Universitat Autònoma de Barcelona

escola d'enginyeria

Predicción y Análisis de los Retrasos en los Vuelos

Estudio del Aeropuerto de Arizona (E.E.U.U.)

Memoria del Proyecto de Fin de Grado
Gestión Aeronáutica
realizado por
Nerea Martínez Domenech
y dirigido por
Liana Napalkova
Sabadell, 08 de Febrero de 2016

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 ***Nerea Martínez Domenech***

I per tal que consti firma la present.

Signat: .....

Sabadell, ...de.....de 2016

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

Títol del projecte: Predicción y Análisis de los Retrasos en los Vuelos

Autor: Nerea Martínez Domenech **Data:** 08 de Febrer de 2016
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 temps de retards en els vols aeris dels aeroports d'Arizona (EUA), mitjançant la utilització de tècniques de mineria de dades, amb la finalitat de crear models de predicció que solucionin els problemes actuals dels retards aeris i les conseqüències que aquests produeixen en tot el sistema de transport aeri. Per a això s'han construït i comparat dos models de predicció: Random Forest i Gradient de Arboles Boosting per comprovar quin model s'adapta millor a l'hora de predir els retards en els vols.

Resumen: El proyecto siguiente trata de investigar los tiempos de retrasos en los vuelos aéreos de los aeropuertos de Arizona (EEUU), mediante la utilización de técnicas de minería de datos, con el fin de crear modelos de predicción que solucionen los problemas actuales de los retrasos aéreos y las consecuencias que estos producen en todo el sistema de transporte aéreo. Para esto se han construido y comparado dos modelos de predicción: *Random Forest* y Gradiente de Arboles *Boosting* para comprobar cual se adapta mejor a la hora de predecir los retrasos en los vuelos. Estos resultados han sido evaluados mediante el RMSE, don

Abstract: The next project is to investigate the time delays in the flights of airports in Arizona (USA), using data mining techniques in order to create predictive models that solve the current problems of air delays and the consequences they produce throughout the air transport system. To this have been built and compared two models of prediction: Random Forest and Gradient Boosting Trees to see which is best suited to predicting the delayed flights.

Tabla de contenido

Tabla de ilustraciones:

INTRODUCCIÓN

Estado del Arte y Motivación

El mundo del transporte aéreo como bien se conoce, es un mundo cambiante, con una evolución constante, tanto tecnológica, como legal y administrativamente.

Es por ello que durante estas dos últimas décadas la demanda del transporte aéreo ha experimentado un crecimiento considerablemente gracias a factores tales como: el aumento de las economías y de los ingresos de las personas, un aumento de la oferta de vuelos y la baja en el precio de los billetes aéreos.

Sin embargo, este acelerado desarrollo de la industria aérea ha provocado un aumento de la densidad de los flujos y la complejidad del manejo del tráfico aéreo, incrementando la congestión en los aeropuertos, y por ende, la probabilidad de retraso en los vuelos. A lo anterior, se agrega el aumento de las medidas de seguridad aplicadas en los aeropuertos y en los propios aviones, contribuyendo también al eventual retraso de los vuelos, e incluso a las cancelaciones de los mismos.

De ahí que las consecuencias actuales de la congestión y de los retrasos que conllevan se traduzcan en innumerables perdidas no sólo en tiempo y dinero para las compañías aéreas y aeropuertos que ofrecen estos servicios, sino también para los usuarios finales en relación a la calidad del servicio recibido.

La Comisión Europea por ejemplo, prevé que para 2030, de continuar con la evolución actual, 19 aeropuertos europeos, estarán saturados. La congestión consiguiente podría provocar retrasos que afectarán al 50 % de todos los vuelos de pasajeros y de mercancías.

Frente a ello, existen diferentes enfoques y mecanismos regulatorios entre los países para hacer frente a esta realidad. Mecanismos tales como técnicas de *data sharing* para mejorar la predictibilidad en el transporte aéreo

Es por ello, que este proyecto se centra en el importante papel de **la predicción y el análisis de los retrasos en los vuelos**, para así conocer más de cerca todos los factores involucrados en él y poder mejorar un servicio que como bien se ha comentado es el epicentro para la evolución y el crecimiento del sector aéreo en general.

Objetivo y tareas del proyecto

Objetivos generales:

El objetivo general de este proyecto consiste en analizar la aplicabilidad de las técnicas de las ciencias de datos para solucionar los problemas de predicción en los tiempos de llegada a los aeropuertos de los vuelos realizados, en este caso, en Arizona(EEUU).

Objetivos parciales:

Para lograr el objetivo general, se especifican a continuación una serie de objetivos parciales que conformarán el conjunto de hitos a corto plazo necesarios para dar respuesta al objetivo principal.

Los objetivos parciales son los siguientes:

- Aprender el conocimiento necesario sobre los impactos de los retrasos aéreos en el transporte aéreo con el fin de realizar una base solida teórica que fundamente el proyecto.

- Conocer las diversas técnicas de análisis y creación de modelos de predicción.

- Analizar y definir los datos históricos de los vuelos realizados en Arizona (Estados Unidos) desde enero a septiembre de 2015. Esta es la base de datos con la que se trabaja a lo largo del proyecto.

- Procesar y transformar todos los datos disponibles en la base de datos de los vuelos de Arizona con el fin de obtener unos datos claros y significativos para su posterior modelaje y extracción de conocimiento.

- Comparar y seleccionar las mejores técnicas y modelos de predicción.

- Creación de los modelos de predicción seleccionados mediante programación informática y técnicas de minería de datos con el fin de proporcionar un modelo preciso de predicción de los tiempos de retrasos aéreos. Estos modelos de predicción a crear son el *Random Forest* y el Gradiente de Árboles *Boosting.*

- Esquematizar y visualizar mediante gráficos los resultados obtenidos en la modelización de los modelos de predicción.

- Evaluación y comparación de los resultados de los modelos creados con el fin de extraer información relevante de cual modelo se adapta mejor para la predicción de los tiempos de vuelos aéreos.

Valor práctico

El valor práctico de este proyecto radica principalmente en solucionar un problema que hoy en día se produce en el transporte aéreo. Este problema es el de los retrasos aéreos.

Es un problema el cual, afecta en gran medida a todos los usuarios involucrados en su proceso, tanto compañías aéreas y aeropuertos en relación a las grandes

pérdidas económicas sufridas debido a la utilización ineficiente del sistema, y a los usuarios del transporte aéreo en relación con la pérdida de tiempo y de coste de oportunidad de no poder realizar otras tareas en ese tiempo perdido, con la consecuente percepción negativa en la calidad del servicio final.
Por ello este proyecto pretende cubrir una necesidad importante mediante el aprendizaje de conocimiento de las técnicas de minería de datos donde el análisis, la investigación, selección y construcción de modelos predictivos son la base para proporcionar una herramienta fiable para dar solución al problema de los retrasos aéreos.

Metodología

Para la realización y consecución de este proyecto y de los objetivos descritos anteriormente se ha llevado a cabo una serie de técnicas de aprendizaje automático, inteligencia artificial, programación informática, modelos de análisis comparativos y de predicción. Todo esto se puede englobar en el proceso de **minería de datos** ya que este abarca a todo el proceso de extracción de la información para luego transformarla a en una estructura comprensible para su uso posterior, objetivo éste, general del proyecto.
Toda esta metodología se desarrolla en cada una de las partes de este proyecto de la siguiente forma:
En relación con la realización de los primeros apartados teóricos de este proyecto se utilizan técnicas de filtrado, análisis comparativo y búsqueda intensiva de datos bibliográficos con el fin de proveer de información contrastada y consistente para este proyecto.
Para el segundo apartado de este proyecto en base a la parte práctica de creación de modelos de predicción para los retrasos aéreos, se ha llevado a cabo principalmente en un ambiente de programación informática mediante la utilización del programa **Pycharm.**
Pycharm es un entorno de desarrollo integrado (IDE) que utiliza un código con lenguaje **Python.** Es en este punto, donde se utilizan las técnicas de programación y modelado de estructuras informáticas para la creación en código python del modelo de predicción a analizar.
Para llevar a cabo toda la realización y construcción del modelo de predicción en código pyhon, paralelamente se sigue un proceso de minería de datos llamado **KDD (*Knowledge Discovery in Databases*)**, que quiere decir, "Descubrimiento de Conocimiento en Bases de Datos" y es en este, donde se encuentran los fundamentos básicos para el correcto seguimiento y realización del modelo. Seguidamente se detallan los pasos fundamentales utilizados en este proceso y las técnicas y teorías utilizadas:

- Análisis y selección de la información y los datos a utilizar para la creación del modelo: en este punto se utilizan **técnicas comparativas y analíticas**.
- Procesamiento y transformación de los datos: se utilizan técnicas de modelado tanto informático como estadístico, tales como, **la transformación de la asimetría**, **centrado y sesgo, estudio de la varianza y de la desviación típica y utilización de las teorías de distribución normal de Gauss.**
- Selección de las características más importantes en la base de datos en relación con la variable a predecir: se utilizan procedimientos como los de **ingeniería de características** (***feature engineering***) donde enfatizan la importancia de una buena preparación de las características y datos con el fin de obtener conocimientos relevantes para dar solución al problema en cuestión.
- Creación de los modelos de predicción: aquí se utilizan modelos básicamente centrados en los **problemas de regresión.** Los modelos utilizados se caracterizan, en grandes rasgos, por utilizar **algoritmos complejos de aleatoriedad, de construcción en forma de arboles y construcción mediante promedios.** Además para su correcto funcionamiento se utilizan métodos de validación como la **validación cruzada.**
- Análisis y evaluación de los resultados obtenidos de los modelos de predicción: se utilizan técnicas para medir y comparar la calidad de los modelos en relación con los resultados generados, tales como, medidas de **error medio cuadrático** *(Root Mean Square Error, RMSE) y* **análisis visuales mediante gráficos.**

Estructura de la tesis

7 Apartados, 29 Ilustraciones, 101 páginas.

- **Introducción:** Apartado que describe la situación de los retrasos en el mundo del transporte aéreo y ofrece al lector una visión de la estructura y contenido de este proyecto.
- **Análisis del impacto de los retrasos en la aviación**
- **Predicción de los retrasos aéreos:** Apartado donde se estudian los diferentes modelos de predicción y los pasos clave para su realización.
- **Caso de Estudio:** Caso de estudio real en el aeropuerto de Arizona(EEUU), y creación de los modelos predictivos *Random Forest* y Gradiente de Árboles *Boosting.*
- **Conclusiones**
- **Bibliografía**
- **Anexo:** Se muestra todo el código informático realizado en lenguaje Python.

1. ANALISIS DEL IMPACTO DE LOS RETRASOS EN LA AVIACIÓN

1.1 Causas de los retrasos y su efecto de propagación en el sistema

Los retraso en los vuelos son un problema grave y generalizado en muchas partes del mundo a día de hoy, dado que sus impactos repercuten negativamente para todos los usuarios implicados.

Pero... a qué se debe este problema? Porque se crean retrasos en los aeropuertos de alrededor del mundo?

Para responder a esta pregunta se empieza por explicar, *a grosso modo*, una de las situaciones negativas que actualmente se viene desarrollando en el transporte aéreo y que es causa importante de los retrasos ocasionados en él. Esta situación señalada, trasciende en un escenario donde la **congestión** es partícipe de ello. La saturación de muchos aeropuertos, así como de las infraestructuras y servicios de control del tráfico aéreo(*air traffic control*, ATC) hacen que el problema de los retrasos para los usuarios del transporte aéreo se haya convertido en algo relativamente habitual.

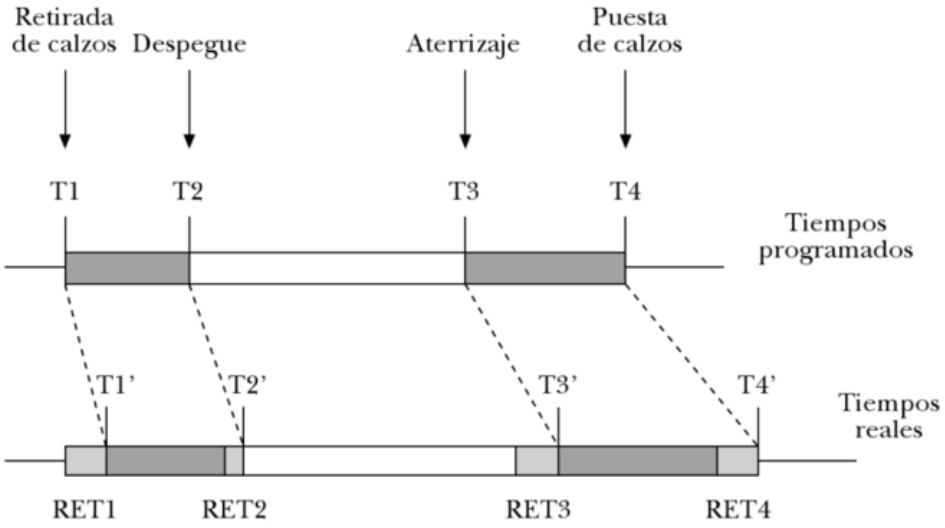
Esto se produce porque en la mayoría de los aeropuertos se permite un número ilimitado de aterrizajes y despegues, y las compañías aéreas añaden nuevos vuelos sin considerar la mayor congestión que ello provocará en las otras líneas aéreas. Todo ello conlleva a un exceso de vuelos programados en las horas *peak* (sobre el equilibrio eficiente) provocando entonces los retrasos tanto en los vuelos propios como en el resto.

Por ello, es de vital importancia que se realice una correcta planificación en la **programación de vuelos (*slots*)** que realizan conjuntamente los aeropuertos y las compañías aéreas. Ya que por ejemplo, si por cualquier motivo un avión no puede utilizar la franja horaria(*slot*) que tiene asignada para realizar una operación, como por definición, la capacidad del aeropuerto no puede ser expandida, debe ser trasladado a otro momento del tiempo con la consecuente externalidad negativa a otros vuelos si la programación de la siguiente hora esta completa.

Aun y así, en la programación de vuelos se ha de comentar que la existencia sistemática de los retrasos ya es tenida en cuenta a la hora de su elaboración, de forma que se añade un cierto margen sobre el tiempo medio que sería técnicamente necesario para realizar las operaciones.

En la ilustración 1.1 se muestra la naturaleza compleja del problema: en la parte superior del esquema se representan los tiempos programados para la operación de un vuelo, detallando los movimientos en tierra del avión(entre T1 y T2 en el aeropuerto de origen, T3 y T4 en el de destino), y la fase de vuelo(entre T2 y T3).

Ilustración 1. Esquema de retrasos de un vuelo



Fuente: Fundación BBVA

El retraso total que se observará en los datos estadísticos correspondientes a un vuelo será la combinación de varios tiempos de retraso que se pueden producir, de forma independiente en las diversas fases:

$$Retraso\ total = T4^{'} - T4 = RET1 + RET2 + RET3 + RET4$$

donde:

$RET1=$ diferencia entre los tiempos programados y reales de retirada de calzos. Este retraso puede deberse a muy diversas causas: pérdida de slot programado en el aeropuerto de destino, llegada tarde del avión previsto, problemas de tripulación, fallos mecánicos, pasajeros o equipaje.

$RET2=$ exceso de tiempo empleado en la fase de tierra del avión en el aeropuerto de origen, debido a saturación del espacio aéreo, problemas con equipos del aeropuerto(rampas, vehículos, etc.), necesidad de despegue del aeropuerto o problemas meteorológicos.

$RET3=$ retrasos en la fase de vuelo, por necesidad de realizar cambios en la ruta programada, o saturación del espacio aéreo.

$RET4=$ exceso de tiempo en la fase de tierra en el aeropuerto de destino, por pérdida de la posición de aparcamiento programada o por problemas con equipos del aeropuerto.

Como se puede observar, los retrasos aéreos son un fenómeno complejo ya que pueden tener su origen en múltiples causas y producirse durante las diferentes fases de una operación aérea.

Existen muchos factores que contribuyen al tiempo de realización de un vuelo. La puntualidad es el "producto final" de interacciones complejas entre las líneas aéreas, operadores de aeropuertos, y proveedores de servicios de navegación aérea (ANSP's) desde las fases de planificación y programación hasta el día de operación. Por esta razón, los efectos en la red tienen una fuerte impacto en el rendimiento del transporte aéreo.

