
```

Output:

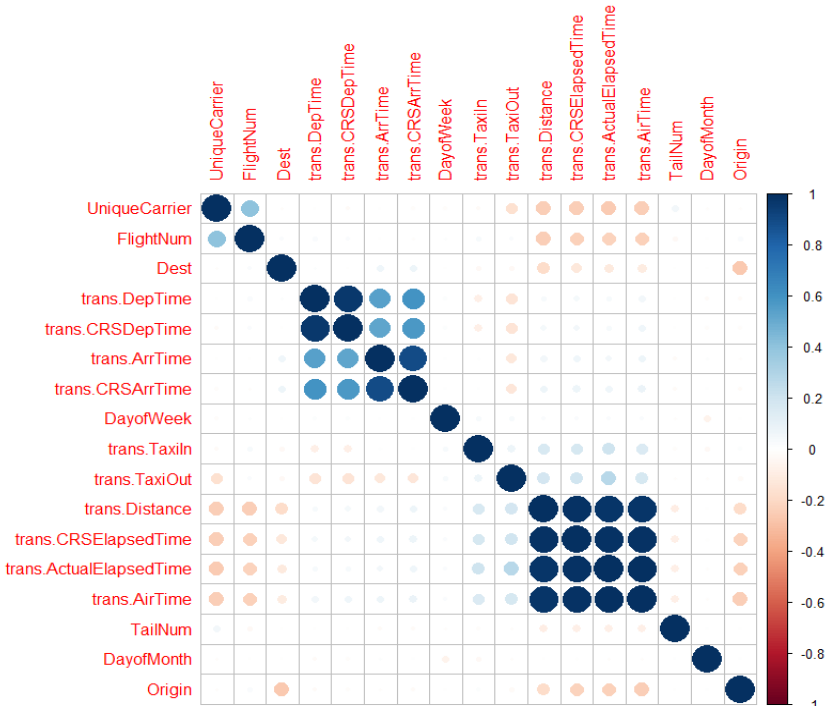


Figura 43, Gráfico de la correlación

Se puede deducir del siguiente gráfico:
Correlación Fuerte: DepTime y CRSDepTime, ArrTime y CRSArrTime, Distance – CRSElapsedTime – ActualElapsedTime – AirTime.

Correlación Moderada: DepTime – ArrTime – CRSArrTime

Para filtrar las variables en función de su correlación, la función findCorrelation aplicará un algoritmo de selección.

```
# FILTERING

highCorr<-findCorrelation(correlations, cutoff = 0.75)
head(highCorr)
filtered_processed_dataset<-processed_dataset[,-highCorr]
```

Este algoritmo establece un límite en el coeficiente de correlación entre todas las permutaciones de variables, en nuestro caso 0.75, eliminando aquellas variables cuyo coeficiente se encuentre por encima de dicho umbral.

Además, crea un nuevo dataframe filtrado donde alojará las variables saneadas llamado filtered_processed_dataset.

Por último se añade la variable IsDelayed al nuevo dataframe:

	trans.CRSDepTime	trans.ArrTime	trans.Distance	trans.Taxiln	trans.TaxiOut	DayofMonth	DayofWeek	UniqueCarrier	FlightNum	TailNum	Origin	Dest	IsDelayed
1	-0.436201637	-0.151853998	0.86520631	0.9750681	1.57019260	1	3	1	69	558	45	55	1
2	-0.436201637	0.433429820	0.86520631	1.1693068	2.57327798	2	4	1	69	628	45	55	1
3	-0.436201637	-0.250659096	0.86520631	0.4773236	-0.23382323	3	5	1	69	649	45	55	1
4	-0.436201637	-0.201346500	0.86520631	0.1463155	1.04722592	5	7	1	69	599	45	55	0
5	-0.436201637	0.168214751	0.86520631	0.4773236	1.32914462	6	1	1	69	648	45	55	1
6	-0.436201637	-0.210843744	0.86520631	0.4773236	-0.44706434	7	2	1	69	527	45	55	0
7	-0.446138313	-0.354369234	0.86520631	-0.2724629	-0.04090618	8	3	1	69	524	45	55	0
8	-0.446138313	-0.373135789	0.86520631	0.4773236	0.58227973	9	4	1	69	559	45	55	0
9	-0.446138313	-0.369384716	0.86520631	0.1463155	0.44464289	10	5	1	69	541	45	55	0
10	-0.446138313	-0.352491042	0.86520631	0.9750681	0.82988394	12	7	1	69	616	45	55	0
11	-0.446138313	-0.363756007	0.86520631	0.7480133	-0.44706434	13	1	1	69	524	45	55	0
12	-0.446138313	-0.408714820	0.86520631	0.9750681	-0.95233305	14	2	1	69	660	45	55	0
13	-0.446138313	-0.369384716	0.86520631	0.7480133	-0.95233305	15	3	1	69	605	45	55	0
14	-0.446138313	-0.391874292	0.86520631	0.7480133	0.44464289	16	4	1	69	515	45	55	0
15	-0.446138313	-0.395618611	0.86520631	0.1463155	-0.04090618	17	5	1	69	613	45	55	0
16	-0.446138313	-0.363756007	0.86520631	0.1463155	-0.04090618	19	7	1	69	563	45	55	0
17	-0.446138313	-0.376885739	0.86520631	0.9750681	-0.68475369	20	1	1	69	515	45	55	0
18	-0.446138313	-0.356247147	0.86520631	0.4773236	0.44464289	21	2	1	69	663	45	55	0
19	-0.446138313	-0.378760293	0.86520631	0.7480133	-0.44706434	22	3	1	69	537	45	55	0
20	-0.446138313	-0.427399557	0.86520631	0.1463155	0.44464289	23	4	1	69	570	45	55	0
21	-0.446138313	-0.390001708	0.86520631	0.4773236	-0.68475369	24	5	1	69	529	45	55	0
22	-0.446138313	-0.361879211	0.86520631	-0.8283631	0.13488072	26	7	1	69	631	45	55	0
23	-0.446138313	-0.360002136	0.86520631	1.3381025	-0.68475369	27	1	1	69	630	45	55	0
24	-0.446138313	-0.369384716	0.86520631	0.4773236	-1.25716959	28	2	1	69	526	45	55	0
25	-0.446138313	-0.243084358	0.86520631	0.9750681	-1.25716959	29	3	1	69	675	45	55	0
26	-0.446138313	-0.348733822	0.86520631	0.4773236	-0.68475369	30	4	1	69	663	45	55	0
27	-0.446138313	-0.378760293	0.86520631	0.4773236	0.13488072	31	5	1	69	537	45	55	0
28	0.189808907	1.223969183	0.86520631	2.1147586	-0.68475369	1	3	1	69	558	55	45	1
29	0.189808907	-2.362708563	0.86520631	1.4866815	-0.04090618	2	4	1	69	628	55	45	1

Figura 44, Vista Dataset Isdelayed

3.6 Regresión Logística

La importancia que tiene para la ciencia la habilidad de hacer predicciones tiene que ver con su capacidad de explicar realmente los procesos naturales, y no solo describirlos. No basta con observar y describir que un gas determinado aumenta su volumen al aumentar la temperatura (a presión constante),

sino que la ciencia debe ser capaz de formular una ley general que se aplique en la mayoría los casos, y a partir de esta ley predecir qué ocurrirá con cualquier situación en determinadas circunstancias, para lograr no solo una explicación satisfactoria del proceso estudiado, sino también sus posibles aplicaciones técnicas.

Por ejemplo, nadie se subiría a un avión si la ciencia no fuera capaz de predecir que cierta velocidad y cierta forma y superficie del ala son suficientes para mantener al avión en el aire. Uno tiene que saber cómo se va a comportar un sistema en determinadas circunstancias, basado en leyes comprobables experimentalmente.

De esta manera, a partir de los datos procesados en los pasos anteriores, es necesario realizar una predicción científica sobre que vuelo es más probable que sufra un retraso, o qué compañía, o que aeropuertos de origen y destino.

Para ello, en primer lugar, se procede a la creación de dos sets que actuarán como base para nuestro análisis. Por un lado el set training_data y por otro el set crossvalidation_data. Este último set es necesario para la comprobación de la veracidad de los modelos de predicción, sirviendo como referencia para aceptar o rechazar dichas hipótesis.

Este paso lo realizaremos con la función createDataPartition de la librería caret a partir de la variable IsDelayed de filtered_processed_dataset con el siguiente código:

```
inTrain<-caret::createDataPartition
(filtered_processed_dataset$IsDelayed,p=.85, list=FALSE)

training_data<-filtered_processed_dataset[inTrain,]

crossvalidation_data<-filtered_processed_dataset[-inTrain,]
```

Este código nos ofrece como resultado los training y crossvalidation como output de la siguiente forma:

correlations	num [1:17, 1:17] 1 0.962 0.55 0.592 0.035 ...
crossvalidation_data	2549 obs. of 13 variables
dataset	17003 obs. of 30 variables
filtered_processed_dataset	17003 obs. of 13 variables
inTrain	int [1:14454, 1] 4 6 7 8 9 10 11 12 13 14 ...
processed_dataset	17003 obs. of 17 variables
result	17339 obs. of 29 variables
training_data	14454 obs. of 13 variables

Figura 45, Training & crossvalidation

APLICACIÓN

Para nuestro caso, se pretende usar un modelo de regresión logística en el set training_data y posteriormente se comparará y validará su calidad y precisión con el set crossvalidation_data.

En primer lugar, se usa la función glm de la librería caret.

Los Modelos Lineales Generalizados (GLM) son tan fáciles de encajar en eI modelo de regresión lineal como en los ordinarios. De hecho, sólo requieren un parámetro adicional para especificar las funciones de varianza y de enlace.

La herramienta básica para los modelos lineales generalizados de ajuste es la función glm, incluída en la librería caret, que tiene la siguiente estructura:

glm(fórmula, familia, datos, pesos ,subconjunto , ...)

El único parámetro que no hemos visto antes es la familia, que es una forma sencilla de especificar una selección de funciones de varianza y de enlace. Hay ocho opciones de familias:

Family	Default Link Function
binomial	(link = "logit")
gaussian	(link = "identity")
Gamma	(link = "inverse")
inverse.gaussian	(link = "1/mu^2")
poisson	(link = "log")
quasi	(link = "identity", variance = "constant")
quasibinomial	(link = "logit")
quasipoisson	(link = "log")

Se introduce en el modelo la siguiente función glm:

```
g = glm(IsDelayed~.,data = training_data, family = binomial("logit"))
```

Glm ha asignado su valor a una llamada IsDelayed objeto (por ajuste de regresión logística). El primer argumento de la función es una fórmula modelo, que define la respuesta y predictor lineal.

Con los datos binomiales la respuesta puede ser un vector o una matriz con dos columnas. Si la respuesta es un vector que puede ser numérico con 0 para el fracaso y 1 para el éxito, o un factor con el primer nivel de representación de "fracaso" y todos los otros que representan el "éxito". En estos casos R genera un vector de unos para representar los denominadores binomiales.

Alternativamente, la respuesta puede ser una matriz donde la primera columna es el número de "éxitos" y la segunda columna es el número de "fracasos". En este caso R añade las dos columnas juntas para producir el denominador binomio correcto.

Dichos datos se almacenan en el valor g.

Podemos mostrar los efectos de esta función en pantalla con el comando print.

```
Print(g)
```

```
Call:  glm(formula = IsDelayed ~ ., family = binomial("logit"), data = training_data)

Coefficients:
     (Intercept)  trans.CRSDepTime     trans.ArrTime    trans.Distance
      -3.078e+00         5.868e-01         1.323e-01        -8.096e-02
    trans.TaxiIn     trans.TaxiOut       DayofMonth        DayofWeek
       1.136e-01         9.137e-02        -5.636e-02         3.230e-02
   UniqueCarrier         FlightNum            TailNum           Origin
       2.174e-01         3.076e-04         3.035e-05        -1.132e-02
            Dest
      -1.130e-03

Degrees of Freedom: 14453 Total (i.e. Null);  14441 Residual
Null Deviance:     7172
Residual Deviance: 6106  AIC: 6132
>
```

Figura 46, Vista valor g

A continuación se introduce la función predict para crear una predicción y compararla mediante el set crossvalidation_data y se almacenan en el valor *p*. Esta función nos retorna un valor numérico correspondiente a la certeza de la predicción.

```
         4          5          6         18         20         44
0.024383757 0.021340230 0.016860560 0.008208602 0.007448353 0.011864216
        52         67         69         72         73         77
0.055883463 0.045544468 0.027872748 0.019817321 0.018176540 0.013078844
        89        101        114        116        124        141
0.006424438 0.025390239 0.009570209 0.009027599 0.005899648 0.099272087
       166        169        176        186        191        196
0.023410181 0.055750189 0.041718281 0.028536469 0.017789851 0.014520932
       197        199        203        204        209        211
0.015021753 0.016791343 0.012675586 0.012142112 0.010397285 0.007498115
       214        226        231        240        245        246
0.005402274 0.003408742 0.033571925 0.013336772 0.009182972 0.011078525
```

Figura 47, Resultado predict

AUC & ROC

En la Teoría de detección de señales una curva ROC (acrónimo de Receiver Operating Characteristic, o Característica Operativa del Receptor) es una representación gráfica de la sensibilidad frente a (1 – especificidad) para un sistema clasificador binario según se varía el umbral de discriminación.

Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o ratio de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo).

ROC también puede significar Relative Operating Characteristic (Característica Operativa Relativa) porque es una comparación de dos características operativas (VPR y FPR).

El análisis de la curva ROC, o simplemente análisis ROC, proporciona herramientas para seleccionar los modelos posiblemente óptimos y descartar modelos subóptimos independientemente de (y antes de especificar) el coste de la distribución de las dos clases sobre las que se decide. La curva ROC es también independiente de la distribución de las clases en la población (en diagnóstico, la prevalencia de una enfermedad en la población).

3.7 Resultados Regresión Logística

El análisis ROC se relaciona de forma directa y natural con el análisis de coste/beneficio en toma de decisiones. Para ello se instala la librería ROCR y se introduce el siguiente código:

```
#ROC

pred <- prediction(p, crossvalidation_data$IsDelayed)
perf <- performance(pred, 'tpr', 'fpr')
plot(perf, main="ROC curve", xlab="False positive rate", ylab="True positive rate")
```

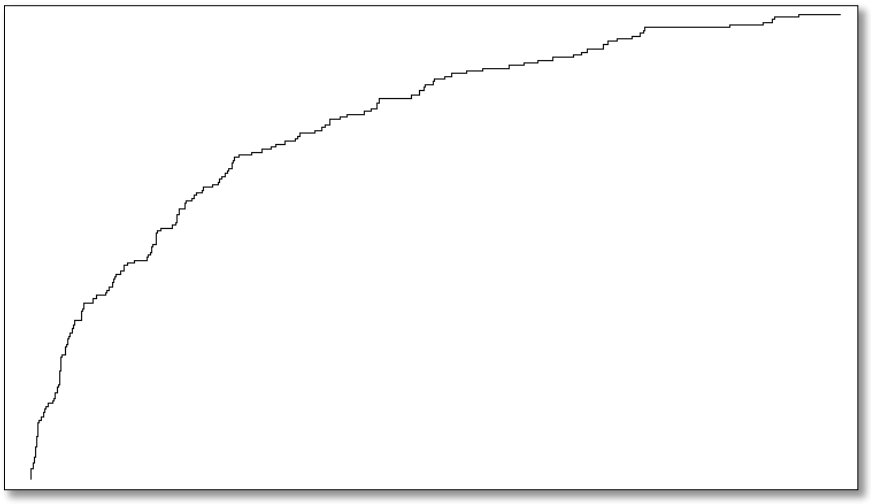


Figura 48, Curva ROC, regresión logística

Si la función encuentra un VP (Verdadero positivo), la gráfica asciende. De lo contrario si la función encuentra un FN (Falso negativo) la gráfica se mantiene constante en horizontalidad. Cuanto más alta y pronunciada es la curva más fiable es la predicción.

Con el propósito de adquirir conclusiones sobre la calidad del modelo almacenado en la variable g, es necesario juzgar el modelo de predicción en el área bajo la curva ROC y calvular el valor AUC (Area under curve).

Para la elección entre dos pruebas diagnósticas distintas, se recurre a las curvas ROC, ya que es una medida global e independiente del punto de corte. Por esto, en el ámbito sanitario, las curvas ROC también se denominan curvas de rendimiento diagnóstico.

La elección se realiza mediante la comparación del área bajo la curva (AUC) de ambas pruebas. Esta área posee un valor comprendido entre 0,5 y 1, donde 1 representa un valor diagnóstico perfecto y 0,5 es una prueba sin capacidad discriminatoria diagnóstica. Es decir, si AUC para una prueba diagnóstica es 0,8 significa que existe un 80% de probabilidad de que el diagnóstico realizado a un enfermo sea más correcto que el de una persona sana escogida al azar. Por esto, siempre se elige la prueba diagnóstica que presente una mayor área bajo la curva.

A modo de guía para interpretar las curvas ROC se han establecido los siguientes intervalos para los valores de AUC:

- [0.5, 0.6): Test malo.
- [0.6, 0.75): Test regular.
- [0.75, 0.9): Test bueno.
- [0.9, 0.97): Test muy bueno.
- [0.97, 1): Test excelente.