Mencionar también que aunque este proyecto se centre en los retrasos de los vuelos, desde un punto de vista operativo, los vuelos que lleguen con **más de 15 minutos de antelación** a lo planificado pueden tener un parecido efecto negativo sobre la utilización de los recursos (respecto a la capacidad en la zona de maniobras de la terminal, en la capacidad de la ruta, la disponibilidad de la puerta, etc.) como en el caso de los retrasos de los vuelos.

Dicho esto, para conocer más de cerca la situación de las causas de los retrasos en el sistema del transporte aéreo, seguidamente se muestra información de diferentes tipos de análisis realizados por organizaciones de transporte aéreo en base a países, por un lado, de la Unión europea y por otro de Estados Unidos.

Respecto a la gestión del trafico aéreo(ATM) de Europa y los EE.UU. se ha de comentar que son muy comparables en términos de área geográfica, longitud media de vuelo, etc. Sin embargo, los EE.UU. controlan aproximadamente el 57% más de vuelos (IFR) con un 17% menos de los controladores aéreos y el 39% menos de personal en general.(Eurocontrol)

A la hora de analizar y clasificar las causas de los retrasos, en términos generales, los retrasos en los EE.UU. y Europa se pueden agrupar en las siguientes categorías principales, siempre teniendo en cuenta que para que se contabilice un retraso en una categoría causal, este retraso ha de haber sido mayor a 15 minutos:

- Aerolínea + *Turnaround* Local:
  La causa del retraso se debe a las circunstancias dentro del control local. Esto incluye las líneas aéreas u otras partes, tales como operadores de tierra que participan en el proceso de *turnaround* (por ejemplo, problemas de mantenimiento o de la tripulación, de limpieza de la aeronave, del equipaje de carga, abastecimiento de combustible, etc.).
- Tiempo Extremo: condiciones meteorológicas significativas (reales o previstas) que a juicio de la compañía, retrasa o impide la realización de un vuelo, como por ejemplo, la formación de hielo, tornados, tormentas de nieve, o huracanes. En los EE.UU., esta categoría es utilizada por las compañías aéreas para eventos muy raros como los huracanes y no es útil para entender el día a día de los impactos del clima. Para realmente ver la importancia que tienen los retrasos debido a condiciones no extremas en el

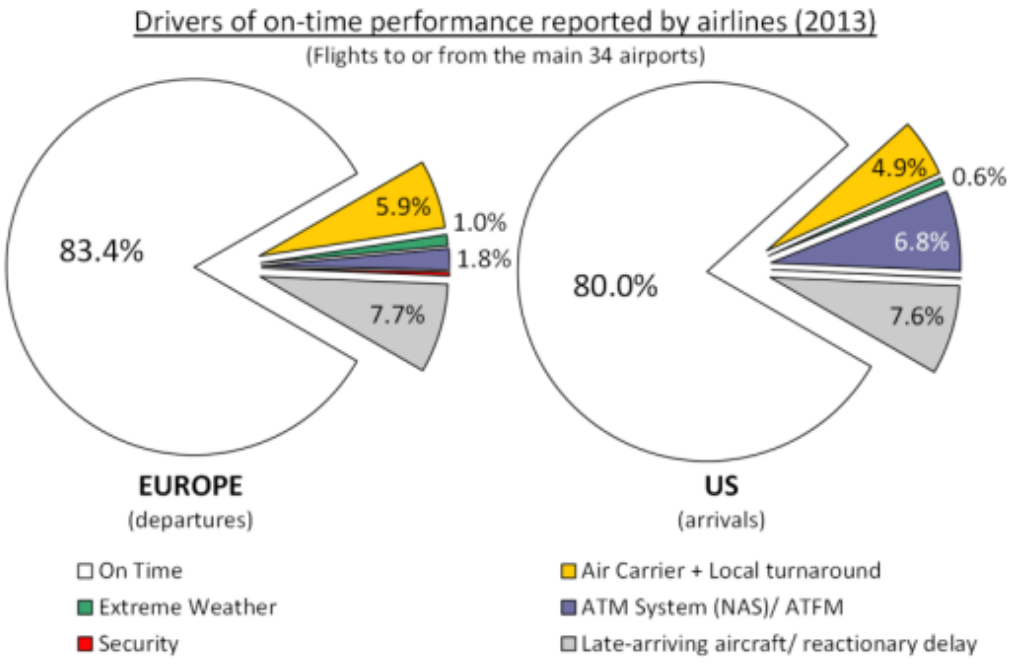
tiempo de EEUU se ha de observar la clasificación de retrasos por causas en el sistema ATM, que es donde éstas se conciben.

- Aeronaves que llegan tarde (o retraso reaccionario): Retraso a causa de la llegada tardía de la anterior aeronave, haciendo que el vuelo continuo a éste se retrase sin poder recuperar el tiempo durante la fase de *turnaround* en el aeropuerto.

  Debido a la naturaleza interconectada del sistema de transporte aéreo, largos retrasos primarios pueden propagarse por toda la red hasta el final del mismo día operacional.
- Seguridad: Retrasos causados por la evacuación de una terminal o zona determinada, re-acceso a las aeronaves debido a fallo de seguridad, equipos de control que no funcionan, y / u otras causas relacionadas con la seguridad.
- Sistema ATM(retrasos ATFM / NAS): Los retrasos atribuibles a ATM se refieren a una amplia gama de condiciones, como las condiciones no extremas del clima, las operaciones aeroportuarias, el volumen de tráfico pesado, Control del Tráfico Aéreo(ATC).

En la siguiente *ilustración 1.2* se muestra un desglose del rendimiento tanto de vuelos realizados dentro del tiempo programado por la compañía aérea, como de las causas de los retrasos primarios que se dan en las compañías aéreas de Estados Unidos y Europa para el año 2013. Estos retrasos se contabilizan y clasifican cuando el retraso es mayor de 15 minutos.

Ilustración 2. Rendimiento en el sistema de vuelos aéreos: UE vs. US

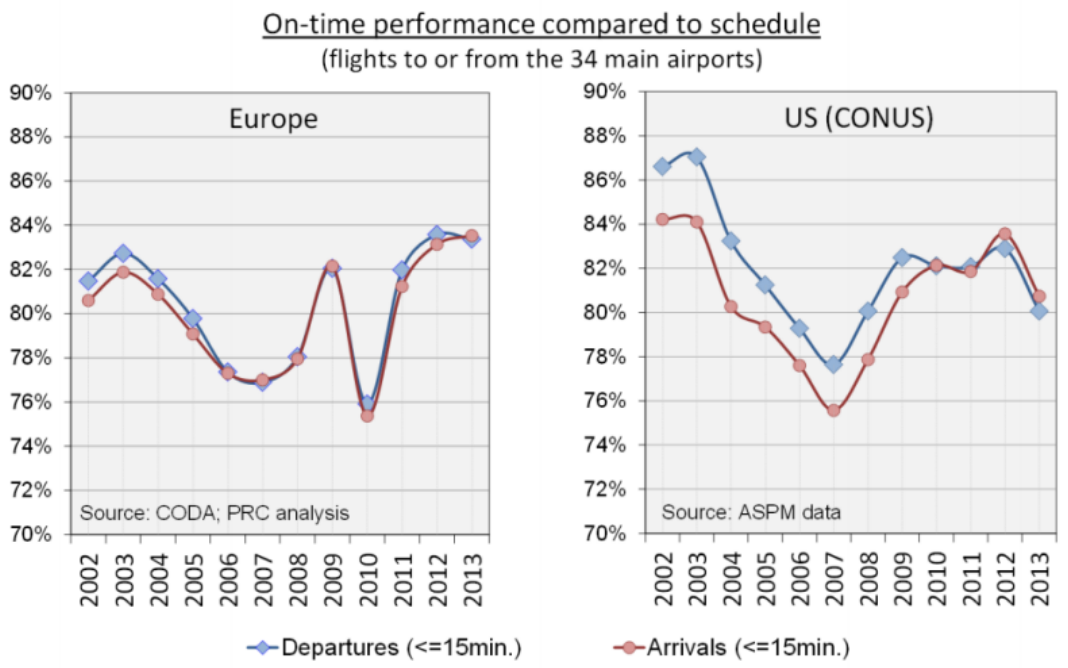


Fuente: Eurocontrol

Claramente se puede ver que a las aerolíneas estadounidenses se les atribuye una fracción mayor del retraso por causa del **sistema ATM** que para las aerolíneas Europeas donde su porcentaje más notable es en relación a problemas en las **compañías aéreas** y en los procedimientos y operaciones de la fase de ***tornaround***.
El porcentaje tan elevado referente al retraso por causa del sistema ATM(NAS) en Estados Unidos es debido mayoritariamente a problemas con **el mal tiempo.**

Además de esto se observa que para las compañías aéreas de Estados Unidos y Europa también se les atribuye una disminución del rendimiento de sus vuelos debido en gran parte por la **llegada tardía de las aeronaves y de los problemas de retrasos reaccionario.**
En el siguiente gráfico se observa el **rendimiento en tiempo** de los vuelos tanto en Europa como en Estados Unidos desde el 2002 hasta el 2013.

Ilustración 3. rendimiento en tiempo de Europa vs US



Fuente: Eurocontrol

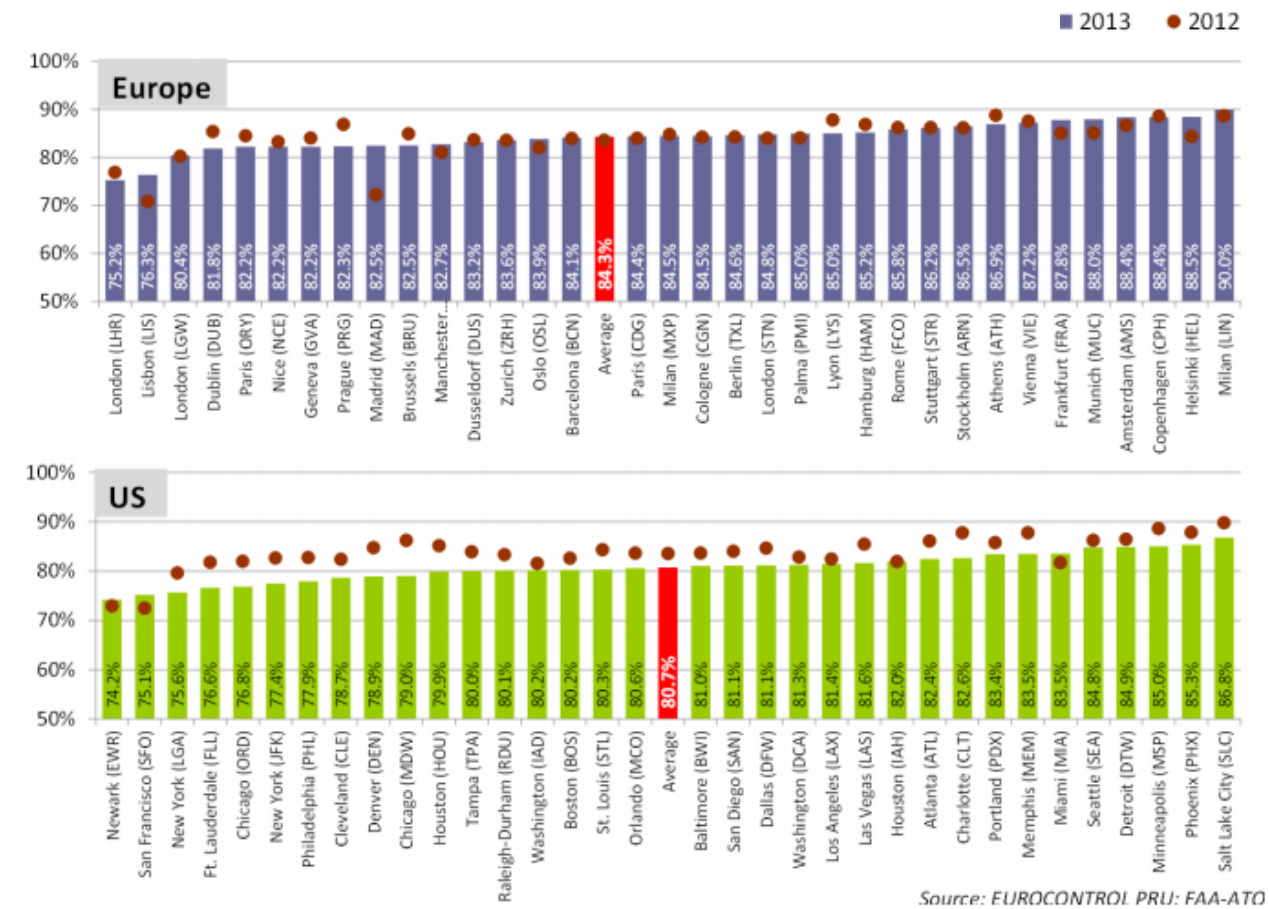
Se puede ver que de 2004 a 2009, el nivel de puntualidad de llegada fu similar tanto en los EE.UU. como en Europa. Estos cambia radicalmente en 2010 cuando la puntualidad degrada dramáticamente en Europa, pero siguió
mejorando en los EE.UU.. Esta mejora en el rendimiento tiene que ser visto en el contexto de la disminución del tráfico como resultado de la crisis financiera y económica mundial a partir de 2008.

En 2010, la puntualidad en Europa fue el peor registrado desde 2001, aunque el tráfico era todavía por debajo de los niveles de 2008. Los principales factores de este deterioro fueron un gran número de acciones industriales y más altos que los retrasos

habituales relacionados con el clima (nieve, condiciones de congelación) durante las temporadas de invierno de 2009 y 2010. La nube de ceniza volcánica en abril y mayo de 2010 tuvo un impacto limitado en la puntualidad, ya que la mayoría de los vuelos fueron cancelados y son, por tanto, excluidos del cálculo de indicadores de puntualidad.

De 2010 a 2012, la puntualidad en Europa ha mejorado de nuevo y ha continuado mejorando en los EE.UU.. Sin embargo en 2013, mientras que la puntualidad en Europa se mantuvo prácticamente sin cambios, la puntualidad en los EE.UU. vió un fuerte descenso que puede ser debido al tiempo desfavorable en 2013 en comparación con años anteriores.
La siguiente figura muestra la puntualidad en las llegadas de entre 34 aeropuertos principales de Europa y Estados Unidos en 2013.

Ilustración 4. Puntualidad en las llegadas en los aeropuertos de US vs. UE



En los EE.UU., Newark (EWR) tuvo el menor rendimiento en tiempo (llegadas), seguido de San Francisco (SFO) y Nueva York Laguardia (LGA). En comparación con 2012, sólo tres aeropuertos mostraron mejoras en la puntualidad de llegada. Estos incluyen San Francisco (+ 2,7% pt.), Miami (+ 1,8% pt.) Y Newark (+ 1,3% pt.).

En Europa, los dos aeropuertos de Londres (LHR, LBV) y Lisboa (LIS) tuvieron el nivel más bajo en puntualidad de las llegadas en 2013. En comparación con 2012, Madrid (+ 10,2% pt.), Lisboa (+ 5,5% pt.) y Helsinki (+ 4,1% pt.) muestran las mayores mejoras.

Como se puede ver, el rendimiento en tiempo de todo el sistema es el resultado de situaciones de contraste entre los aeropuertos.
Por ello es importante entender, como bien se ha mencionado, que el rendimiento puntual (*on-time performance*) es el **"producto final"** de interacciones complejas con muchos actores, incluyendo ATM. La puntualidad de llegada se ve influenciada por la puntualidad de salida en el aeropuerto de origen y, a menudo por los retrasos que ya se produjeron en vuelos anteriores.
Dependiendo del tipo de operación en los aeropuertos (*hub and spoke* vs. punto a punto) y el itinerario de rutas de la compañía, el rendimiento local puede tener un impacto ya no únicamente en las operaciones del propio aeropuerto, sino en toda la red a través de un **efecto dominó o reaccionario.**

1.2 Consecuencias de los retrasos

Las consecuencias de los retrasos en el transporte aéreo como bien se ha comentado con anterioridad afectan gravemente al buen rendimiento del sector aéreo, en particular, tanto para los empresarios de las compañías aéreas y aeropuertos, como para los usuarios del transporte aéreo, y en general, a todo el entorno que le rodea tanto económico como medio ambiental.

1.2.1 Consecuencias para las compañías aéreas y aeropuertos

En referencia con las consecuencias de los retrasos para los empresarios tanto de las compañías aéreas como de los aeropuertos se refiere a efectos monetarios mayoritariamente como veremos a continuación.

Por ejemplo, en Estados Unidos, los retrasos del transporte aéreo tienen consecuencias importantes para su economía. Sólo para los vuelos domésticos del año 2007, se estima que los pasajeros sufrieron una demora de 320 millones de horas, con un costo para la economía estadounidense de más de US$41 mil millones. También se ha estimado que debido a las demoras aumentó el costo de las operaciones aéreas domésticas en US$19 mil millones. (poner fuente de información)

Análisis más recientes demuestran datos similares a los comentados para el año 2007. En un análisis del año 2014 extraído de *Airlines for America* se muestra que el coste total directo de los retrasos en relación a las operaciones de aviones fue de $81.18 por minuto, mientras que los costes debidos a retrasos en relación con el total directo de operaciones fueron de $9.149**:**

Ilustración 5. Tabla de costes de los retrasos en 2014 para US

Calendar Year 2014	Direct Aircraft Operating Cost per Block Minute	Δ vs. 2013	2014 Delay Costs ($mil)
Fuel	$38.34	1.8%	$4,321
Crew – Pilots/Flight Attendants	18.95	7.8%	2,136
Maintenance	12.36	0.0%	1,393
Aircraft Ownership	8.52	-1.1%	960
Other	3.01	6.8%	339
Total Direct Operating Costs	**$81.18**	**2.7%**	**$9,149**

Fuente: Airlines for America

1.2.2 Consecuencias para los usuarios

Las consecuencias para los usuarios se radican básicamente en relación con el tiempo y el coste de oportunidad perdido a causa de los retrasos sufridos. A demás de una percepción menor en base a la calidad del sistema de transporte aéreo.

1.2.2.1 Marco regulatorio de los retrasos en US y Europa

Estados Unidos:

En este país el tratamiento de estas materias es fundamentalmente desregulado, situación que explica la ausencia de un marco regulatorio específico para el tratamiento de situaciones como los retrasos y cancelaciones de vuelo. Es común además que las aerolíneas tengan regulaciones de carácter voluntario en su "Compromiso de Servicio al Cliente", pero su aplicación no ha sido del todo bien evaluada por la autoridad federal, la que ha promovido cambios regulatorios en algunas materias, tales como el tiempo máximo en que se permite mantener en espera a pasajeros a bordo de un avión en pista (*Tarmac Delay*).

La única materia regulada en Estados Unidos es la **denegación de embarque** causado por la sobreventa de un vuelo –overbooking. En este caso, la ley exige compensación y otros beneficios para aquellos pasajeros con billete aéreo confirmado y que no pudieron viajar por causa de la sobreventa. Sin embargo, como bien se ha mencionado, el resto de las situaciones como es el caso de las **cancelaciones de vuelos**

o atrasos prolongados, no tienen ninguna forma de compensación reglamentada en Estados Unidos.

Europa:

En Europa a diferencia de los Estados Unido sí que existe una regulación para compensar a los usuarios frente a retrasos aéreos, denegaciones de embarque o cancelaciones de vuelos. Esta regulación se creó en el 2004 mediante la publicación del **Reglamento N° 261/2004** y se aplica a toda clase de vuelos, chárter incluidos, que hayan despegado de aeropuertos de un país de la Unión Europea al que resulte aplicable la normativa, así como también a aquéllos que, despegando desde un terminal aéreo ubicado en un tercer país, tenga como destino uno de aquéllos, cuando los transportistas aéreos encargados de efectuar los vuelos sean aerolíneas europeas procedentes de la Unión Europea. Respecto a los retrasos, el Reglamento N° 261/2004 establece máximos de tiempo de espera en función de la distancia del vuelo respectivo. Así, en caso de un retraso mayor a esos máximos, la compañía aérea debe compensar a los pasajeros afectados en cuestiones que van desde suministrar gratuitamente comida y refrescos suficientes hasta el reembolso del valor del pasaje. *(para más información de las cantidades a reembolsar visite la página web siguiente: http://www.seguridadaerea.gob.es/lang_castellano/particulares/derechos_pax/info_derechos/default.aspx)*

1.2.3 Consecuencias para el entorno

En última instancia y no menos importante los impactos de los retrasos aéreos también afectan en contra del medio ambiente.

Esto es así dado que si una aeronave por ejemplo, ha de estar más tiempo efectuando las **maniobras de espera en el aire** o **esperando en las rodaduras de la pista** debido a cambios en los tiempos programados de otros vuelos que afecten en su programa de vuelo actual, esto ser verá traducido en mayores cantidades de gasto de combustible por parte de las aeronaves repercutiendo así con mayor cantidad de emisiones que deterioran la calidad del aire en la atmosfera y además de una mayor contaminación en relación al ruido emitido por los movimientos de las aeronaves.

1.3 Enfoques para minimizar los retrasos aéreos

Para reducir los impactos mencionados de los retrasos en el sector aéreo, existen multitud de técnicas y procedimientos creados en diversos programas por organizaciones del transporte aéreo. La base de muchos de estos programas se centra en técnicas de **intercambio de datos** (*data sharing)* para mejorar la previsibilidad en el transporte aéreo y así poder controlar y disminuir los problemas de los retrasos en los vuelos. Algunos de estos programas se describen a continuación:

- **Airport Collaborative Decision Making** (**A-CDM**): La Toma de Decisiones Colaborativas para los Aeropuertos(A-CDM) es un programa conjunto europeo creado por las organizaciones Eurocontrol, ACI-Europe, CANSO (Civil Air Navigation Services Organisation) e IATA con el objetivo de mejorar la eficiencia de las operaciones aeroportuarias mediante la reducción de los retrasos, el aumento de la previsibilidad de los acontecimientos durante el progreso de un vuelo y la optimización de la utilización de recursos.
  Todo esto aumentará la capacidad en los aeropuertos participantes. Este objetivo se debe conseguir a través de la mejora del **intercambio de información** en tiempo real entre los operadores aeroportuarios, los operadores aéreos, operadores de tierra y control del tráfico aéreo. El concepto en sí, implica una implementación de un conjunto de procedimientos operativos y procesos automatizados. A-CDM ha sido implementado ya en un gran número de aeropuertos europeos, y en concreto, durante el año pasado en el aeropuerto del Prat de Barcelona.

- **System Wide Information Management** (**SWIM**): La Gestión de la Información de todo un Sistema(SWIM) es un programa de tecnología avanzada creado por la Administración Federal de Aviación(*Federal Aviation Administration, FAA*) para facilitar un mayor sistema de información a la Gestión del Tráfico Aéreo(ATM). La información se gestiona a lo largo de todo el ciclo de vida y en base a todo el sistema europeo ATM. Éste provee de acceso para la información de la aviación a través de una única conexión. SWIM utiliza una arquitectura orientada a servicios(SOA) que facilita la incorporación de nuevos sistemas y el intercambio de datos y aumenta la conciencia de la situación común. Cabe decir que SWIM facilita los requisitos de intercambio de datos para NextGen, convirtiéndose así en columna vertebral para la realización de sus metas.

- The **Next Generation Air Transportation System** (**NextGen**): El Sistema de Transporte Aéreo de Próxima Generación (NextGen) es un nuevo Sistema Nacional del Espacio Aéreo, el cual, propone transformar el sistema de control del tráfico aéreo de los Estados Unidos de un sistema basado en radar con comunicación por radio a uno basado en satélites. La tecnología GPS se utiliza

para acortar rutas y obtener trayectorias más eficientes con el fin de ahorrar tiempo y combustible, reducir los retrasos en el tráfico, aumentar la capacidad, y los controladores de permiso para supervisar y gestionar las aeronaves con mayores márgenes de seguridad. De este modo, las comunicaciones por radio serán reemplazadas cada vez más por el intercambio de datos y la automatización reducirá la cantidad de información que el personal de vuelo debe procesar a la vez.

- **Single European Sky ATM Research (SESAR):** SESAR es el nombre que se le ha dado al proyecto tecnológico y operativo para modernizar la Gestión del Tránsito Aéreo (ATM) en Europa y que complementa el marco regulatorio de la iniciativa comunitaria de Cielo Único Europeo. El objetivo primordial de SESAR es garantizar el desarrollo sostenible del transporte aéreo en Europa de forma eficiente y segura a través de un enfoque orientado a los resultados.
  Para afrontar el problema de los retrasos plantea herramientas tales como:
  - *User-Driven Prioritisation Process (UDPP) departures*: Se trata de un proceso de priorización en las salidas para los usuarios del tráfico aéreo, en el cual, se ofrece una herramienta que permite ganar eficiencia en el proceso de consulta y en la identificación de un compañero para poder realizar el cambio de *slot* si fuera necesario.
  - Gestión de la trayectoria inicial en cuatro dimensiones(i4D): Pretende superar las ineficiencias de los radares que utilizan los controladores aéreos(ya que estos solo predicen la trayectoria del avión de hasta 5 minutos por delante) mediante la conexión de las aeronaves y de los sistemas de tierra para así optimizar la trayectoria de la aeronave en tres dimensiones más el tiempo ofreciendo una mayor predictibilidad.

2. PREDICCIÓN DE LOS RETRASOS AÉREOS

2.1 Análisis de los modelos de predicción

Los **modelos predictivos** (poner referencia) son modelos de relación entre el rendimiento específico de un sujeto en una muestra y uno o más atributos o características del mismo sujeto. El objetivo del modelo es evaluar la probabilidad de que un sujeto similar tenga el mismo rendimiento en una muestra diferente. Esto permite valorar riesgos o probabilidades asociadas sobre la base de un conjunto de condiciones, guiando así al decisor durante las operaciones de la organización.

El análisis de estos modelos predictivos se engloba dentro de una aérea de la **Minería de Datos (*Data Mining*),** el cual, este último es un campo multidisciplinario que combina las aéreas de estadística, de aprendizaje automático(*machine learning*), de inteligencia artificial y el de la tecnología de base de datos, con el fin de, descubrir patrones en grandes volúmenes de conjuntos de datos (*Knowledge Discovery in Databases*, *KDD*)**.**

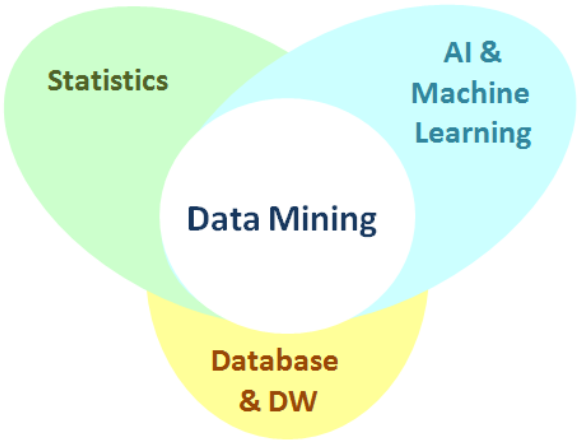


Ilustración 6. Esquema *Data Mining*

Este proceso se compone de una serie de complejas fases, las cuales, se intentan simplificar en la siguiente enumeración:

1. Comprensión: del negocio y del problema que se quiere resolver.
2. Determinación, obtención y limpieza: de los datos necesarios.
3. Creación de modelos matemáticos.
4. Validación, comunicación: de los resultados obtenidos.
5. Integración: si procede, de los resultados en un sistema transaccional o similar.

Estas fases se estudian más detalladamente en el ***apartado 2.2.,*** aplicándolas a las fases clave que se llevan a cabo en toda realización de un modelo de predicción.

Volviendo al punto de inicio de los modelos de predicción, existen diversas técnicas para su creación y desarrollo. Dichas técnicas, no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener unos resultados.

Las técnicas más representativas e influyentes en este proyecto se explican a continuación:

- **Modelos de Regresión *(Regression models):***

  El análisis de regresión es ampliamente utilizado para la predicción y previsión, donde su uso tiene especial importancia en el campo de **aprendizaje automático.** El análisis de regresión se utiliza también para comprender cuales de las variables independientes están relacionadas con la variable dependiente, y explorar las formas de estas relaciones.

  En circunstancias limitadas, el análisis de regresión puede utilizarse para inferir relaciones causales entre las variables independientes y dependientes. Sin embargo, esto puede llevar a ilusiones o falsas relaciones, ya que por ejemplo, una correlación entre variables no implica causalidad.

  Se han desarrollado muchas técnicas para llevar a cabo análisis de regresión. Su desempeño en la práctica depende de la forma del proceso de generación de datos, y cómo se relaciona con el método de regresión que se utiliza. Dado que la forma verdadera del proceso de generación de datos generalmente no se conoce, el análisis de regresión depende a menudo hasta cierto punto de hacer suposiciones acerca de este proceso.

  - Modelos de Regresión lineal:
    - Regresión lineal simple

      Dadas dos variables (Y: variable dependiente; X: independiente) se trata de encontrar una función simple (lineal) de X que nos permita aproximar Y mediante:

      $$\hat{Y} = a + bX$$

      a (ordenada en el origen, constante)

      b (pendiente de la recta)

      A la cantidad $e=Y-\hat{Y}$ se le denomina residuo o error residual.

- Regresión lineal múltiple(MLR)

  Es un método utilizado para modelar la relación lineal entre una variable dependiente (objetivo) y una o más variables independientes (predictoras):

$$observed\ data \rightarrow \quad y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p + \varepsilon$$

$$predicted\ data \rightarrow \quad y' = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$$

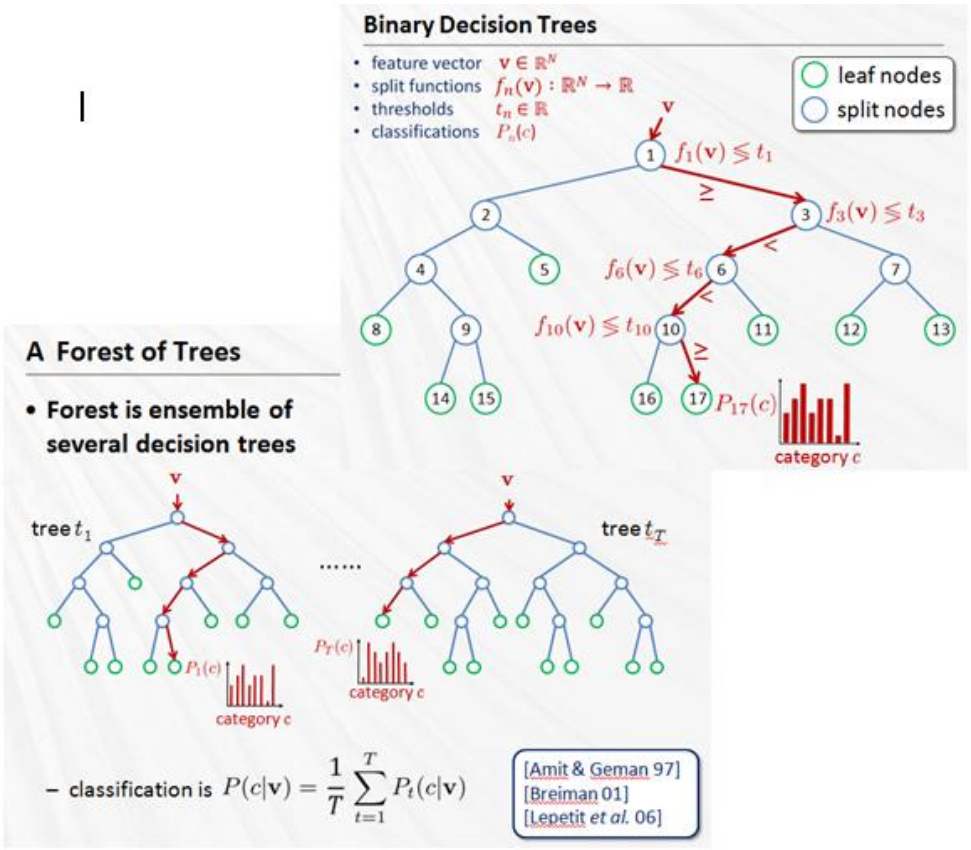
$$error \rightarrow \quad \varepsilon = y - y'$$

  MLR se basa en los mínimos cuadrados ordinarios(OLS), donde el modelo se ajusta de tal manera que la suma de los cuadrados de las diferencias de los valores observados y predichos se minimizan.