```
# AUC
auc_rdock <- performance(pred, "auc")
auc.area_rdock <- slot(auc_rdock, "y.values")[[1]]
cat("AUC: \n")
cat(auc.area_rdock)
cat("\n\n")
```

Output:

```
AUC:
0.772478
```

Podemos observar a partir del valor del AUC, como situaríamos el test en una valoración de buena o aceptable para la prueba efectuada.

3.8 Gradient Boosting Machine

Los modelos de impulso o *boosting*[27] fueron desarrollados originalmente para los problemas de clasificación y se extendieron posteriormente a la configuración de regresión.

Esta historia comienza con el algoritmo AdaBoost[23] y evoluciona a la *gradient boosting machine* estocástica de Friedman, que ahora es ampliamente aceptado como el algoritmo de boosting de mayor elección, en cuyo caso aplicaremos en esta sección.

A principios de 1990 aparecieron los primeros algoritmos de *boosting* (Schapire 1990; Freund 1995; Schapire 1999), que fueron influenciados por la teoría del aprendizaje (Valiant 1984; Kearns y Valiant 1989), en el que una serie de débiles clasificadores (un clasificador que predice un poco mejor que al azar) se combinan (o impulsan) para producir un clasificador conjunto con una tasa de error generalizada superior.

Los investigadores se esforzaron por un tiempo para hallar una implementación reflexiva de impulsar la teoría, hasta que Freund y Schapire colaboraron para producir el algoritmo AdaBoost (Schapire 1999). AdaBoost proporciona una implementación práctica del concepto de boosting.

Impulso o boosting, especialmente en la forma del algoritmo AdaBoost, ha demostrado ser una herramienta de predicción potente, por lo general superando cualquier modelo individual. Su éxito llamó la atención de la comunidad de modelado y su uso se generalizó con aplicaciones en la expresión génica (Dudoit et al 2002;.. Ben-Dor et al 2000), quimiometría (. Varmuza et al 2003), y el género de la música (Bergstra et al., 2006).

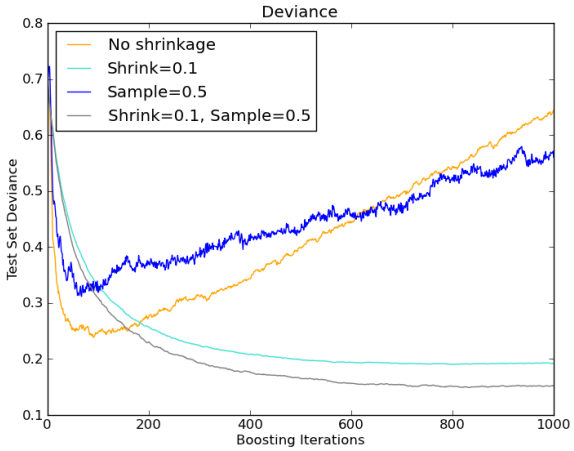


Figura 49, Gradient Boosting Machine

El algoritmo AdaBoost claramente funcionó, y después de su llegada con éxito, varios investigadores (Friedman et al. 2000) conecta el algoritmo AdaBoost a los conceptos estadísticos de funciones de pérdida, modelado aditivo y de regresión logística y mostró que el aumento se puede interpretar como un aditivo por etapas hacia adelante en un modelo modelo que minimiza la pérdida exponencial. Esta comprensión fundamental de boosting llevado a una nueva visión de impulso que facilitó a varias generalizaciones algorítmicas los problemas de clasificación. Por otra parte, esta nueva perspectiva también permitió el método se extienda a problemas de regresión.

La capacidad de Friedman para ver el marco estadístico de boosting arrojó un algoritmo simple, elegante y altamente adaptable para diferentes clases de problemas (Friedman 2001). Él llama a este método "gradient boosting machine", que abarcaba tanto la clasificación como la regresión.

Los principios básicos de gradient boosting son los siguientes: dado una función de pérdida (por ejemplo, error cuadrático para la regresión) y un principiante débil (por ejemplo, árboles de regresión), el algoritmo busca para hallar un modelo aditivo que minimiza la función de pérdida. El algoritmo se inicializa típicamente con la mejor estimación de la respuesta (por ejemplo, la media de la respuesta en la regresión).

1 Selecciona la profundidad del árbol, D, y el número de iteraciones, K
2 Computa la average response, y, y se usa como valor inicial predictor para cada muestra
3 **For** k = 1 to K **do**
4 Computa el residual, la diferencia entre el valor observado y el valor actual predicho, para cada muestra
5 Encaja un árbol de regresión de profundidad, D, usando los residuales como response
6 Predice cada muestra usando el árbol de regresión del paso anterior
7 Actualiza el valor de la predicción de cada muestra añadiendo el valor de la iteración anterior al valor de la predicción generado en el paso anterior
8 **end**

Figura 50, Metodología GBM

El gradiente (por ejemplo, residual) se calcula, y un modelo entonces se encaja a los residuos para reducir al mínimo la función de pérdida. El modelo actual se añade al modelo anterior, y el procedimiento continúa por un número clasificado por el usuario un número específico de iteraciones.

Shrinkage: Una parte importante del método de gradient boosting machine es la regularización por la contracción o shrinkage, que consiste en la modificación de la regla de actualización de la siguiente manera:

$$F_m(x) = F_{m-1}(x) + \nu \cdot \gamma_m h_m(x), \quad 0 < \nu \leq 1,$$

donde el parámetro v es llamada la " tasa de aprendizaje " .

Empíricamente se ha encontrado que el uso de pequeñas tasas de aprendizaje (como v < 0,1) produce mejoras en la capacidad de generalización del modelo de gradient boosting machine sin encogerse (v = 1) Sin embargo, se trata al precio de aumentar el tiempo computacional, tanto durante el entrenamiento y consulta: a menor tasa de aprendizaje, más iteraciones.

Creación de un modelo de ensamblado y análisis de los resultados.

Como continuación al estudio sobre el aeropuerto de Washington, se procede a la creación de un modelo de Gradient Boosting Machine (GBM) y al análisis de sus resultados.

Se crea el modelo GBM usando training_data y se checkea la calidad del modelo comparándolo con crossvalidation_data.

En primer lugar, es importante conocer qué tipo de modelado en particular soporta nuestro modelo. Para ello, usaremos la función del paquete caret getModelInfo:

```
> getModelInfo()$gbm$type
[1] "Regression"     "Classification"
>
```

Como podemos observar, este output nos indica que gbm soporta tanto regression como classification. Como se va a trabajar con valores clasificadores binarios (0 y 1), necesitamos forzar a gbm a usar el modo classification.

Esto puede realizarse cambiando la variable en factor con el siguiente código:

```
# TRAINING & CROSSVALIDATION SETS

filtered_processed_dataset$IsDelayed<-
ifelse(filtered_processed_dataset$IsDelayed==1,'yes','nope')

filtered_processed_dataset$IsDelayed<-
as.factor(filtered_processed_dataset$IsDelayed)

outcomeName<-'IsDelayed'

predictorsNames<-
names(filtered_processed_dataset)[names(filtered_processed_dataset)!=outcom
eName]
```

```
# TRAINING & CROSSVALIDATION SETS

filtered_processed_dataset$IsDelayed<-ifelse(filtered_processed_dataset$IsDelayed==1,'yes','nope')
filtered_processed_dataset$IsDelayed<-as.factor(filtered_processed_dataset$IsDelayed)
outcomeName<-'IsDelayed'
predictorsNames<-names(filtered_processed_dataset)[names(filtered_processed_dataset)!=outcomeName]

inTrain<-caret::createDataPartition(filtered_processed_dataset$IsDelayed,p=.85,list=FALSE)
training_data<-filtered_processed_dataset[inTrain,]
crossvalidation_data<-filtered_processed_dataset[-inTrain,]
```

Figura 51, Training & crossvalidation sets

A continuación, se sustituye el modelo de regresión logística con el código de implementación de gradient boosting machine (GBM):

```
rownames(training_data)<-NULL
gbmGrid<-expand.grid(interaction.depth=c(1,5,9), n.trees= (35:40)*50,
shrinkage=0.1, n.minobsinnode=50))
```

gbmGrid: Función que establece los parámetros de búsqueda, la profundidad de las iteraciones, el número de árboles, el valor de shrinkage comentado anteriormente, en este caso 0.1, y el minobsinnode. Esta función influye directamente sobre el resultado de la predicción, pues a mayor profundidad y a mayor número de árboles más fiable es el resultado.

```
fitControl<-trainControl(
    method='cv',
    number=3,
    returnResamp='none',
    verbose=FALSE,
    summaryFunction=twoClassSummary,
    classProbs=TRUE)
```

fitControl: Función auxiliar para el control de ajuste del modelo.