  El modelo de MLR se basa en varios supuestos (por ejemplo, los errores se distribuyen normalmente con media cero y varianza constante). Siempre que los supuestos se cumplen, los estimadores de regresión son óptimos en el sentido de que son **insesgados/centrados** (*unbiased*), **eficientes** y **consistentes**. Insesgado significa que el valor esperado del estimador es igual al valor verdadero del parámetro. Eficiente significa que el estimador tiene una varianza más pequeña que cualquier otro estimador. Consistente significa que el sesgo y la varianza del estimador de enfoque de cero como el tamaño de la muestra se aproxima al infinito.

- **Random Forests (Arboles aleatorios):**
  Es una combinación de **árboles de predicción** tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. En esencia, es una modificación sustancial de la técnica de ***bagging*** que construye una larga colección de árboles no correlacionados y luego los promedia.
  Seguidamente en la *ilustración 2.1* se puede ver una muestra en general de la estructura de los árboles de decisión y, en concreto, del proceso del modelo *Random Forests*:

Ilustración 6. Esquema Random Forests



Los árboles son los candidatos ideales para el *bagging,* dado que ellos pueden registrar estructuras de interacción compleja en los datos, con la característica de que si crecen suficientemente profundo, tienen relativamente alta imparcialidad (sin influencias de sesgos o desviaciones en la muestra).

En *random forests*, no hay necesidad de **validación cruzada**(*cross-validation*) o un conjunto de test(*test set*) separado para obtener una estimación **no sesgada** del error de prueba, ya que, esto anterior se estima internamente durante la ejecución, de la siguiente manera:

Cada árbol se construye utilizando una muestra ***bootstrap aggregating* (*bagging*)** diferente de los datos originales. Alrededor de un tercio de los casos se quedan fuera de la muestra de arranque(***out of bag, OOB***) y no se utiliza en la construcción del árbol.

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.

Las **ventajas** del *random forests* son:

- Es uno de los algoritmos de aprendizaje más certeros que hay disponibles. Para un set de datos lo suficientemente grande produce un clasificador muy certero.
- Rapidez de ejecución.
- Puede manejar cientos de variables de entrada sin excluir ninguna.
- Da estimaciones 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, ofreciendo así una localización de valores atípicos, y produciendo vistas interesantes de los datos.
- Ofrece un método experimental para detectar las interacciones de las variables.

Las **desventajas** del *random forests* son:

- Se ha observado que *Random forests* aún y proveer de una estimación de errores interna(*out of bag estimate, obb*) y de su cálculo de proximidades, en ciertos grupos de datos con tareas de clasificación/regresión ruidosas **se puede dar un sobreajusta(*overfit*)** aun y no siendo el caso habitual.
- 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.

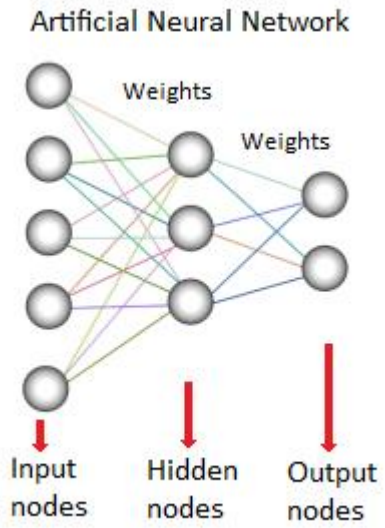
- Si los datos contienen grupos de atributos correlacionados con similar relevancia para el rendimiento, entonces los grupos más pequeños están favorecidos sobre los grupos más grandes.

- **Redes neuronales *(Artificial Neural Networks):***

Una red neuronal artificial (ANN) es un sistema que se basa en la red neural biológica, como el cerebro, aunque no es comparable ya que el número y la complejidad de las neuronas utilizadas en una red neural biológica es muchas veces más que los de una red neural artificial.

Un ANN se compone de una red de neuronas artificiales (también conocido como "nodos"). Estos nodos están conectados entre sí, y a la fuerza de sus conexiones entre sí se les asigna un valor basado en su fuerza: **de inhibición** (máxima siendo -1,0) o **de excitación** (ser máximo 1,0). Si el valor de la conexión es alta, entonces esto indica que hay una **conexión fuerte**. Dentro del diseño de cada nodo, se construye una función de transferencia. Hay tres tipos de neuronas en una ANN: **nodos de entrada, nodos ocultos y nodos de salida.** En el siguiente gráfico se muestra el diseño de una ANN:

Ilustración 7. Esquema de Red Neuronal Artificial



Los nodos de entrada toman la información, en la forma que puedan expresarse numéricamente. La información se presenta como valores de activación, donde cada nodo se le asigna un número, el cual contra mayor es el número, mayor es la activación. Esta información se pasa entonces a través de la red. Sobre la base de los puntos fuertes de conexión (pesos), la

inhibición o excitación, las funciones de transferencia, y el valor de activación son pasados de nodo a nodo. Cada uno de los nodos suma los valores de activación que recibe; a continuación, modifica el valor basado en su función de transferencia. La activación fluye a través de la red, a través de capas ocultas, hasta que llega a los nodos de salida. Finalmente estos nodos de salida reflejan la información procesada de manera significativa para el mundo exterior.

- **Gradiente de árboles *boosting (Gradient Boosting Trees):***

  El gradiente de arboles *boosting* es una generalización del método ***boosting*** con una diferencia arbitraria en la función de perdida. Crea modelos de predicción mediante el conjunto de modelos débiles de predicción y el uso de arboles de decisión.

  Es un proceso efectivo y preciso el cual se puede utilizar tanto en problemas de regresión como de clasificación.

  Las ventajas de GBRT son:

  - manejo natural de los datos de tipo mixto (= características heterogéneas)
  - La capacidad de predicción
  - Robustez a los valores atípicos en el espacio de salida (a través de robustas funciones de pérdida)

  Las desventajas de GBRT son:

  - Escalabilidad, debido a la naturaleza secuencial de *boosting*, donde para construir cada clasificador es necesario haber construido el anterior, esto hace que difícilmente pueda ser paralelizado.

2.2 Pasos clave para la creación de los modelos de predicción

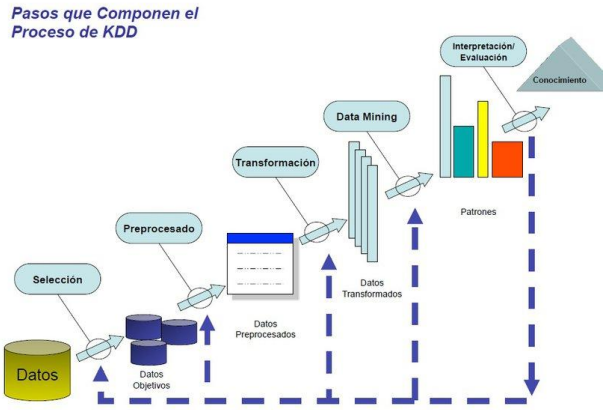
Para la creación de los modelos de predicción existen una serie de pasos clave para su correcto funcionamiento. Estos pasos se generalizan en las siguientes tareas:

1. **Selección del conjunto de datos (*Selection*)**, tanto en lo que se refiere a las variables objetivo (aquellas que se quiere predecir, calcular o inferir), como a las variables independientes (las que sirven para hacer el cálculo o proceso), como posiblemente al muestreo de los registros disponibles. Es decir se basa en la selección de subconjuntos de características o en la construcción de un nuevo conjunto de características con el fin de facilitar el aprendizaje y mejorar la generalización y la interpretación.

2. **Análisis de las propiedades de los datos**, en especial los histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).
3. **Transformación del conjunto de datos de entrada(*transformation*)**, se realizará de diversas formas en función del análisis previo, con el objetivo de prepararlo para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema, a este paso también se le conoce como **pre procesamiento** de los datos.
4. **Seleccionar y aplicar la técnica de minería de datos(*data mining*)**, se construye el modelo predictivo, de clasificación o regresión.
5. **Extracción de conocimiento**, mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un pre procesado diferente de los datos.
6. **Interpretación y evaluación de datos**, una vez obtenido el modelo, se debe proceder a su validación comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema. Si ninguno de los modelos alcanza los resultados esperados, debe alterarse alguno de los pasos anteriores para generar nuevos modelos.

Seguidamente se muestra una ilustración del procesos KDD que es originario en la minería de datos y se utiliza para la construcción de modelos, como en este caso, el de predicción:

Ilustración 8. Ilustración de las fases del proceso de Knowledge Discovery in Databases(KDD)



3. CASO DE ESTUDIO: Análisis de los retrasos en los vuelos de los aeropuertos de Arizona(E.E.U.U)

3.1 Descripción situacional de los datos a utilizar: Entorno, Aeropuertos y Compañías Aéreas de Arizona.

Para la creación del caso de estudio se utilizan datos históricos de los vuelos realizados, por las diferentes compañías aéreas operadoras, en los **Aeropuertos de Arizona (EEUU).**

Entorno de Arizona:

Arizona es uno de los 50 estados que conforman los Estados Unidos de América, localizado en el suroeste del país. Es muy conocido por su paisaje desértico, sus cactus y la cosmopolita ciudad de Phoenix.

Debido a su gran extensión y a las variaciones de altitud, el estado presenta una extensa variedad de condiciones climáticas localizadas.

Gran parte de Arizona tiene un clima árido o semiárido. Estas regiones reciben menos de 40 centímetros de lluvia al año, siendo muy calurosas en verano y suaves en invierno. No obstante, las regiones montañosas de mayor altitud poseen un clima más húmedo y frío. La temporada de monzón se extiende de mediados de julio a agosto y trae vientos, relámpagos, tormentas y lluvias torrenciales.

La mayoría del estado está escasamente habitado: la mayor parte de la población de Arizona se concentra en dos centros urbanos: Phoenix, la ciudad con mayor crecimiento de Estados Unidos, la mayor ciudad y capital del estado, y Tucson.(poner referencia)

Aeropuertos y compañías aéreas utilizadas en la base de datos:

Los aeropuertos y las compañías aéreas estadounidenses utilizadas en la base de datos se muestran en las dos siguientes tablas.

En esta primera tabla se pueden observar las 12 compañías aéreas las cuales son utilizadas para el análisis de predicción:

Ilustración 9. Tabla de Base de datos: Compañías aéreas

CODIGO IATA	ID DE LA COMPAÑIA AEREA	COMPAÑÍA AEREA
AA	19805	American Airlines
AS	19930	Alaska Airlines Inc
B6	20409	JetBlue Aiways
DL	19790	Delta Airlines Inc.
EV	20366	Atlantic Southeast Airlines
F9	20436	Frontier Airlines
HA	19690	Hawaiian Airlines
NK	20416	Spirit Airlines
OO	20304	SkyWest Airlines
UA	19977	United Airlines, Inc.
US	20355	US Airways
WN	19393	Southwest Airlines (Texas)

En la siguiente tabla 2, se muestran los 80 aeropuertos, tanto de destino y origen, analizados en los vuelos en la base de datos. Los aeropuertos de Arizona son los marcados en color amarillo. Cada vuelo analizado en este proyecto tiene como origen o destino uno de los 4 aeropuertos de Arizona(AZ):

Ilustración 10. Tabla de Base de datos: Aeropuerto

Código de aeropuerto IATA	Ciudad del Aeropuerto
ABQ	Albuquerque, NM
ANC	Anchorage, AK
ATL	Atlanta, GA
AUS	Austin, TX
BFL	Bakersfield, CA
BWI	Baltimore, MD
BOI	Boise, ID
BOS	Boston, MA
BUF	Buffalo, NY
BUR	Burbank, CA
CLT	Charlotte, NC
MDW	Chicago, IL
ORD	Chicago, IL
CVG	Cincinnati, OH
CLE	Cleveland, OH
CMH	Columbus, OH
DAL	Dallas, TX
DFW	Dallas/Fort Worth, TX
DEN	Denver, CO
DSM	Des Moines, IA
DTW	Detroit, MI
DRO	Durango, CO
ELP	El Paso, TX
FLG	Flagstaff, AZ
FLL	Fort Lauderdale, FL
FAT	Fresno, CA
GJT	Grand Junction, CO
HNL	Honolulu, HI
HOU	Houston, TX
IAH	Houston, TX
IND	Indianapolis, IN
OGG	Kahului, HI
MCI	Kansas City, MO
KOA	Kona, HI
LAS	Las Vegas, NV
LIH	Lihue, HI
LIT	Little Rock, AR
LGB	Long Beach, CA
LAX	Los Angeles, CA

SDF	Louisville, KY
MHT	Manchester, NH
MIA	Miami, FL
MKE	Milwaukee, WI
MSP	Minneapolis, MN
MRY	Monterey, CA
MTJ	Montrose/Delta, CO
BNA	Nashville, TN
MSY	New Orleans, LA
JFK	New York, NY
EWR	Newark, NJ
OAK	Oakland, CA
OKC	Oklahoma City, OK
OMA	Omaha, NE
ONT	Ontario, CA
MCO	Orlando, FL
PSP	Palm Springs, CA
PHL	Philadelphia, PA
PHX	Phoenix, AZ
PIT	Pittsburgh, PA
PDX	Portland, OR

RDU	Raleigh/Durham, NC
RNO	Reno, NV
SMF	Sacramento, CA
SLC	Salt Lake City, UT
SAT	San Antonio, TX
SAN	San Diego, CA
SFO	San Francisco, CA
SJC	San Jose, CA
SBP	San Luis Obispo, CA
SNA	Santa Ana, CA
SBA	Santa Barbara, CA
SEA	Seattle, WA
GEG	Spokane, WA
STL	St. Louis, MO
TPA	Tampa, FL
TUS	Tucson, AZ
TUL	Tulsa, OK
DCA	Washington, DC
IAD	Washington, DC
YUM	Yuma, AZ

Análisis actuales realizados por *U.S. Department of Transportation (US DOT),* ofrecen diferentes tipos de información y rankings de las compañías aéreas estadounidenses. En un ranking realizado para el mes de noviembre de 2015 se muestra el rendimiento en tiempo del vuelo de las compañías con los mejores y los peores porcentajes al respecto:

- En general:
  83.7 por ciento de las llegadas a tiempo

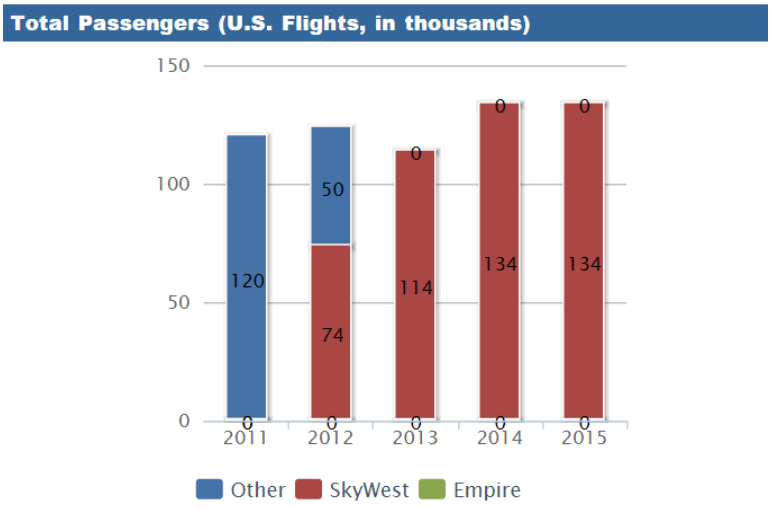
- Ranking de las compañías aéreas con en el porcentaje de llegadas más alto en relación con la hora programada:
  1. Hawaiian Airlines - 93,9 por ciento
  2. Delta Air Lines- 89,5 por ciento
  3. Alaska Airlines - 85,5 por ciento

- Ranking de las compañías aéreas con en el porcentaje de llegadas más bajo en relación con la hora programada:
  1. Frontier Airlines - 74,0 por ciento
  2. Spirit Airlines - 75,3 por ciento
  3. Aerolíneas ExpressJet - 80,8 por ciento

En relación con los 80 aeropuertos de la base de datos a utilizar en el análisis de predicción, a continuación se verá una ligera pincelada de las características de los 4 aeropuertos de Arizona. Como bien se ha comentado anteriormente estos son los protagonistas de todos los vuelos que se utilizarán para hacer el análisis de predicción posterior. Las características en relación con el flujo de pasajeros y las compañías que operan en ellos se detalla a continuación:

- Aeropuerto de Flagstaff Pulliam (FLG):
  Como se ve en la siguiente imagen, actualmente tiene como única compañía aérea operadora **SkyWest** con un total de 174 mil pasajeros.

Ilustración 11. Gráfico de Total de pasajeros para vuelos en FLG (en miles)



* Before October 2002, only carriers operating aircraft with more than 60 seats or 18,000 pounds in payload reported traffic data.
** 2015 represents data for November 2014 - October 2015.

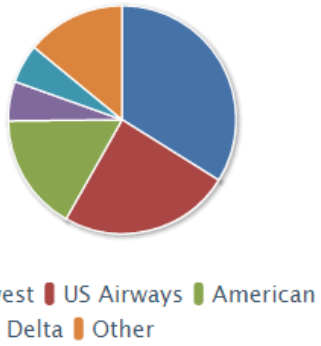
Además el aeropuerto de Flagstaff únicamente opera vuelos con destino al aeropuerto internacional de Phoenix(PHX).

- Aeropuerto Internacional de Phoenix-Sky Harbor(PHX):
  Es el aeropuerto **más grande y concurrido** de Arizona, y se encuentra entre los aeropuertos comerciales más grandes de Estados Unidos.
  En el siguiente gráfico se muestra como **Southwest** y **US Airways** son sus mayores compañías aéreas operadoras de entre otras, con aproximadamente 14 y 10 millones de pasajeros transportados respectivamente en el año 2015. Si a esto le sumamos los 17 millones de otras compañías operadoras en el aeropuerto suman un total de 41 millones de pasajeros transportados en 2015.

Ilustración 12. Gráfico de Partición de compañías en PHX para nov. 2014 - oct. 2015

Carrier Shares for November 2014 - October 2015

Carrier	Passengers	Share
Southwest	13,661	33.89%
US Airways	9,798	24.31%
American	6,708	16.64%
Mesa	2,257	5.60%
Delta	2,219	5.51%
Other	5,667	14.06%

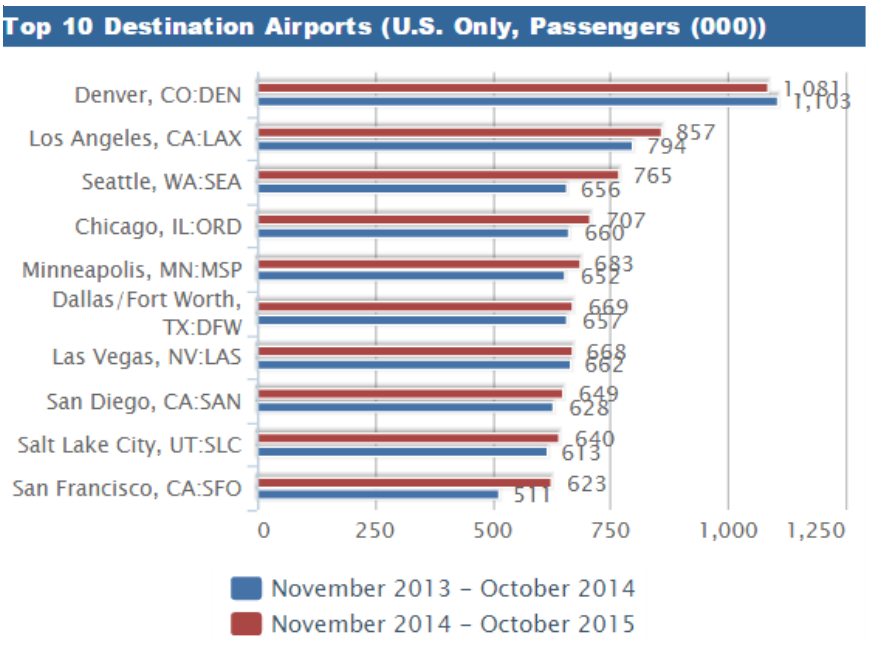


Based on enplaned passengers(000) both arriving and departing.

Fuente: Bureau of transportation statistics

En el siguiente gráfico se pueden observar cuáles son sus 10 aeropuertos de destino más utilizados. El aeropuerto de destino que encabeza la lista es Denver(CO) con 1 millón de pasajeros transportados en 2015.

Ilustración 13. Gráfico de Top 10 destinos aeropuertos de PHX (Pasajeros, en miles)



Source: T-100 Domestic Market (US Carriers).

Fuente: Bureau of transportation statistics

- <u>Aeropuerto Internacional de Tucson (TUS):</u>

  Es el segundo aeropuerto más **concurrido** de Arizona, después del de Phoenix.

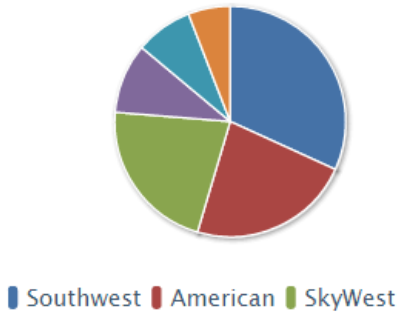
  En la siguiente imagen se pueden ver sus mayores compañías aéreas operadoras. Encabezando al mayor porcentaje se encuentra una vez mas Southwest con 986 mil pasajeros transportados para el año 2015 (de noviembre 2014 a octubre de 2015) un 31, 81% del total para ese periodo.

  En total se transportaron 3.100.000 de pasajeros para el periodo de noviembre 2014 a octubre de 2015.

Ilustración 14. Gráfico de Partición de compañías en TUS para nov. 2014 - oct. 2015

Carrier Shares for November 2014 - October 2015

Carrier	Passengers	Share
Southwest	986	31.81%
American	706	22.77%
SkyWest	673	21.70%
Mesa	304	9.80%
Delta	249	8.02%
Other	183	5.90%

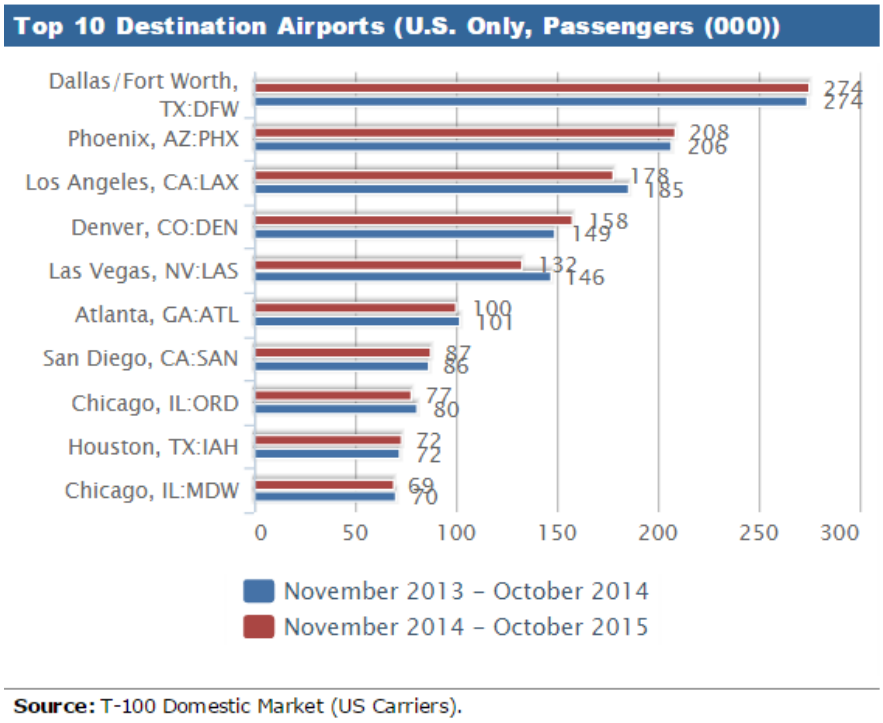


Based on enplaned passengers(000) both arriving and departing.

Fuente: Bureau of transportation statistics

En el siguiente gráfico se pueden observar cuáles son sus 10 aeropuertos de destino más utilizados y la variación de 2013-2014. El aeropuerto de destino que encabeza la lista es Dallas/Fort Worth, TX (DFX) con 274 mil pasajeros transportados en el periodo de noviembre 2014 a octubre de 2015.

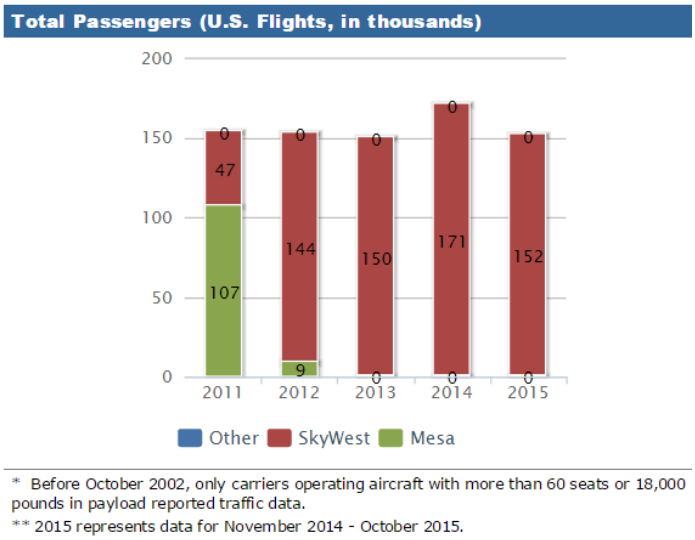
Ilustración 15. Gráfico de Top 10 destinos aeropuertos de TUS (Pasajeros, en miles)



Fuente: Bureau of transportation statistics

- <u>Aeropuerto Internacional de Yuma(YUM):</u>
  Es un aeropuerto de uso compartido junto con la *Marine Corps Air Station Yuma,* por ello se usa sobre todo para la aviación militar, aunque como podemos ver en el siguiente gráfico, actualmente opera mayoritariamente la compañía aérea **Skywest** con un movimiento de pasajeros para el año 2015 de **152 mil pasajeros.**

Ilustración 16. Gráfico de Total de pasajeros para vuelos en YUM (en miles)

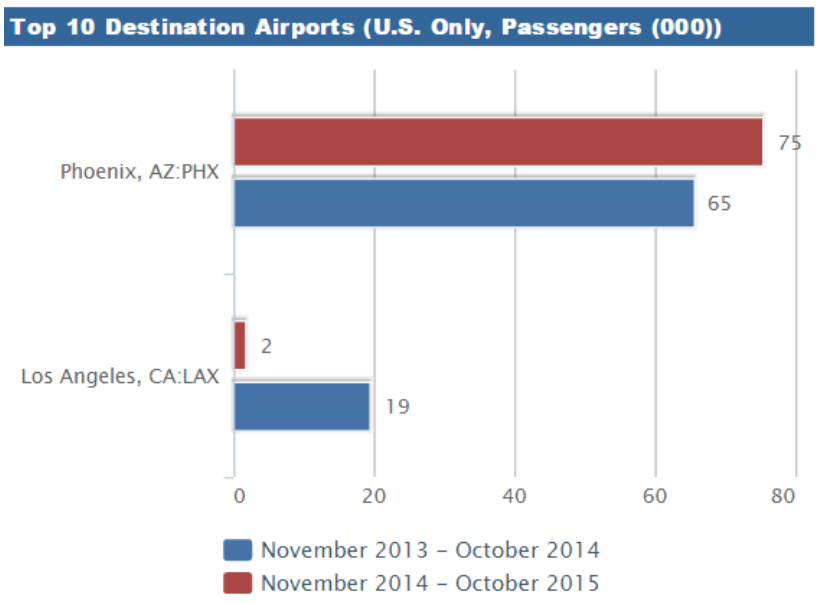


* Before October 2002, only carriers operating aircraft with more than 60 seats or 18,000 pounds in payload reported traffic data.
** 2015 represents data for November 2014 - October 2015.

Fuente: Bureau of transportation statistics

Los destinos del aeropuerto de Yuma únicamente son dos: aeropuerto de Phoenix (PHX) y aeropuerto de los Ángeles (LAX). El mayor porcentaje de pasajeros transportados es para el aeropuerto de Phoenix con 75 mil pasajeros transportados durante el periodo de noviembre 2014 a octubre de 2015. Aún y así se puede ver un incremento notable en relación con los pasajeros trasportados en el año 2014 al año 2015 para el aeropuerto de los Ángeles (LAX).Estos datos se pueden ver reflejados en el siguiente gráfico:

Ilustración 17. Gráfico de Aeropuertos de destino de YUM



Source: T-100 Domestic Market (US Carriers).

Fuente: Bureau of transportation statistics

3.2 Creación del modelo de predicción en Pycharm

3.2.1 Exploración de los datos

Como bien se ha visto en el anterior punto, para la creación del modelo de predicción se utilizan datos históricos de los vuelos realizados, por las diferentes compañías aéreas operadoras, (referencia de tabla) en los **Aeropuertos de Arizona (EEUU)** (referencia de tabla)**.**

En la base de datos, concretamente, se trabaja con **9 archivos**, los cuales, corresponden cada uno a un mes en particular del año **2015 (meses de enero a septiembre)**. Todos los archivos contienen la misma información en columnas, pero con distintos datos procedentes de cada mes en cuestión. En total se analizan **262.770 vuelos** realizados entre los 9 meses de 2015.

Como se verá seguidamente en la explicación, existen campos que contienen un ID, este ID representa un número de identificación asignado por US DOT, Departamento de Transporte de los Estados Unidos, que es la fuente de donde se extrae toda la información de la base de datos a analizar.

Dicha información está representada en 38 campos específicos (columnas), los cuales, se agrupan en 9 campos más generales, detallados a continuación:

- PERIODO DE TIEMPO
    - ***Year*****:** Año de realización del vuelo.
    - ***Quarter:*** Trimestre de realización del vuelo.
    - ***Month*****:** Mes de realización del vuelo.
    - ***Day_of_month*****:** Día del mes de realización del vuelo.
    - ***Day_of_week*****:** Día de la semana de realización del vuelo.
    - ***Fl_date***: Fecha completa de la realización del vuelo (dd/mm/aaaa).
- AEROLÍNEA
    - ***Unique_Carrier*****:** Código único de la compañía aérea. Cuando un mismo código sea utilizado por múltiples compañías aéreas, se utilizará un número sufijo para las compañías que primero lo utilizaron, por ejemplo: PA, PA (1), PA (2). Se utilizará este campo para el análisis a través de varios años.
    - ***Airline ID*****:** ID de la aerolínea. Es un número que identifica a una única aerolínea. Una única aerolínea se define como un ***"holding and reporting"*** bajo un mismo certificado DOT, independientemente de su código, nombre o sociedad de cartera/empresa.

- ***Carrier***: Código asignado por la IATA y comúnmente usado para identificar a las compañías aéreas. El código no siempre es único, ya que el mismo código puede ser asignado a diferentes empresas tras el paso del tiempo. Por lo tanto para crear análisis con espacios de tiempo diferentes es mejor utilizar el "*Unique_Carrier".*
- ***Fl_Num*:** Número de vuelo.

- ORIGEN
  - ***Origin_Airport_ID***: ID del aeropuerto de origen. Se utilizará este campo para analizar el aeropuerto a través de varios años ya que un aeropuerto puede cambiar su código y además puede ser reutilizado.
  - ***Origin_Airport_Seq_ID***: ID de la secuencia del aeropuerto de origen. Identifica a un único aeropuerto en un punto dado en el tiempo. Atribuye al aeropuerto, como el nombre o coordenadas, los cuales, pueden cambiar con el tiempo.
  - ***Origin_City_Marquet_ID***: ID del mercado de la ciudad de origen. Este campo se utiliza para consolidar los aeropuertos que sirven a un mismo mercado de la ciudad.
  - ***Origin***: Origen del aeropuerto. Campo que muestra el código IATA del aeropuerto de origen.
  - ***Origin_City_Name***: Nombre de origen de la ciudad.