```
g<-train(training_data[,predictorsNames], training_data[,outcomeName],
         method='gbm',
         trControl=fitControl,
         metric="ROC",
         tuneGrid=gbmGrid)
```

Iter	TrainDeviance	ValidDeviance	StepSize	Improve
1	0.4652	nan	0.1000	0.0157
2	0.4483	nan	0.1000	0.0075
3	0.4341	nan	0.1000	0.0058
4	0.4241	nan	0.1000	0.0045
5	0.4158	nan	0.1000	0.0032
6	0.4072	nan	0.1000	0.0034
7	0.4017	nan	0.1000	0.0022
8	0.3969	nan	0.1000	0.0018
9	0.3902	nan	0.1000	0.0026
10	0.3857	nan	0.1000	0.0017
20	0.3551	nan	0.1000	0.0003
40	0.3213	nan	0.1000	0.0011
60	0.2985	nan	0.1000	0.0001
80	0.2812	nan	0.1000	0.0001
100	0.2689	nan	0.1000	-0.0000
120	0.2566	nan	0.1000	-0.0000
140	0.2462	nan	0.1000	-0.0001
160	0.2373	nan	0.1000	-0.0001
180	0.2304	nan	0.1000	-0.0001
200	0.2228	nan	0.1000	-0.0002
220	0.2160	nan	0.1000	-0.0001
240	0.2098	nan	0.1000	-0.0001
260	0.2041	nan	0.1000	-0.0001
280	0.1980	nan	0.1000	-0.0002
300	0.1925	nan	0.1000	-0.0000
320	0.1878	nan	0.1000	-0.0001
340	0.1824	nan	0.1000	-0.0001
360	0.1779	nan	0.1000	-0.0001
380	0.1735	nan	0.1000	-0.0001
400	0.1694	nan	0.1000	-0.0002
420	0.1651	nan	0.1000	-0.0003
440	0.1610	nan	0.1000	-0.0002
460	0.1578	nan	0.1000	-0.0001
480	0.1545	nan	0.1000	-0.0001
1240	0.0677	nan	0.1000	-0.0001
1260	0.0663	nan	0.1000	-0.0001
1280	0.0650	nan	0.1000	-0.0001
1300	0.0638	nan	0.1000	-0.0000
1320	0.0626	nan	0.1000	-0.0001
1340	0.0612	nan	0.1000	-0.0001
1360	0.0600	nan	0.1000	-0.0001
1380	0.0588	nan	0.1000	-0.0001
1400	0.0576	nan	0.1000	-0.0001
1420	0.0565	nan	0.1000	-0.0000
1440	0.0553	nan	0.1000	-0.0000
1460	0.0543	nan	0.1000	-0.0001
1480	0.0532	nan	0.1000	-0.0001
1500	0.0522	nan	0.1000	-0.0001
1520	0.0512	nan	0.1000	-0.0000
1540	0.0502	nan	0.1000	-0.0000
1560	0.0495	nan	0.1000	-0.0001
1580	0.0486	nan	0.1000	-0.0000
1600	0.0475	nan	0.1000	-0.0000
1620	0.0465	nan	0.1000	-0.0001
1640	0.0455	nan	0.1000	-0.0000
1660	0.0447	nan	0.1000	-0.0001
1680	0.0439	nan	0.1000	-0.0000
1700	0.0432	nan	0.1000	-0.0001
1720	0.0423	nan	0.1000	-0.0001
1740	0.0414	nan	0.1000	-0.0000
1760	0.0407	nan	0.1000	-0.0001
1780	0.0399	nan	0.1000	-0.0000
1800	0.0392	nan	0.1000	-0.0001
1820	0.0384	nan	0.1000	-0.0000
1840	0.0377	nan	0.1000	-0.0000
1850	0.0374	nan	0.1000	-0.0000

Figura 52, Vista iteraciones GBM

El código del test de Gradient Boosting Machine queda de la siguiente forma:

```
187
188  # GRADIENT BOOSTING MACHINE
189
190  getModelInfo()$gbm$type
191
192  rownames(training_data)<-NULL
193  gbmGrid<-expand.grid(interaction.depth=c(1,5,9), n.trees=(35:40)*50, shrinkage=0.1,n.minobsinnode=50)
194
195  fitControl<-trainControl(
196      method='cv',
197      number=3,
198      returnResamp='none',
199      verbose=FALSE,
200      summaryFunction=twoClassSummary,
201      classProbs=TRUE)
202
203  g<-train(training_data[,predictorsNames], training_data[,outcomeName],
204           method='gbm',
205           trControl=fitControl,
206           metric="ROC",
207           tuneGrid=gbmGrid)
208
209  crossvalidation_data<-as.data.frame(crossvalidation_data)
210  rownames(crossvalidation_data)<-NULL
211
212  p_gbm<-predict(g, crossvalidation_data[,predictorsNames], type='prob')
213
214  auc<-pROC::roc(ifelse(crossvalidation_data[,outcomeName]=="yes",1,0),p_gbm[[2]])
215
216  print(auc)
217
218  plot(auc, main="ROC curve", xlab="False positive rate", ylab="True positive rate")
219
220
221
222
```

Figura 53, Código GBM

3.9 Resultados Gradient Boosting Machine

Como se ha indicado en el test de regresión logística, los rangos para el área under curve son:

- [0.5, 0.6): Test malo.
- [0.6, 0.75): Test regular.
- [0.75, 0.9): Test bueno.
- [0.9, 0.97): Test muy bueno.
- [0.97, 1): Test excelente.

```
Call:
roc.default(response = ifelse(crossvalidation_data[, outcomeName] ==     "yes", 1, 0), predictor = p_gbm[[2]])

Data: p_gbm[[2]] in 2377 controls (ifelse(crossvalidation_data[, outcomeName] == "yes", 1, 0) 0) < 172 cases (
Area under the curve: 0.9317
```

Area under the curve: 0.9317

Puede observarse como ha aumentado el parámetro AUC (“Area Under Curve”) hasta sobrepasar holgadamente el umbral de 0.9, comparándose con el test de regresión logística cuyo valor AUC es de 0.77.

Por tanto el test realizado con gradient boosting machine arroja una predicción con un auc de 0.9317, pudiendo considerarse un test de muy buena calidad.

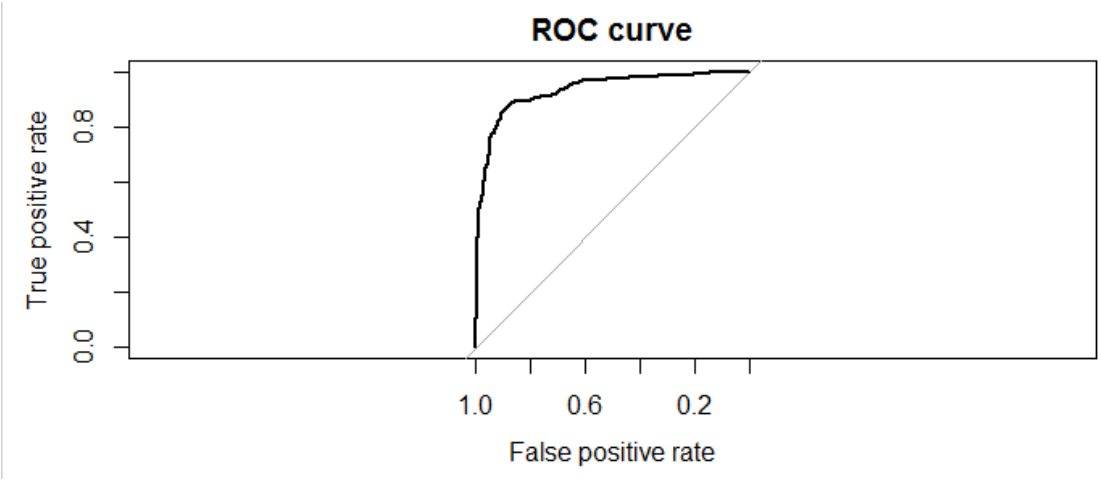


Figura 54, ROC curve, GBM

Puede observarse como también la curva ROC es mucho más pronunciada, arrojando mucho más espacio bajo ella en comparación con el test de regresión logística.

3.10 Modelo de Predicción Real

Ante el buen resultado obtenido en el modelo de gradient boosting machine, el objetivo de este proyecto ha sido crear un modelo de predicción real sin las variables de hora de llegada, tiempo de vuelo, etc., con el propósito de crear un modelo capaz de predecir sin dichas variables con que probabilidad un vuelo va a sufrir retraso en el aeropuerto de Seattle-Tacoma.

Se crea el nuevo modelo GBM usando training_data y se checkea la calidad del modelo comparándolo con crossvalidation_data como en el caso anterior.

En primer lugar, es importante conocer qué tipo de modelado en particular soporta nuestro modelo. Para ello, volvemos a utilizar la función del paquete caret getModelInfo:

```
getModelInfo()$gbm$type
```

Output:

```
> getModelInfo()$gbm$type
[1] "Regression"     "Classification"
>
```

Como podemos observar, este output nos vuelve a indicar que gbm soporta tanto regression como classification. Como se va a trabajar con valores clasificadores binarios (0 y 1), necesitamos forzar de nuevo a gbm a usar el modo classification.

A continuación, se procede a crear un nuevo dataset, llamado prediction_dataset. En dicho dataset, se filtran los valores de la base de datos para que contenga únicamente los vuelos con destino a Seattle, código “SEA”.

Esto se realiza mediante el siguiente código:

```
prediction_dataset<-dataset[dataset$Dest=="SEA",]
```

TailNum	ActualElapsedTime	CRSElapsedTime	AirTime	ArrDelay	DepDelay	Origin	Dest	Distance
N3CVAA	283	265	246	79	61	ORD	SEA	1721
N3GWAA	309	265	252	260	216	ORD	SEA	1721
N3JFAA	264	265	244	27	28	ORD	SEA	1721
N3FAAA	267	265	239	53	51	ORD	SEA	1721
N3JEAA	280	265	248	165	150	ORD	SEA	1721
N3BFAA	277	265	258	48	36	ORD	SEA	1721
N3BCAA	261	265	242	17	21	ORD	SEA	1721
N3CYAA	270	265	245	7	2	ORD	SEA	1721
N3BXAA	275	265	252	9	-1	ORD	SEA	1721
N3FYAA	281	265	252	18	2	ORD	SEA	1721
N3BCAA	277	265	257	12	0	ORD	SEA	1721
N3KAAA	257	265	238	-12	-4	ORD	SEA	1721
N3FHAA	276	265	258	9	-2	ORD	SEA	1721
N3ANAA	260	265	235	-3	2	ORD	SEA	1721
N3FUAA	265	265	245	-5	-5	ORD	SEA	1721
N3DDAA	275	265	255	12	2	ORD	SEA	1721

Además, se vuelve a crear otro dataset llamado prediction_processed_dataset, que dejaremos vacío para su rellenado mediante el siguiente código:

```
prediction_processed_dataset<-NULL
```

Y se añade al nuevo dataset las variables únicamente que sabemos en la salida de un vuelo: Hora de salida, Hora de salida de CRS, Distancia, y tiempo de Taxi.

Y a continuación, se convierten a valor numérico.