- DESTINACIÓN

  La descripción de los campos de este apartado es el mismo que para el campo de origen, con la única diferencia que éste informa de los datos de destino.
  - ***Dest_Airport_ID***: ID que identifica el destino de un único aeropuerto. Al igual que el ID del aeropuerto de origen, se utilizará este campo para analizar el aeropuerto a través de varios años ya que un aeropuerto puede cambiar su código y además puede ser reutilizado.
  - ***Dest_Airport_Seq_ID***: ID de la secuencia del aeropuerto de destino. Identifica a un único aeropuerto en un punto dado en el tiempo. Atribuye al aeropuerto, como el nombre o coordenadas, los cuales, pueden cambiar con el tiempo.
  - ***Dest_City_Market_ID***: ID del mercado de la ciudad de destino. Este campo se utiliza para consolidar los aeropuertos que sirven a un mismo mercado en la ciudad.
  - ***Dest***: Destino del aeropuerto. Origen del aeropuerto. Campo que muestra el código IATA del aeropuerto de destino.
  - ***Dest_City_Name***: Nombre de la ciudad de destino.

- INFORMACIÓN/PLAN DE SALIDA
    - ***Dep_Time***: Tiempo actual de salida del vuelo (hora local: hhmm).
    - ***Dep_Delay_New***: Diferencia en minutos entre la hora de salida prevista y la hora de salida real. Nos muestra si hay retraso o no. (X = 0 → Sin retraso o salidas anticipadas a la hora prevista / / X > 0→ Con retraso).
    - ***Dep_Del15***: Indicador de retrasos en las salidas de 15 o más minutos (1= Si hay retrasos de 15 o más minutos).
    - ***Taxi_Out***: Tiempo en minutos del *Taxi Out* (desde que el avión está rodando hasta que sale de pista).
    - ***Wheels_Off***: Momento en el que las ruedas (aeronave) están fuera del suelo, es decir, momento en que la aeronave despega.

- INFORMACIÓN/PLAN DE LLEGADA
    - ***Wheels_On***: Momento en el que las ruedas (aeronave) están en tierra, momento en que la aeronave aterriza.
    - ***Taxi_in***: Tiempo en *Taxi* _in(desde que el avión aterriza hasta que llega al finger).
    - ***Arr_Time***: Tiempo actual de llegada (hora local: hhmm).
    - ***Arr_Delay_New***: Diferencia en minutos entre la hora de llegada prevista y la hora de llegada real. Las llegadas antes de tiempo se establecen con un 0.
    - ***Arr_Delay15***: Indicador de retrasos en las llegadas de 15 o más minutos (1= Cuando hay retrasos de 15 o más minutos).

- CANCELACIONES
    - ***Cancelled*****:** Indicador de cancelación de vuelo (1= Cuando hay cancelaciones)
    - ***Cancellation_Code*****:** Especifica la razón de la cancelación mediante un código.

  Cuando tengamos el indicador a 1 conforme hay cancelaciones, los campos de información de salida e información de llegada, estarán en blanco, no contendrán información.

- SUMARIO DE VUELO
    - ***Distance*****:** Distancia entre aeropuertos (millas).

- CAUSAS DE LOS RETRASOS
    - ***Carrier_Delay***: Retraso a causa de la aerolínea (minutos).Por ejemplo: problemas de manteniemiento o en la tripulacion, limpieza de los aviones, equipaje de carga, abastecimiento de combustible, etc.)
    - ***Weather_Delay***: Retraso a causa del temporal (minutos) real o previsto.

- ***NAS_Delay***: Retraso a causa del Sistema Aéreo Nacional (minutos).Se refiere a 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.
- ***Security_Delay***: Retraso a causa de la seguridad (minutos). Por ejemplo: por evacuación de una terminal o explanada, re-acceso a las aeronaves debido a un fallo de seguridad, equipos de control que no funcionan y / o las largas colas de más de 29 minutos en las áreas de detección.
- ***Late_Aircraft_Delay***: Retraso a causa de la llegada tardía de la aeronave, haciendo que el vuelo continuo a éste se retrase. (minutos).

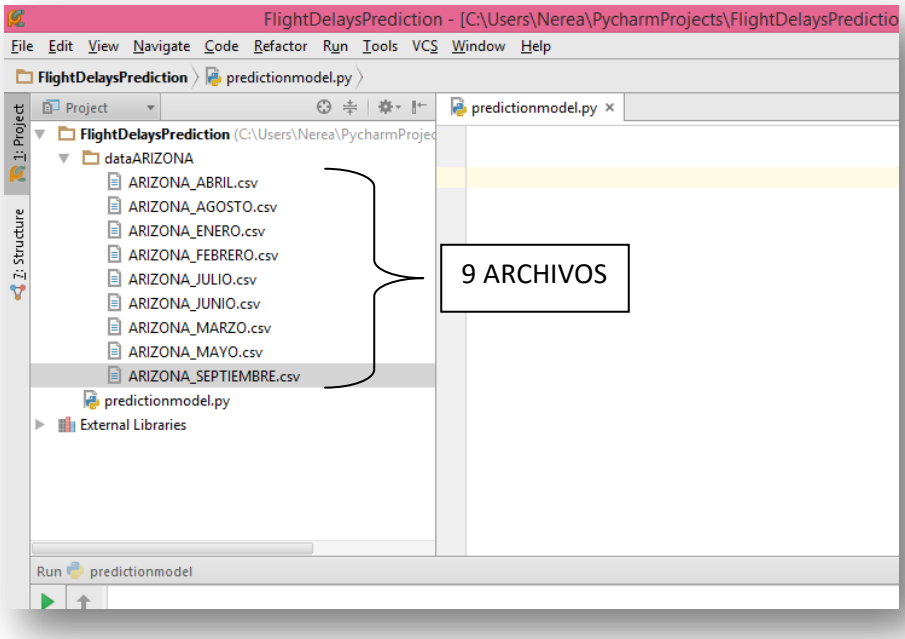
Sólo tendremos información de las causas de los retrasos si ha habido retrasos de más de 15 minutos en las salidas, llegadas o en ambas.

Para analizar toda esta información se utiliza un lenguaje de programación informática llamado ***Python.*** Este lenguaje se desarrolla en un programa informático llamado ***Pycharm.***

Dicho esto, a continuación, se verán los pasos fundamentales para la creación del nuevo proyecto, y su posterior volcado y manejo de los 9 archivos en el programa Pycharm:

Primero, ha sido necesario crear un nuevo proyecto, con el nombre de **"FlightDelaysPrediction".** Dentro de este proyecto se ha creado un fichero Python llamado **"predictionmodel.py",** que es donde se desarrollará el código en Python.

Seguidamente, dentro del proyecto principal se ha creado una carpeta llamada "dataARIZONA", la cual, contendrá los 9 archivos copiados con la información de los vuelos realizados en los Aeropuertos de Arizona.



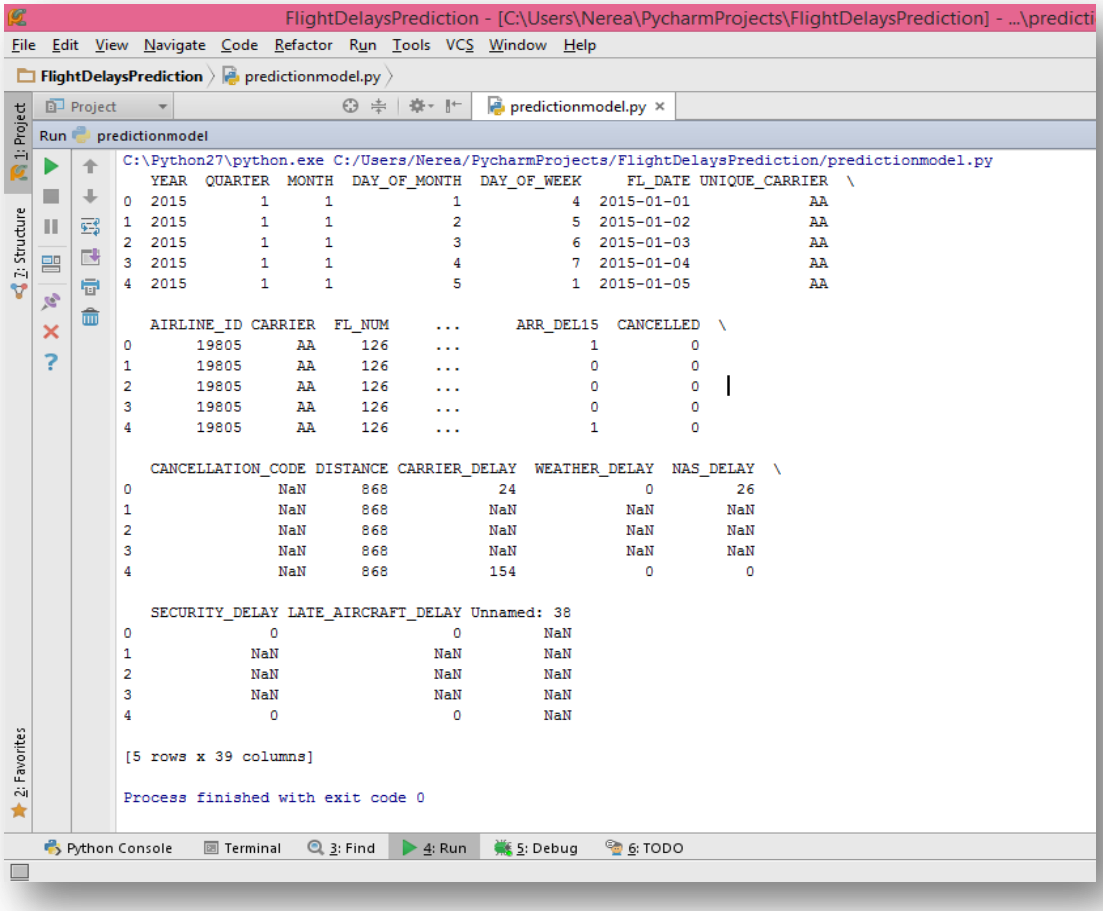
Para poder procesar y cargar todos los datos de los 9 archivos en el fichero de Python, se ha tenido que empezar a crear el siguiente código:

```
import pandas as pd

_author_ = 'Nerea'

df1 = pd.read_csv("dataARIZONA/ARIZONA_ENERO.csv")
df2 = pd.read_csv("dataARIZONA/ARIZONA_FEBRERO.csv")
df3 = pd.read_csv("dataARIZONA/ARIZONA_MARZO.csv")
df4 = pd.read_csv("dataARIZONA/ARIZONA_ABRIL.csv")
df5 = pd.read_csv("dataARIZONA/ARIZONA_MAYO.csv")
df6 = pd.read_csv("dataARIZONA/ARIZONA_JUNIO.csv")
df7 = pd.read_csv("dataARIZONA/ARIZONA_JULIO.csv")
df8 = pd.read_csv("dataARIZONA/ARIZONA_AGOSTO.csv")
df9 = pd.read_csv("dataARIZONA/ARIZONA_SEPTIEMBRE.csv")


result = (df1, df2, df3, df4, df5, df6, df7, df8, df9)
df = pd.concat(result)

print df.head()
```

Aquí lo que se ha hecho es que desde una de las muchas librerías que tiene el lenguaje Python, en este caso desde **"pandas",** se ha llamado a la función *"read"* para que nos cargue y nos lea los 9 archivos.

Seguidamente todos estos 9 archivos que están separados y que han sido asignados a *data frames* distintas(df1, df2, df3, df4, df5, df6, df7, df8, df9), se han ajuntado en un único archivo ***(DataFrame)*** llamado **"df"** mediante el comando ***concat*** de la librería de pandas.

Finalmente, para poder mostrar los resultados de la unión de todos los archivos, hemos "*printeado*" por pantalla las primeras filas del *data frame* **"df"** con el comando "**head()".** Los resultados han sido los siguientes:



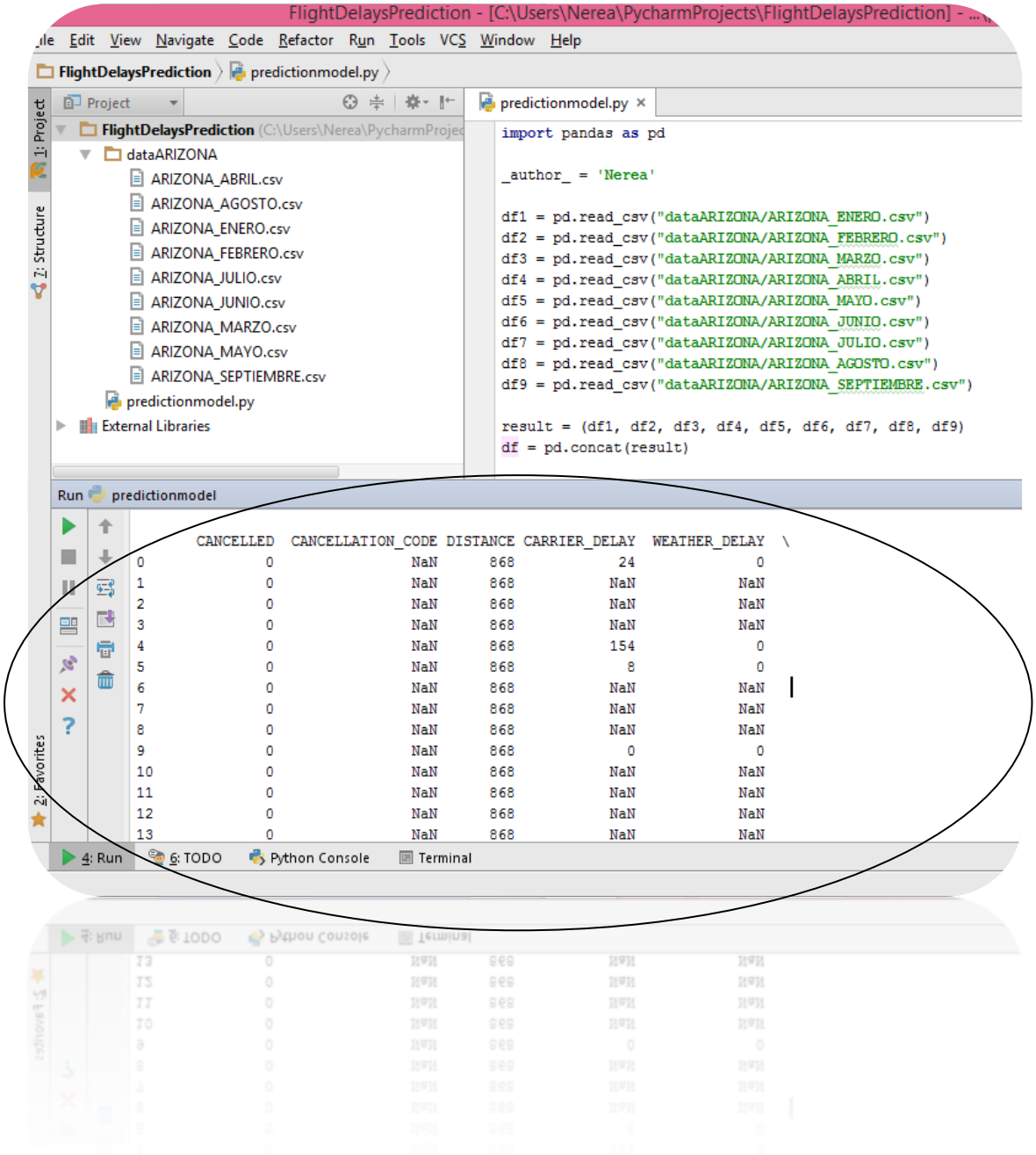
Se han mostrado por pantalla los 38 campos iguales(columnas) que contenían todos los archivos, y únicamente 5 primeras filas, con su información pertinente, tal y como se le había pedido en el código.

3.2.2 Pre procesamiento de los datos

El **"Pre procesamiento de Datos" / "La Preparación de Datos"** engloba a todas aquellas técnicas de análisis de datos que permite mejorar la calidad de un conjunto de datos de modo que las técnicas de extracción de conocimiento/minería de datos puedan obtener mayor y mejor información (mejor porcentaje de clasificación, reglas con más completitud, etc.). Es por esto, que este bloque siguiente presenta los apartados de limpieza de datos, análisis visual, transformación de los datos y selección de características.