```
prediction_processed_dataset$DepTime<-as.numeric(dataset$DepTime)
prediction_processed_dataset$CRSDepTime<-as.numeric(dataset$CRSDepTime)
prediction_processed_dataset$Distance<-as.numeric(dataset$Distance)
prediction_processed_dataset$TaxiIn<-as.numeric(dataset$TaxiIn)
```

```
prediction_processed_dataset<-NULL

prediction_processed_dataset$DepTime<-as.numeric(dataset$DepTime)
prediction_processed_dataset$CRSDepTime<-as.numeric(dataset$CRSDepTime)
prediction_processed_dataset$Distance<-as.numeric(dataset$Distance)
prediction_processed_dataset$TaxiIn<-as.numeric(dataset$TaxiIn)
```

Se transforman los datos como en el modelo anterior para conseguir que su distribución se asemeje lo máximo posible a una normal:

```
# DATA TRANSFORMATION

prediction_processed_dataset<-as.data.frame(prediction_processed_dataset)
trans=preProcess(prediction_processed_dataset,c("BoxCox","center","scale"))
prediction_processed_dataset=data.frame(trans=predict(trans,prediction_processed_dataset))
```

Y se añaden las variables predictoras restantes que darán sentido a la variable independiente como factor:

```
# ADDING PREDICTORS

prediction_processed_dataset$Year<-as.numeric(as.factor(dataset$Year))
prediction_processed_dataset$Month<-as.numeric(as.factor(dataset$Month))
prediction_processed_dataset$DayofMonth<-as.numeric(as.factor(dataset$DayofMonth))
prediction_processed_dataset$DayofWeek<-as.numeric(as.factor(dataset$DayofWeek))
prediction_processed_dataset$UniqueCarrier<-as.numeric(as.factor(dataset$UniqueCarrier))
prediction_processed_dataset$FlightNum<-as.numeric(as.factor(dataset$FlightNum))
prediction_processed_dataset$TailNum<-as.numeric(as.factor(dataset$TailNum))
prediction_processed_dataset$Origin<-as.numeric(as.factor(dataset$Origin))
prediction_processed_dataset$Dest<-as.numeric(as.factor(dataset$Dest))
prediction_processed_dataset$Cancelled<-as.numeric(as.factor(dataset$Cancelled))
prediction_processed_dataset$CancellationCode<-as.numeric(as.factor(dataset$CancellationCode))
prediction_processed_dataset$Diverted<-as.numeric(as.factor(dataset$Diverted))
```

Como en los modelos anteriores, es necesario analizar la oblicuidad o skewness para comprobar cómo las variables han variado para asemejarse más a una distribución normal. Se realiza con el siguiente código:

```
# SKEWNESS ANALYSIS


skewness(dataset$DepTime)
skewness(prediction_processed_dataset$trans.DepTime)
skewness(dataset$CRSDepTime)
skewness(prediction_processed_dataset$trans.CRSDepTime)
skewness(dataset$Distance)
skewness(prediction_processed_dataset$trans.Distance)
skewness(dataset$TaxiIn)
skewness(prediction_processed_dataset$trans.TaxiIn)
```

Output:

```
> skewness(dataset$DepTime)
[1] 0.01611072
> skewness(prediction_processed_dataset$trans.DepTime)
[1] 0.01611072
> skewness(dataset$CRSDepTime)
[1] 0.06408598
> skewness(prediction_processed_dataset$trans.CRSDepTime)
[1] 0.06408598
> skewness(dataset$Distance)
[1] 0.8604543
> skewness(prediction_processed_dataset$trans.Distance)
[1] 0.09226696
> skewness(dataset$TaxiIn)
[1] 6.278846
> skewness(prediction_processed_dataset$trans.TaxiIn)
[1] -0.1358623
```

Se procede con la selección de características, para en primer lugar mostrar por pantalla aquellas variables con varianza 0.

```
nearZeroVar(prediction_processed_dataset)
```

```
> nearZeroVar(prediction_processed_dataset)
[1]  5  6 14 15 16
```

Y se eliminan del dataset con el siguiente código:

```
prediction_processed_dataset$Year<-NULL
prediction_processed_dataset$Month<-NULL
prediction_processed_dataset$Cancelled<-NULL
prediction_processed_dataset$CancellationCode<-NULL
prediction_processed_dataset$Diverted<-NULL
```

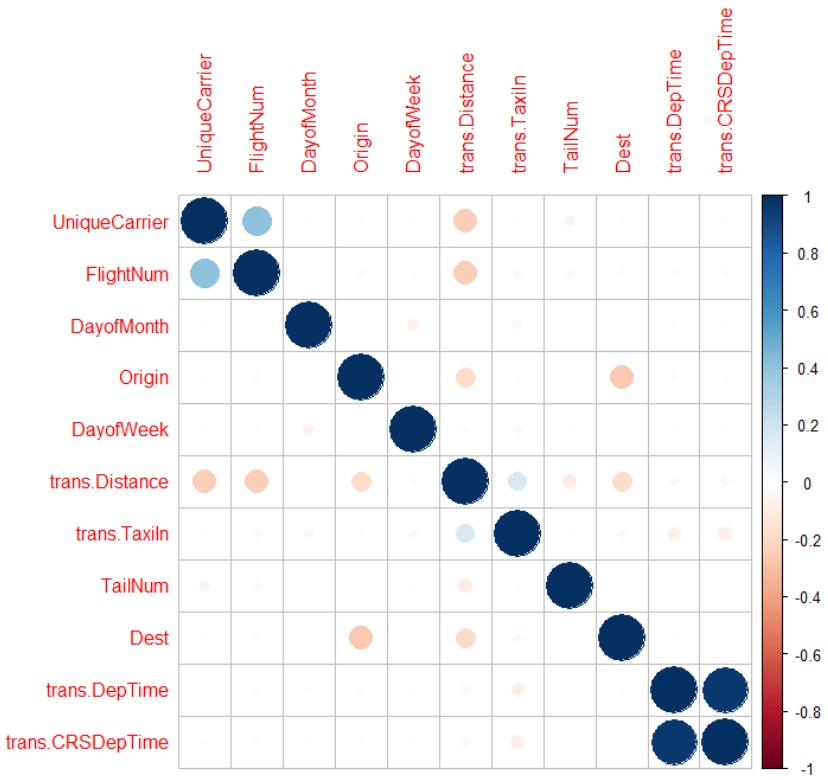
Resultando en un conjunto de datos limpio y homogéneo, listo para trabajar con el modelado predictivo.

A continuación, se realiza el análisis de correlación entre las variables, con el propósito de detectar aquellas que aportan información semejante, y que puedan ser eliminadas del modelo objetivo.

Se realiza con el siguiente código:

```
correlations<-cor(prediction_processed_dataset)

corrplot(correlations, order="hclust")
```

Resultando como output el gráfico siguiente:



Se puede deducir del gráfico anterior que sufren una correlación muy fuerte las variables Departure Time y CRS Departure Time, con un coeficiente cercano a 1, a causa de ser la variable que especifica el tiempo real de salida y el tiempo estimado por la compañía respectivamente.

También puede observarse una correlación notable entre las variables compañía y numero de vuelo.

Por último, se correlacionan las variables Distancia y numero de vuelo, Origen y Destino, Destino y Distancia, y Origen y Distancia.

A continuación, se filtran aquellas variables con una correlación superior al 0,75 con el propósito de crear un dataset únicamente con aquellas variables esenciales para el modelado predictivo, llamado filtered_processed_dataset.