3.2.2.1 Limpieza de los datos no significativos

En esta segunda fase se procede a eliminar los datos que no aportan ningún tipo de información para el análisis de predicción. Más concretamente se eliminan los campos relacionados con el tiempo que no contienen ningún tipo de información, es decir, campos vacios debido a vuelos cancelados. Estos campos se muestran, en la pantalla de resultados de Pycharm, con el símbolo **Na**. Se puede ver una muestra de ello en la siguiente imagen:



Para empezar se eliminarán los vuelos cancelados ya que no son de utilidad para analizar y predecir los retrasos en los vuelos.

Para ello, referente a los campos de las cancelaciones(*Cancelled* y *Cancellation_Code*), se recuerda que un vuelo está cancelado cuando en su campo indica un **1**, y un **0** cuando el vuelo no ha sido cancelado. Respectivamente si un vuelo no ha sido cancelado, el campo de *Cancellation_Code* estará vacío, es decir no contendrá ningún tipo de información**(Na)**, y si hay retraso, contendrá un código informativo, con lo cual, el campo no será nulo.

Así, para eliminar los vuelos cancelados, primero, se ha procedido a asignar como **nulo** a todas las filas del campo "*Cancellation_Code":*

```
df = df[df.CANCELLATION_CODE.isnull()]
```

Seguidamente se observarán tanto los campos que contienen valores nulos como los que contienen algún tipo de información con el siguiente comando:

```
print df.isnull().any()
```

En la siguiente imagen se pueden ver los resultados obtenidos en Pycharm:

```
C:\Python27\python.exe C:/Users/Nerea/PycharmProjects/FlightDelaysPrediction/predictionmodel.py
YEAR                    False
QUARTER                 False
MONTH                   False
DAY_OF_MONTH            False
DAY_OF_WEEK             False
FL_DATE                 False
UNIQUE_CARRIER          False
AIRLINE_ID              False
CARRIER                 False
FL_NUM                  False
ORIGIN_AIRPORT_ID       False
ORIGIN_AIRPORT_SEQ_ID   False
ORIGIN_CITY_MARKET_ID   False
ORIGIN                  False
ORIGIN_CITY_NAME        False
DEST_AIRPORT_ID         False
DEST_AIRPORT_SEQ_ID     False
DEST_CITY_MARKET_ID     False
DEST                    False
DEST_CITY_NAME          False
DEP_TIME                False
DEP_DELAY_NEW           False
DEP_DEL15               False
TAXI_OUT                False
WHEELS_OFF              False
WHEELS_ON               True
TAXI_IN                 True
ARR_TIME                True
ARR_DELAY_NEW           True
ARR_DEL15               True
CANCELLED               False
CANCELLATION_CODE       True
DISTANCE                False
CARRIER_DELAY           True
WEATHER_DELAY           True
NAS_DELAY               True
SECURITY_DELAY          True
LATE_AIRCRAFT_DELAY     True
Unnamed: 38             True
dtype: bool
Process finished with exit code 0
```

La información que se muestra en la anterior imagen informa con un ***False*** de que no hay ninguna fila, del campo en cuestión, que sea **nula** ,es decir, que no contenga ningún tipo de información. Respectivamente los campos que contienen ***True***, nos informan que hay filas nulas(Na) , las cuales, no contienen ningún tipo de información.

Además de esto se observa también, que se ha resaltado en rojo unos campos que por ahora contienen información nula, entre otros, en alguna de sus filas, el objetivo en este punto es hacer que contengan algún tipo de información ya que se requieren para poder analizar los retrasos en los vuelos.

Para poder mantener estos campos con algún tipo de información, se procede a cambiar los valores nulos(Na) de estas columnas por valores con el numero 0. Este proceso se formaliza con el código en Pycharm siguiente:

```
delay_column_names = ["ARR_DELAY_NEW", "ARR_DEL15", "CARRIER_DELAY",
"WEATHER_DELAY", "NAS_DELAY", "SECURITY_DELAY", "LATE_AIRCRAFT_DELAY"]

df[delay_column_names] = df[delay_column_names].fillna(0)
```

Para comprobar que se han realizado los cambios pertinentes a continuación, se muestran por pantalla los nuevos resultados extraídos de Pycharm:

```
C:\Python27\python.exe C:/Users/Nerea/PycharmProjects/FlightDelaysPrediction/predictionmodel.py
YEAR                    False
QUARTER                 False
MONTH                   False
DAY_OF_MONTH            False
DAY_OF_WEEK             False
FL_DATE                 False
UNIQUE_CARRIER          False
AIRLINE_ID              False
CARRIER                 False
FL_NUM                  False
ORIGIN_AIRPORT_ID       False
ORIGIN_AIRPORT_SEQ_ID   False
ORIGIN_CITY_MARKET_ID   False
ORIGIN                  False
ORIGIN_CITY_NAME        False
DEST_AIRPORT_ID         False
DEST_AIRPORT_SEQ_ID     False
DEST_CITY_MARKET_ID     False
DEST                    False
DEST_CITY_NAME          False
DEP_TIME                False
DEP_DELAY_NEW           False
DEP_DEL15               False
TAXI_OUT                False
WHEELS_OFF              False
WHEELS_ON               True
TAXI_IN                 True
ARR_TIME                True
ARR_DELAY_NEW           False
ARR_DEL15               False
CANCELLED               False
CANCELLATION_CODE       True
DISTANCE                False
CARRIER_DELAY           False
WEATHER_DELAY           False
NAS_DELAY               False
SECURITY_DELAY          False
LATE_AIRCRAFT_DELAY     False
Unnamed: 38             True
dtype: bool
Process finished with exit code 0
```

Como se puede observar en la anterior imagen, los cambios se han resaltado en color verde. Los campos pertinentes han pasado de contener valores nulos*(True)* a contener valores con información*(False)*, en este caso con valor 0.

Seguidamente se procede a sustituir los valores que contiene el campo "CARRIER_DELAY" por los valores del campo "ARR_DELAY_NEW" solo en las filas, donde la suma de las columnas de los campos con retraso den 0 y el valor de "ARR_DELAY_NEW" sea mayor que 0. Utilizamos el campo de "ARR_DELAY_NEW" ya que es el campo que registra **el total** de los retrasos obtenidos en un vuelo en cuestión.

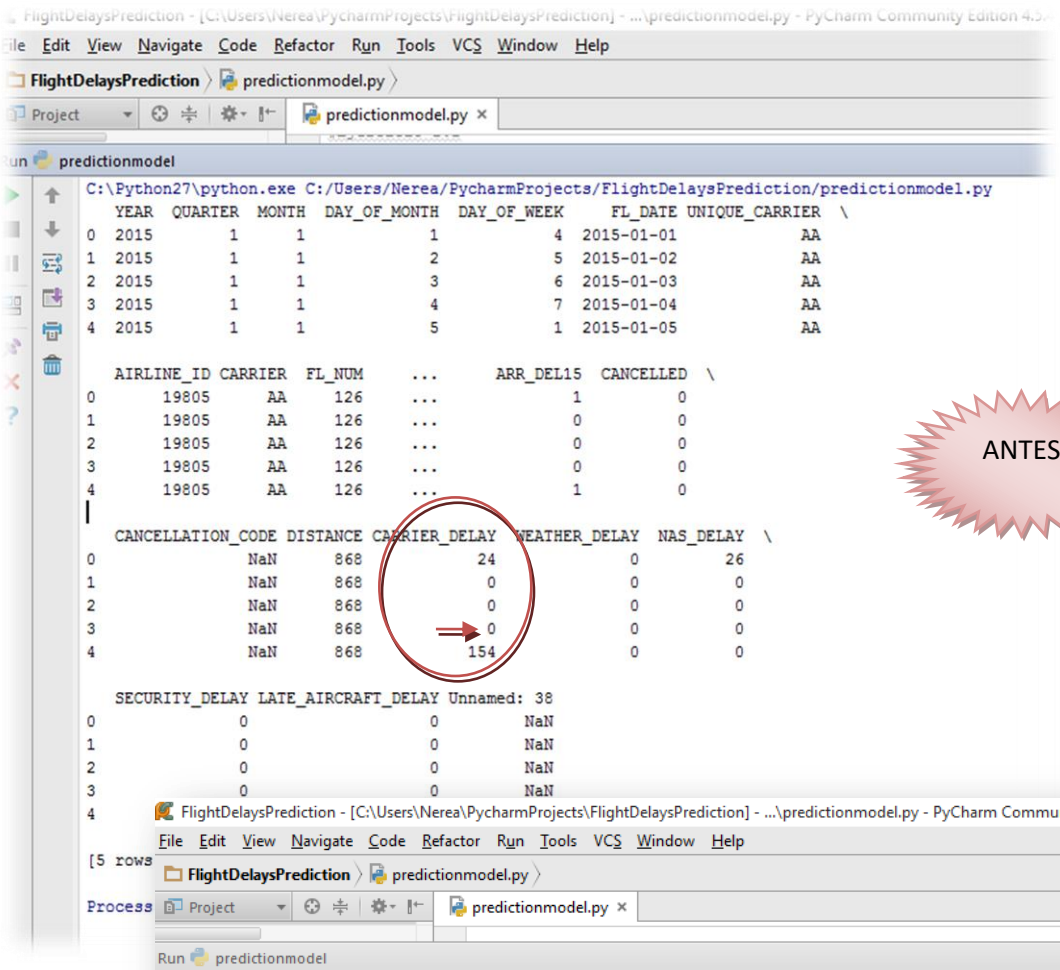
Si no se realizara esto lo que sucedería es que, los campos de las causas de los retrasos, no contendrían toda la información exacta de los retrasos, omitiendo información útil de si los vuelos han llegado a su destino a la hora establecida o han llegado con algún retraso. Esto es debido a que , si se recuerda, en la base de datos a utilizar se explicó que, solo habría información de las causas de los retrasos si se diesen retrasos tanto en el aeropuerto de salida como en el aeropuerto de destino.

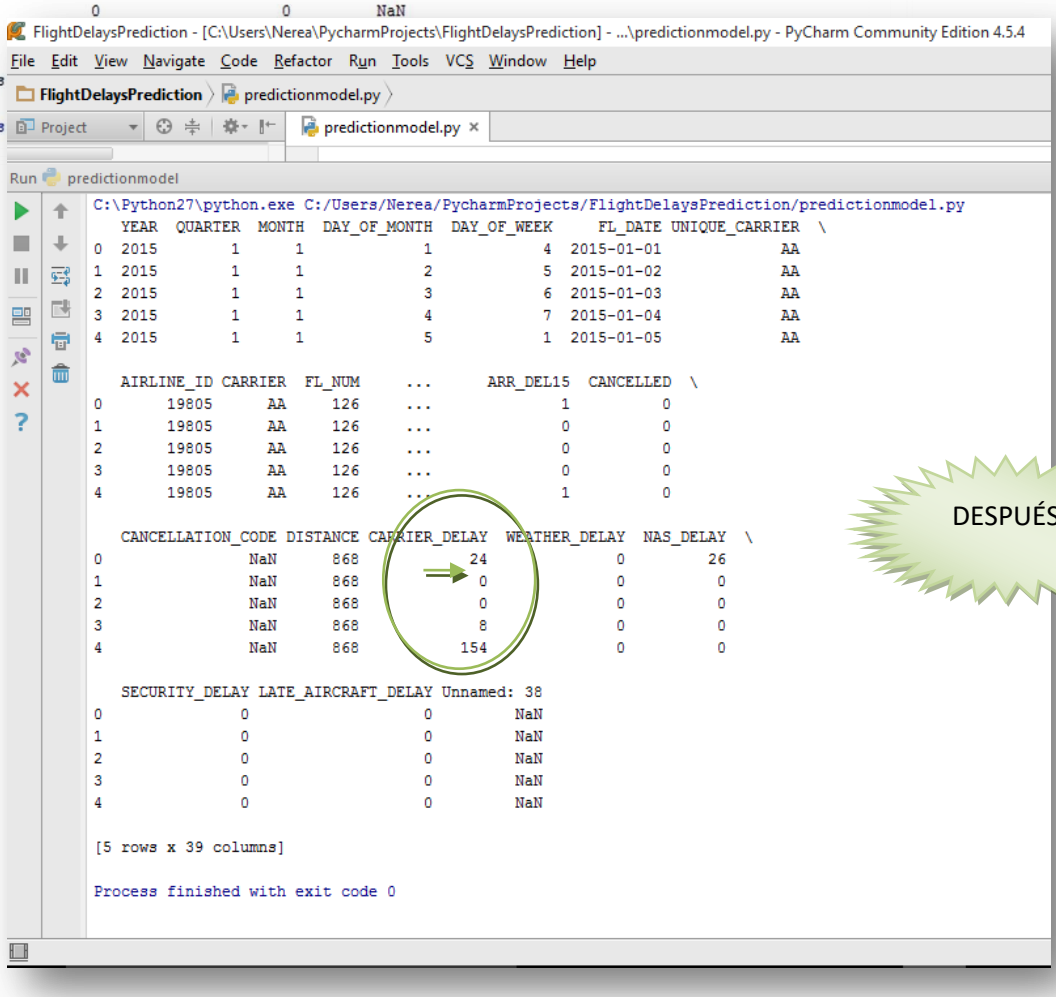
Lo que se consigue con esto es poder saber correctamente la información final de si el vuelo realizado ha llegado a su destino con algún tipo de retraso o ha llegado sin retraso y ha cumplido con su hora establecida de llegada, que a la fin y al cabo, esto último, es lo que se pretende cuando se realiza cualquier servicio de transporte.

El código utilizado en Pycharm es el siguiente:

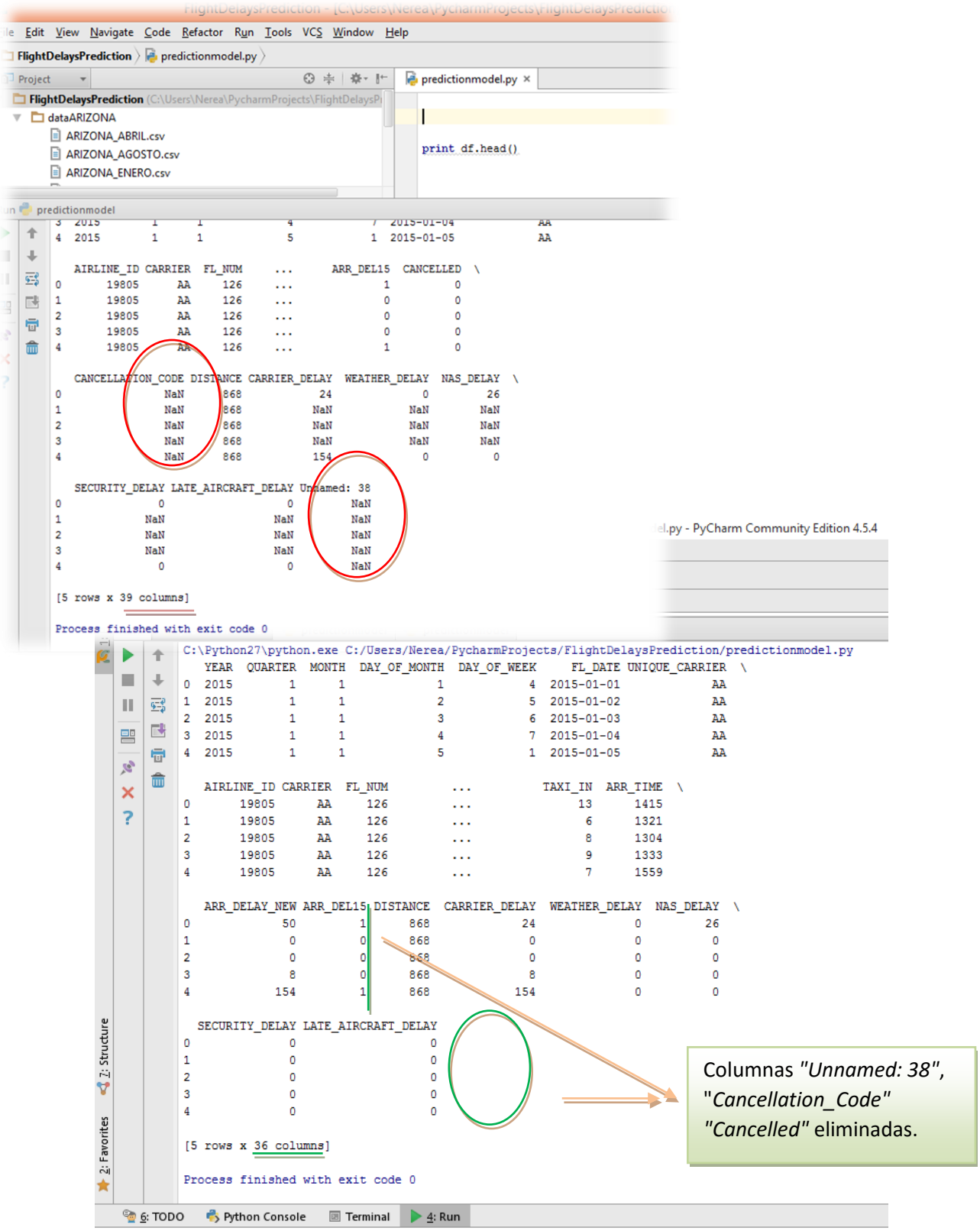
```
selected_delay_column_names = ["CARRIER_DELAY", "WEATHER_DELAY", "NAS_DELAY",
"SECURITY_DELAY", "LATE_AIRCRAFT_DELAY"]

mask = (df['ARR_DELAY_NEW'] > 0) & (df[selected_delay_column_names].sum(axis=1) == 0)
df.ix[mask, 'CARRIER_DELAY'] = df.ix[mask, 'ARR_DELAY_NEW']
```