```
# FILTERING

highCorr<-findCorrelation(correlations, cutoff = 0.75)
head(highCorr)
prediction_filtered_processed_dataset<-prediction_processed_dataset[,-highCorr]
```

Y se añade la variable IsDelayed:

```
# ADDING ISDELAYED

prediction_filtered_processed_dataset$IsDelayed<-dataset$IsDelayed
```

Se crean los dos sets, training y el set de validación cruzada o crossvalidation, que validarán la predicción y permitirán determinar la fiabilidad o precisión de la predicción:

TRAINING & CROSSVALIDATION SETS

```
prediction_filtered_processed_dataset$IsDelayed<-
ifelse(prediction_filtered_processed_dataset$IsDelayed==1,'yes','nope')
prediction_filtered_processed_dataset$IsDelayed<-
as.factor(prediction_filtered_processed_dataset$IsDelayed)
outcomeName<-'IsDelayed'
predictorsNames<-
names(prediction_filtered_processed_dataset)[names(prediction_filtered_proc
essed_dataset)!=outcomeName]

inTrain<-
caret::createDataPartition(prediction_filtered_processed_dataset$IsDelayed,
p=.85,list=FALSE)
training_data<-prediction_filtered_processed_dataset[inTrain,]
crossvalidation_data<-prediction_filtered_processed_dataset[-inTrain,]
```

```
# TRAINING & CROSSVALIDATION SETS

prediction_filtered_processed_dataset$IsDelayed<-ifelse(prediction_filtered_processed_dataset$IsDelayed==1,'yes','nope')
prediction_filtered_processed_dataset$IsDelayed<-as.factor(prediction_filtered_processed_dataset$IsDelayed)
outcomeName<-'IsDelayed'
predictorsNames<-names(prediction_filtered_processed_dataset)[names(prediction_filtered_processed_dataset)!=outcomeName]

inTrain<-caret::createDataPartition(prediction_filtered_processed_dataset$IsDelayed,p=.85,list=FALSE)
training_data<-prediction_filtered_processed_dataset[inTrain,]
crossvalidation_data<-prediction_filtered_processed_dataset[-inTrain,]
```

row.names	trans.DepTime	trans.Distance	trans.TaxiIn	DayofMonth	DayofWeek	UniqueCarrier	FlightNum	TailNum	Origin	Dest	IsDelayed
1	-0.235654410	0.86520631	0.9750681	1	3	1	69	558	45	55	yes
2	0.221662967	0.86520631	1.1693068	2	4	1	69	628	45	55	yes
3	-0.377714702	0.86520631	0.4773236	3	5	1	69	649	45	55	yes
4	-0.255114724	0.86520631	0.1463155	5	7	1	69	599	45	55	nope
5	0.015383640	0.86520631	0.4773236	6	1	1	69	648	45	55	yes
6	-0.362146451	0.86520631	0.4773236	7	2	1	69	527	45	55	nope
8	-0.438041675	0.86520631	0.4773236	9	4	1	69	559	45	55	nope
9	-0.443879769	0.86520631	0.1463155	10	5	1	69	541	45	55	nope
10	-0.438041675	0.86520631	0.9750681	12	7	1	69	616	45	55	nope
11	-0.441933738	0.86520631	0.7480133	13	1	1	69	524	45	55	nope
12	-0.449717863	0.86520631	0.9750681	14	2	1	69	660	45	55	nope

Una vez se tienen los datos preparados, se crea el método de modelado predictivo avanzado GBM Gradient Boosting Machine utilizando los nuevos datasets sin las variables que determinan la hora de llegada, con el propósito de predecir la probabilidad en la que un vuelo va a llegar con demora.

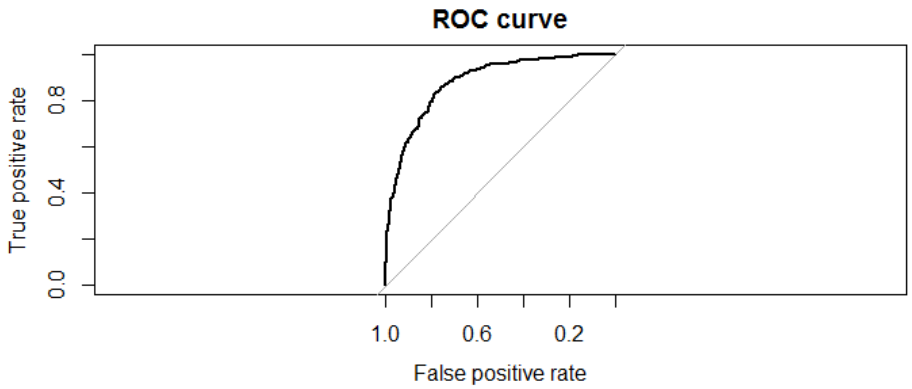
El código quedaría de la siguiente manera:

```
# GRADIENT BOOSTING MACHINE

getModelInfo()$gbm$type

rownames(training_data)<-NULL
gbmGrid<-expand.grid(interaction.depth=c(1,5,9), n.trees=100, shrinkage=0.1,n.minobsinnode=50)

fitControl<-trainControl(
  method='cv',
  number=3,
  returnResamp='none',
  verbose=FALSE,
  summaryFunction=twoClassSummary,
  classProbs=TRUE)

g<-train(training_data[,predictorsNames], training_data[,outcomeName],
         method='gbm',
         trControl=fitControl,
         metric="ROC",
         tuneGrid=gbmGrid)

crossvalidation_data<-as.data.frame(crossvalidation_data)
rownames(crossvalidation_data)<-NULL

p_gbm<-predict(g, crossvalidation_data[,predictorsNames], type='prob')

auc<-pROC::roc(ifelse(crossvalidation_data[,outcomeName]=="yes",1,0),p_gbm[[2]])

print(auc)

plot(auc, main="ROC curve", xlab="False positive rate", ylab="True positive rate")
```

	trans.DepTime	trans.Distance	trans.TaxiIn	DayofMonth	DayofWeek	UniqueCarrier	FlightNum	TailNum	Origin	Dest	IsDelayed
1	-0.4341496122	0.86520631	-0.8283631	26	7	1	69	631	45	55	no
2	0.4376724521	0.86520631	2.1147586	1	3	1	69	558	55	45	yes
3	0.3520470708	0.86520631	0.9750681	7	2	1	69	527	55	45	no
4	0.1944185279	0.86520631	1.1693068	13	1	1	69	524	55	45	no
5	0.1846883709	0.86520631	0.4773236	16	4	1	69	515	55	45	no
6	0.1613359942	0.86520631	2.5539814	20	1	1	69	515	55	45	no
7	0.2275010616	0.86520631	1.7375388	22	3	1	69	537	55	45	no
8	1.1888405700	1.66838598	0.7480133	7	2	1	106	584	27	55	yes
9	1.3328468931	1.66838598	0.1463155	11	6	1	106	618	27	55	yes
10	0.9786691795	1.66838598	1.9426832	19	7	1	106	576	27	55	no
11	-0.9595780885	0.78661090	1.6188688	19	7	1	139	521	55	15	no
12	-0.8369781107	0.78661090	1.1693068	24	5	1	139	609	55	15	no
13	-1.1931018557	1.66838598	2.5539814	2	4	1	107	559	55	27	no
14	-1.2067240754	1.66838598	0.7480133	16	4	1	107	543	55	27	no
15	-1.2047780440	1.66838598	-0.2724629	18	6	1	107	634	55	27	no
16	-1.2125621696	1.66838598	1.6188688	20	1	1	107	551	55	27	no
17	-1.1872637615	1.66838598	-0.2724629	29	3	1	107	596	55	27	no
18	-1.2047780440	1.66838598	0.1463155	30	4	1	107	644	55	27	no
19	0.9631009283	0.78661090	-0.2724629	4	6	1	454	542	15	55	no
20	1.0156437760	0.78661090	0.4773236	6	1	1	454	568	15	55	no
21	1.2199770723	0.78661090	0.4773236	12	7	1	454	508	15	55	yes
22	0.9708850539	0.78661090	0.1463155	16	4	1	454	655	15	55	no
23	0.9689390225	0.78661090	1.1693068	26	7	1	454	532	15	55	no
24	0.7334692239	1.97096343	0.7480133	1	3	1	463	1089	38	55	no
25	0.7373612867	1.97096343	0.7480133	4	6	1	463	1096	38	55	no
26	1.1654881933	1.97096343	2.0322651	26	7	1	463	1087	38	55	yes
27	0.7354152553	1.97096343	-0.8283631	27	1	1	463	1099	38	55	no
28	0.7315231925	1.97096343	0.7480133	30	4	1	463	1215	38	55	no
29	-1.5083589414	0.78661090	2.2619640	1	3	1	481	652	55	15	no

3.11 Resultados Modelo de Predicción Real

Se printan por pantalla la curva ROC y el "área under curve" AUC para determinar la precisión del test:



Como se ha indicado en el test de regresión logística, los rangos para el AUC son:

- [0.5, 0.6): Test malo.
- [0.6, 0.75): Test regular.
- [0.75, 0.9): Test bueno.
- [0.9, 0.97): Test muy bueno.
- [0.97, 1): Test excelente.

```
Data: p_gbm[[2]] in 2377 controls (ifelse(crossvalidation_data[, outcomeName] == "yes"
, 1, 0) 0) < 172 cases (ifelse(crossvalidation_data[, outcomeName] == "yes", 1, 0) 1).
Area under the curve: 0.9033
>
```

Area under the curve: 0.9033

Deducimos que la predicción es MUY BUENA.

Output de probabilidad de vuelo retrasado:

	nope	yes
1	0.98448083	0.0155191693
2	0.95227678	0.0477232196
3	0.95677980	0.0432201955
4	0.98729916	0.0127008398
5	0.98019409	0.0198059055
6	0.61625845	0.3837415515
7	0.97212845	0.0278715511
8	0.89868193	0.1013180676
9	0.90511759	0.0948824140
10	0.62461761	0.3753823917
11	0.99563185	0.0043681467
12	0.98634215	0.0136578499
13	0.99928565	0.0007143465
14	0.99760976	0.0023902363
15	0.99745564	0.0025443627
16	0.99887602	0.0011239817
17	0.99440820	0.0055918038
18	0.99630359	0.0036964122
19	0.99575737	0.0042426263
20	0.99864435	0.0013556485
21	0.99599823	0.0040017656
22	0.99898749	0.0010125105
23	0.99886526	0.0011347389
24	0.61937123	0.3806287725

CONCLUSIONES

Siendo el objetivo de este proyecto la investigación y predicción de los retrasos de vuelo, mediante, el análisis del impacto de las demoras en los vuelos en el sistema de transporte aéreo, se han podido extraer un conjunto de conclusiones que permitirán mejorar, comprender y optimizar el sistema de las demoras de vuelos en la operativa aeronáutica general.

Se ha extraído como conclusión el verdadero problema que suponen los retrasos de vuelo en el mundo aeronáutico, con datos tanto de Eurocontrol como de la agencia de estadística aeronáutica estadounidense.

Puede extraerse como conclusión que el tiempo medio de retraso de una aeronave con demora es de 9,3 minutos según la CODA Europea, lo que supone si lo relacionamos con la media de coste de aeronave parada, 25€/min, supone un coste para la aerolínea de aproximadamente 232€ por vuelo. Si contáramos Madrid con aproximadamente 600 vuelos diarios, supone 139.500€ en pérdidas diarias a causa del retraso de aeronaves, además de la satisfacción y comodidad del pasaje.

En el caso del aeropuerto de Seattle-Tacoma, aproximadamente un 33% de los vuelos diarios van demorados, por lo que una optimización del sistema operativo de la aeronave y una mejor respuesta ante las demoras permitirían detener el efecto dominó o en cadena de los retrasos de vuelos reduciendo el impacto, consecuencias y propagación de los mismos, resultando en el ahorro de millones de euros anuales y en el aumento de la satisfacción de los pasajeros.

Se ha podido extraer información de vital importancia acerca del mundo de la minería de datos y el modelado predictivo, como algunos modelos permiten mediante un previo análisis de datos estadísticos de cualquier aeropuerto del mundo, realizar una predicción fiable acerca de qué vuelos tienen más probabilidad de sufrir un retraso. El estudio de dichos modelos, ha permitido conocer en profundidad cómo funcionan algunos de los algoritmos de predicción científica más útiles y eficaces en la actualidad, como GBM o Random Forests, obteniendo una visión profunda de la precisión de dichos modelos y cuan adecuados son estos algoritmos para cualquier situación que requiera una predicción de alta precisión.

Además, se ha aprendido un nuevo lenguaje de programación estadística como es R, el proyecto de software libre y entorno de programación y análisis estadístico y gráfico. Este modelo ha servido de gran ayuda para programar los modelos predictivos avanzados de los que trata este proyecto, que aunque ha requerido de un gran consumo de recursos temporales para su aprendizaje, ha facilitado la tarea de desarrollo de los algoritmos.

Se ha conseguido crear dos algoritmos de modelado predictivo de los estudiados y analizados en la parte de revisión metódica e investigación, de los que utilizando los datos reales del aeropuerto de Seattle-Tacoma, ha permitido extraer como conclusión que el modelado predictivo es capaz de aprender de los resultados de una base de datos, para predecir con fiabilidad y eficiencia los retrasos de las aeronaves en cualquier aeropuerto del mundo.

Se extrae como conclusión de los modelos que el algoritmo de regresión logística ha arrojado un Area Bajo la Curva de 0,77, lo que es un test aceptable para la simplicidad metódica de este modelo. Esto quiere decir que a menudo no está directamente relacionado la complejidad o profundidad del algoritmo a utilizar con la bondad de ajuste o la precisión del resultado final.

Puede concluirse entonces que, en el caso de la predicción de los retrasos de vuelo, el modelo que ha arrojado un mejor resultado AUC y, por ende, una mayor precisión en la predicción, ha sido el algoritmo de Gradient Boosting Machine, con una Área Bajo la Curva de 0.9317, lo que se traduce en un muy buen test.

Además, se ha conseguido crear un modelo de predicción real, sin las variables de tiempo de llegada para asemejarse más a la operativa real de predicción de retrasos de vuelo, con resultados óptimos.

El algoritmo del modelo creado con Gradient Boosting Machine real, ha predicho con un notable acierto la probabilidad de que cada vuelo resulte retrasado o no con destino al aeropuerto de Seattle (Ej. Vuelo 1: No 0.98448083, Sí 0.0155191693)-Ver tabla anexa. Por lo que puede concluirse que es posible predecir con eficacia los retrasos en algunos vuelos, utilizando modelos de predicción científica.

Dicho algoritmo predictivo ha realizado una predicción con un Área Bajo la curva de 0.9033, lo que lo convierte en un test muy bueno.

La constante evolución del mundo aeronáutico requiere de una rápida adaptación y mejora tanto de los equipamientos e instalaciones, como de los procedimientos operacionales diarios en cualquiera de sus actividades, mejora que hoy en día apenas es perceptible.

La predicción de los retrasos de las aeronaves es una mejora esencial para la evolución de dicho sistema. Predecir los retrasos permitirá ahorrar muchísimos recursos tanto a las aerolíneas como a las empresas gestoras de aeropuertos y a las empresas de servicios en tierra, mejorando la calidad global del sistema aeronáutico general y aumentando la satisfacción de proveedores de servicios y clientes.

La evolución de las tecnologías permitirá gestionar las diferentes situaciones anómalas con mejor previsión y por ende, con mejor eficiencia y eficacia. Una mejora del modelo presentado en este trabajo, sería capaz de realizar una predicción en profundidad de alta precisión, pudiendo predecir el tiempo exacto de retraso en minutos, permitiendo una reasignación de recursos a tiempo para solventar los problemas de dicho retraso.

Raúl Monje Solá, a 04 de Julio de 2015

BIBLIOGRAFÍA

1 "Central Office for Delay Analysis - CODA | Eurocontrol." 2015. Accessed July 7. http://www.eurocontrol.int/articles/central-office-delay-analysis-coda.

2 "CODA Report May 2015." 2015. Accessed July 7. http://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda-reports/flad-may-2015.pdf.pdf.

3 "CODA Digest Q1 2015." 2015. Accessed July 7. http://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda-reports/digest-q1-2015.pdf.

4 "Flight Delays Are Costing Airlines Serious Money." 2015. Accessed July 7. http://mashable.com/2014/12/10/cost-of-delayed-flights/.

5 "Retrasos - Aena." 2015. Accessed July 7. http://www.aena.es/csee/Satellite/aeropuertos/es/Page/1048858945591/Retrasos.html.

6 "RITA | BTS | Title from h2." 2015. Accessed July 7. http://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp.

7 Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. doi:10.1007/978-1-4614-6849-3.

8 Hosmer, David W., and Stanley Lemeshow. 2000. *Applied Logistic Regression*. Wiley.

9 Einicke, G.A., and G.A. Einicke. 2012. *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*. Intech. http://www.intechopen.com/books/smoothing-filtering-and-prediction-estimating-the-past-present-and-future.

10 Denning, D.E. 1987. "An Intrusion-Detection Model." *IEEE Transactions on Software Engineering* SE-13 (2): 222–32. doi:10.1109/TSE.1987.232894.

11 Chandola, Varun, Arindam Banerjee, and Vipin Kumar. 2009. "Anomaly Detection." *ACM Computing Surveys* 41 (3): 1–58. doi:10.1145/1541880.1541882.

12 Roweis, S T, and L K Saul. 2000. "Nonlinear Dimensionality Reduction by Locally Linear Embedding." *Science (New York, N.Y.)* 290 (5500): 2323–26. doi:10.1126/science.290.5500.2323.

13 "Discover Feature Engineering, How to Engineer Features and How to Get Good at It - Machine Learning Mastery." 2015. Accessed July 7. http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.

14 Yan, Xin. 2009. *Linear Regression Analysis: Theory and Computing*. World Scientific. https://books.google.com/books?id=MjNv6rGv8NIC&pg=PA1.

15 Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–97. doi:10.1007/BF00994018.

16 Breiman, Leo. 2001. “Random Forests.” *Machine Learning* 45 (1): 5–32. doi:10.1023/A:1010933404324.

17 Quinlan, J.R. 1987. “Simplifying Decision Trees.” *International Journal of Man-Machine Studies* 27 (3): 221–34. doi:10.1016/S0020-7373(87)80053-6.

18 Bhadeshia, H. K. D. H. 1999. “Neural Networks in Materials Science.” *ISIJ International* 39 (10): 966–79. doi:10.2355/isijinternational.39.966.

19 De Rigo, D., Rizzoli, A. E., Soncini-Sessa, R., Weber, E., Zenesi, P., and D. de Rigo. 2001. *Neuro-Dynamic Programming for the Efficient Management of Reservoir Networks*. *Proceedings of MODSIM 2001, International Congress on Modelling and Simulation*. Canberra, Australia: Modelling and Simulation Society of Australia and New Zealand. doi:10.5281/zenodo.7481.

20 Tetko, Igor V., David J. Livingstone, and Alexander I. Luik. 1995. “Neural Network Studies. 1. Comparison of Overfitting and Overtraining.” *Journal of Chemical Information and Modeling* 35 (5): 826–33. doi:10.1021/ci00027a006.

21 *The Machine Learning Dictionary*. 2015. Accessed July 7. http://www.cse.unsw.edu.au/~billw/mldict.html#activnfn.

22 “Understanding the Reporting of Causes of Flight Delays and Cancellations | Bureau of Transportation Statistics.” 2015. Accessed July 7. http://www.rita.dot.gov/bts/help/aviation/html/understanding.html#q4.

23 Cheng Li, and Cheng Li. n.d. *A Gentle Introduction to Gradient Boosting*. Northeastern University.

24 “Code School - Try R.” 2015. Accessed July 7. http://tryr.codeschool.com/.

25 *Cross Validation*. 2015. Accessed July 7. http://www.cs.cmu.edu/~schneide/tut5/node42.html.

26 Picard, Richard R., and R. Dennis Cook. 1984. “Cross-Validation of Regression Models.” *Journal of the American Statistical Association* 79 (387): 575. doi:10.2307/2288403.

27 Mason, L., P. L. Bartlett, J. Baxter, and Marcus Frean. 1999. *Boosting Algorithms as Gradient Descent*. *Advances in Neural Information Processing Systems 12*. MIT Press.

28 “RITA | BTS | Transtats.” 2015. Accessed July 7. http://www.transtats.bts.gov/Fields.asp?Table_ID=236.

ANEXO

Algoritmo de modelado predictivo del caso real:

```
library(RSQLite)
library(DBI)
library(car)
library(ggplot2)
library(caret)
library(e1071)
library(corrplot)
library(ROCR)
library(pROC)

sqLiteConnect<-function(database,table){
  con<-dbConnect(RSQLite::SQLite(),dbname=database)
  result<-dbGetQuery(con,paste("select*from ontime"));
  dbDisconnect(con)
  return(result)
}

result <-
sqLiteConnect("C:/Users/Raul/Desktop/TFG/ontime.sqlite","ontime")

bad<-is.na(result[,"CarrierDelay"])
result[bad,"CarrierDelay"]<-0

bad<-is.na(result[,"WeatherDelay"])
result[bad,"WeatherDelay"]<-0

bad<-is.na(result[,"NASDelay"])
result[bad,"NASDelay"]<-0

bad<-is.na(result[,"SecurityDelay"])
result[bad,"SecurityDelay"]<-0
```

```
bad<-is.na(result[,"LateAircraftDelay"])
result[bad,"LateAircraftDelay"]<-0

table(result[,"CancellationCode"])

bad<-is.na(result[,"CancellationCode"])
result[bad,"CancellationCode"]<-"No code"

good<-complete.cases(result)
dataset<-result[good,]

# OVERVIEW OF DATASET

# str(dataset)

dataset$IsDelayed<-

factor(car::Recode(dataset$LateAircraftDelay,"0:15=0;else=1"),ordere
d=TRUE)



# ********** TASK 3 *********

# CREATING THE DATASET

prediction_dataset<-dataset[dataset$Dest=="SEA",]

prediction_processed_dataset<-NULL

prediction_processed_dataset$DepTime<-as.numeric(dataset$DepTime)
prediction_processed_dataset$CRSDepTime<-
as.numeric(dataset$CRSDepTime)
prediction_processed_dataset$Distance<-as.numeric(dataset$Distance)
prediction_processed_dataset$TaxiIn<-as.numeric(dataset$TaxiIn)
```

```
# DATA TRANSFORMATION


prediction_processed_dataset<-
as.data.frame(prediction_processed_dataset)

trans=preProcess(prediction_processed_dataset,c("BoxCox","center","s
cale"))

prediction_processed_dataset=data.frame(trans=predict(trans,predicti
on_processed_dataset))


# ADDING PREDICTORS


prediction_processed_dataset$Year<-
as.numeric(as.factor(dataset$Year))

prediction_processed_dataset$Month<-
as.numeric(as.factor(dataset$Month))

prediction_processed_dataset$DayofMonth<-
as.numeric(as.factor(dataset$DayofMonth))

prediction_processed_dataset$DayofWeek<-
as.numeric(as.factor(dataset$DayofWeek))

prediction_processed_dataset$UniqueCarrier<-
as.numeric(as.factor(dataset$UniqueCarrier))

prediction_processed_dataset$FlightNum<-
as.numeric(as.factor(dataset$FlightNum))

prediction_processed_dataset$TailNum<-
as.numeric(as.factor(dataset$TailNum))

prediction_processed_dataset$Origin<-
as.numeric(as.factor(dataset$Origin))

prediction_processed_dataset$Dest<-
as.numeric(as.factor(dataset$Dest))

prediction_processed_dataset$Cancelled<-
as.numeric(as.factor(dataset$Cancelled))

prediction_processed_dataset$CancellationCode<-
as.numeric(as.factor(dataset$CancellationCode))

prediction_processed_dataset$Diverted<-
as.numeric(as.factor(dataset$Diverted))
```

```
# SKEWNESS ANALYSIS

skewness(dataset$DepTime)
skewness(prediction_processed_dataset$trans.DepTime)
skewness(dataset$CRSDepTime)
skewness(prediction_processed_dataset$trans.CRSDepTime)
skewness(dataset$Distance)
skewness(prediction_processed_dataset$trans.Distance)
skewness(dataset$TaxiIn)
skewness(prediction_processed_dataset$trans.TaxiIn)

# HISTOGRAMS

hist(dataset$DepTime)
hist(prediction_processed_dataset$trans.DepTime)
hist(dataset$CRSDepTime)
hist(prediction_processed_dataset$trans.CRSDepTime)
hist(dataset$Distance)
hist(prediction_processed_dataset$trans.Distance)
hist(dataset$TaxiIn)
hist(prediction_processed_dataset$trans.TaxiIn)


# ******* FEATURE SELECTION ********

nearZeroVar(prediction_processed_dataset)

# DELETING THE 0VAR VARIABLES

prediction_processed_dataset$Year<-NULL
prediction_processed_dataset$Month<-NULL
prediction_processed_dataset$Cancelled<-NULL
prediction_processed_dataset$CancellationCode<-NULL
prediction_processed_dataset$Diverted<-NULL
```

```
# CORRELATION ANALYSIS


correlations<-cor(prediction_processed_dataset)


corrplot(correlations, order="hclust")


# FILTERING


highCorr<-findCorrelation(correlations, cutoff = 0.75)

head(highCorr)

prediction_filtered_processed_dataset<-
prediction_processed_dataset[,-highCorr]


# ADDING ISDELAYED


prediction_filtered_processed_dataset$IsDelayed<-dataset$IsDelayed




# TRAINING & CROSSVALIDATION SETS


prediction_filtered_processed_dataset$IsDelayed<-
ifelse(prediction_filtered_processed_dataset$IsDelayed==1,'yes','nop
e')

prediction_filtered_processed_dataset$IsDelayed<-
as.factor(prediction_filtered_processed_dataset$IsDelayed)

outcomeName<-'IsDelayed'

predictorsNames<-
names(prediction_filtered_processed_dataset)[names(prediction_filter
ed_processed_dataset)!=outcomeName]


inTrain<-
caret::createDataPartition(prediction_filtered_processed_dataset$IsD
elayed,p=.85,list=FALSE)

training_data<-prediction_filtered_processed_dataset[inTrain,]

crossvalidation_data<-prediction_filtered_processed_dataset[-
inTrain,]
```

```
# GRADIENT BOOSTING MACHINE


getModelInfo()$gbm$type


rownames(training_data)<-NULL

gbmGrid<-expand.grid(interaction.depth=c(1,5,9), n.trees=(100),
shrinkage=0.1,n.minobsinnode=50)


fitControl<-trainControl(

  method='cv',

  number=3,

  returnResamp='none',

  verbose=FALSE,

  summaryFunction=twoClassSummary,

  classProbs=TRUE)


g<-train(training_data[,predictorsNames],
training_data[,outcomeName],

         method='gbm',

         trControl=fitControl,

         metric="ROC",

         tuneGrid=gbmGrid)


crossvalidation_data<-as.data.frame(crossvalidation_data)

rownames(crossvalidation_data)<-NULL


p_gbm<-predict(g, crossvalidation_data[,predictorsNames],
type='prob')


auc<-
pROC::roc(ifelse(crossvalidation_data[,outcomeName]=="yes",1,0),p_gb
m[[2]])

print(auc)

plot(auc, main="ROC curve", xlab="False positive rate", ylab="True
positive rate")
```