Los resultados se pueden observar en la siguiente página. Veremos cómo los valores del campo de "CARRIER_DELAY" cambian de antes a después de realizar la acción del código anterior. Se ha imprimido sólo las 5 primeras filas, ya que en ellas ya se puede ver el cambio, donde en el mismo vuelo anterior no se veía el retraso total, ahora se ve un retraso total de **8 minutos.**


FlightDelaysPrediction - [C:\Users\Nerea\PycharmProjects\FlightDelaysPrediction] - ...\predictionmodel.py - PyCharm Community Edition 4.5.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
FlightDelaysPrediction predictionmodel.py
Project
predictionmodel.py
predictionmodel
C:\Python27\python.exe C:/Users/Nerea/PycharmProjects/FlightDelaysPrediction/predictionmodel.py
YEAR QUARTER MONTH DAY_OF_MONTH DAY_OF_WEEK FL_DATE UNIQUE_CARRIER
0 2015 1 1 1 4 2015-01-01 AA
1 2015 1 1 2 5 2015-01-02 AA
2 2015 1 1 3 6 2015-01-03 AA
3 2015 1 1 4 7 2015-01-04 AA
4 2015 1 1 5 1 2015-01-05 AA
AIRLINE_ID CARRIER FL_NUM ... ARR_DEL15 CANCELLED
0 19805 AA 126 ... 1 0
1 19805 AA 126 ... 0 0
2 19805 AA 126 ... 0 0
3 19805 AA 126 ... 0 0
4 19805 AA 126 ... 1 0
CANCELLATION_CODE DISTANCE CARRIER_DELAY WEATHER_DELAY NAS_DELAY
0 NaN 868 24 0 26
1 NaN 868 0 0 0
2 NaN 868 0 0 0
3 NaN 868 0 0 0
4 NaN 868 154 0 0
SECURITY_DELAY LATE_AIRCRAFT_DELAY Unnamed: 38
0 0 0 NaN
1 0 0 NaN
2 0 0 NaN
3 0 0 NaN
4
[5 rows
Process
ANTES


FlightDelaysPrediction - [C:\Users\Nerea\PycharmProjects\FlightDelaysPrediction] - ...\predictionmodel.py - PyCharm Community Edition 4.5.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
FlightDelaysPrediction predictionmodel.py
Project
predictionmodel.py
Run predictionmodel
C:\Python27\python.exe C:/Users/Nerea/PycharmProjects/FlightDelaysPrediction/predictionmodel.py
YEAR QUARTER MONTH DAY_OF_MONTH DAY_OF_WEEK FL_DATE UNIQUE_CARRIER
0 2015 1 1 1 4 2015-01-01 AA
1 2015 1 1 2 5 2015-01-02 AA
2 2015 1 1 3 6 2015-01-03 AA
3 2015 1 1 4 7 2015-01-04 AA
4 2015 1 1 5 1 2015-01-05 AA
AIRLINE_ID CARRIER FL_NUM ... ARR_DEL15 CANCELLED
0 19805 AA 126 ... 1 0
1 19805 AA 126 ... 0 0
2 19805 AA 126 ... 0 0
3 19805 AA 126 ... 0 0
4 19805 AA 126 ... 1 0
CANCELLATION_CODE DISTANCE CARRIER_DELAY WEATHER_DELAY NAS_DELAY
0 NaN 868 24 0 26
1 NaN 868 0 0 0
2 NaN 868 0 0 0
3 NaN 868 8 0 0
4 NaN 868 154 0 0
SECURITY_DELAY LATE_AIRCRAFT_DELAY Unnamed: 38
0 0 0 NaN
1 0 0 NaN
2 0 0 NaN
3 0 0 NaN
4 0 0 NaN
[5 rows x 39 columns]
Process finished with exit code 0
DESPUÉS

También, se procede a hacer una mejora en los resultados por pantalla que nos ofrece Pycharm eliminando así, la columna **"*Unnamed: 38"*** y la columna **"*Cancellation_Code"*** ya que no contiene ningún tipo de información(Na). Además también eliminaremos la columna ***"Cancelled***" ya que no será de utilidad para este estudio. El procedimiento y los resultados se ven en la siguiente imagen:



Seguidamente con el comando **" df = df.dropna() "** se eliminaran todas las filas que aún sigan sin ningún tipo de información (Na). El resultado se muestra en la siguiente ilustración:

```
C:\Python27\python.exe C:/Users/Nerea/PycharmProjects/FlightDelaysPrediction/predictionmodel.py
YEAR                 False
QUARTER              False
MONTH                False
DAY_OF_MONTH         False
DAY_OF_WEEK          False
FL_DATE              False
UNIQUE_CARRIER       False
AIRLINE_ID           False
CARRIER              False
FL_NUM               False
ORIGIN_AIRPORT_ID    False
ORIGIN_AIRPORT_SEQ_ID False
ORIGIN_CITY_MARKET_ID False
ORIGIN               False
ORIGIN_CITY_NAME     False
DEST_AIRPORT_ID      False
DEST_AIRPORT_SEQ_ID  False
DEST_CITY_MARKET_ID  False
DEST                 False
DEST_CITY_NAME       False
DEP_TIME             False
DEP_DELAY_NEW        False
DEP_DEL15            False
TAXI_OUT             False
WHEELS_OFF           False
WHEELS_ON            False
TAXI_IN              False
ARR_TIME             False
ARR_DELAY_NEW        False
ARR_DEL15            False
DISTANCE             False
CARRIER_DELAY        False
WEATHER_DELAY        False
NAS_DELAY            False
SECURITY_DELAY       False
LATE_AIRCRAFT_DELAY  False
dtype: bool
Process finished with exit code 0
```

Se puede observar como todos los campos contienen algún tipo de información y no son nulos(Na), lo vemos mediante la información mostrada de los campos con un ***False***, referente de que no hay ninguna fila que sea **nula,** como ya se ha descrito anteriormente.

Finalmente creamos un archivo CSV con el nombre ***"flights3"*** para guardar todos los cambios realizados en los archivos de los vuelos contenidos en el programa Pycharm.

3.2.2.2 Análisis visual de los datos

Este siguiente fase consiste principalmente en la visualización mediante gráficos de todos los datos de los que se disponen, para conseguir así un mejor análisis representativo de ellos.

Para poder hacer los gráficos en Pycharm, primero, se ha tenido que instalar otra librería más en el programa, esta librería se llama **"seaborn"**. Si se hace memoria, de momento, ya se disponen de las siguientes librerías instaladas:

- numpy
- scipy
- pandas
- **matplotlib**
- **seaborn**

→ Se utilizan, entre otras funciones, para la creación de gráficos.

Seguidamente, antes de crear el código en Pycharm, se han de analizar muy bien las variables de las que se disponen, puesto que hay muchas y se ha de ser lo más preciso posible para crear un análisis que aporte significado y valor al estudio del proyecto. Dicho esto se va a hacer un repaso a continuación del tipo y contenido de datos de los que se dispone:

- PERIODO DE TIEMPO
  - ***Year*: número**
  - ***Quarter:* número**
  - ***Month*: número**
  - ***Day_of_month*: número**
  - ***Day_of_week*: número**
  - ***Fl_date*: número**
- AEROLÍNEA
  - ***Unique_Carrier*: carácter**
  - ***Airline ID*: número**
  - ***Carrier*: carácter**
  - ***Fl_Num*: número**
- ORIGEN
  - ***Origin_Airport_ID*: número**
  - ***Origin_Airport_Seq_ID*: número**
  - ***Origin_City_Marquet_ID*: número**
  - ***Origin*: carácter**
  - ***Origin_City_Name*: carácter**
- DESTINACIÓN
  - ***Dest_Airport_ID*: número**
  - ***Dest_Airport_Seq_ID*: número**
  - ***Dest_City_Market_ID:* número**
  - ***Dest:* carácter**
  - ***Dest_City_Name* carácter**
- INFORMACIÓN/PLAN DE SALIDA
  - ***Dep_Time:* número**
  - ***Dep_Delay_New:* número**
  - ***Dep_Del15:* número**
  - ***Taxi_Out:* número**
  - ***Wheels_Off*: número**
- INFORMACIÓN/PLAN DE LLEGADA
  - ***Wheels_On*: número**
  - ***Taxi_in*: número**
  - ***Arr_Time*: número**
  - ***Arr_Delay_New*: número**
  - ***Arr_Delay15*: número**
- SUMARIO DE VUELO
  - ***Distance*: número**
- CAUSAS DE LOS RETRASOS
  - ***Carrier_Delay*: número**
  - ***Weather_Delay*: número**
  - ***NAS_Delay*: número**
  - ***Security_Delay*: número**
  - ***Late_Aircraft_Delay:* número**

BINARIO(0 ó 1)
HORA(hhmm)
TIEMPO(minutos)
KM

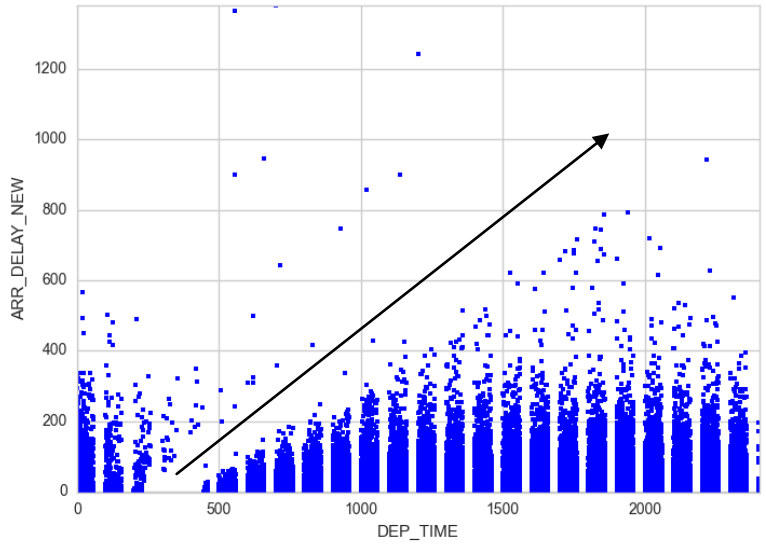
Una vez analizados los datos de los que se disponen, se prosigue a la realización de los gráficos mediante código *Python* en *Pycharm*.
Se han realizado distintos tipos de gráficos, de los cuales, su código y la visualización gráfica de cada uno de ellos se explica a continuación:

- **Gráfico de dispersión ("scatterplot"):**
  Es la representación gráfica más útil para describir el comportamiento conjunto de dos variables, donde cada caso aparece representado como un punto en el plano definido por las variables x, y.
  A la hora de la realización de este tipo de gráfico, se ha llevado a cabo con la realización de la pregunta siguiente:
  *"Hay algún tipo de relación lineal entre **la hora de salida de un vuelo** y la creación de los **retrasos en el aeropuerto de destino**?"*
  Esta pregunta se ha resuelto mediante la creación del siguiente código en Pycharm:

```
def scatterplot(x, y, x_title, y_title):
    plt.plot(x, y, 'b.')
    plt.xlabel(x_title)
    plt.ylabel(y_title)
    plt.xlim(min(x)-1, max(x)+1)
    plt.ylim(min(y)-1, max(y)+1)
    plt.show()

scatterplot(df.DEP_TIME, df.ARR_DELAY_NEW, "DEP_TIME",
"ARR_DELAY_NEW")
print scatterplot
```

Como se puede observar se han utilizado las variables **Dep_Time (hora de salida)** y **Arr_Delay_New(retraso en las llegadas)** para el estudio conjunto de las dos. Los resultados se pueden ver en el siguiente gráfico de dispersión:

Ilustración 18.Gráfica de Relación retrasos en las llegadas vs. hora de salida



Fuente: Elaboración propia

Como se observa hay una **relación lineal creciente** entre la hora de salida de un vuelo y el tiempo de retraso en minutos de este. Esto quiere decir que hay más retrasos conforme van pasando las horas del día por el efecto que causan los retrasos de **propagación de los vuelos.**

- **Histograma ("histplot"):**

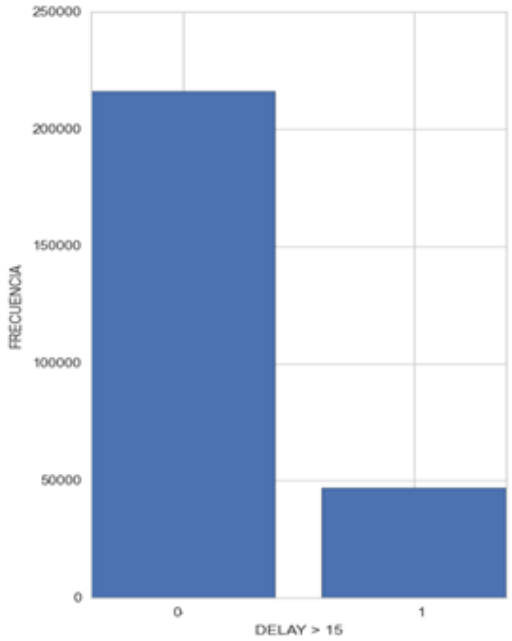
Es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados, ya sea en forma diferencial o acumulada.

El código realizado en Pycharm para la creación de histogramas es el siguiente:

```
def barplot(labels, data, x_title, y_title):
    pos = arange(len(data))
    plt.xlabel(x_title)
    plt.ylabel(y_title)
    plt.xticks(pos+0.4, labels)
    plt.bar(pos, data)
    plt.show()

def histplot(data,x_title, y_title,bins= None,nbins= 10):
    minx, maxx = min(data), max(data)
    space = (maxx-minx)/float(nbins)
    if not bins:
        bins = arange(minx, maxx, space)
    binned = [bisect.bisect(bins, x) for x in data]
    l = ['%i' % x for x in list(bins)+[maxx]]\
        if space < 1 \
        else [str(int(x))
              for x in list(bins)+[maxx]]
    displab = [x+'-'+y for x, y in zip(l[:-1], l[1:])]

    barplot(displab, [binned.count(x+1)for x in
range(len(bins))], x_title, y_title)

histplot(df.DEP_TIME, 'DEPARTURE TIME', 'FRECUENCIA')
print histplot
```

En el siguiente histograma se muestra la frecuencia en relación con el rendimiento de los vuelos realizados en el tiempo programado y los que poseen retrasos de más de 15 minutos.

Ilustración 19. Rendimiento para los vuelos de enero 2015 - septiembre 2015



0= Vuelos realizados en el tiempo planificado o con menos de 15 minutos de retraso

1= Vuelos con retrasos de más de 15 minutos

Fuente: Elaboración propia

- **<u>Gráficos de barras ("barchart"):</u>**

Son una forma de representar y comparar gráficamente un conjunto de datos o valores, y está conformado por barras rectangulares de longitudes proporcionales a los valores representados.

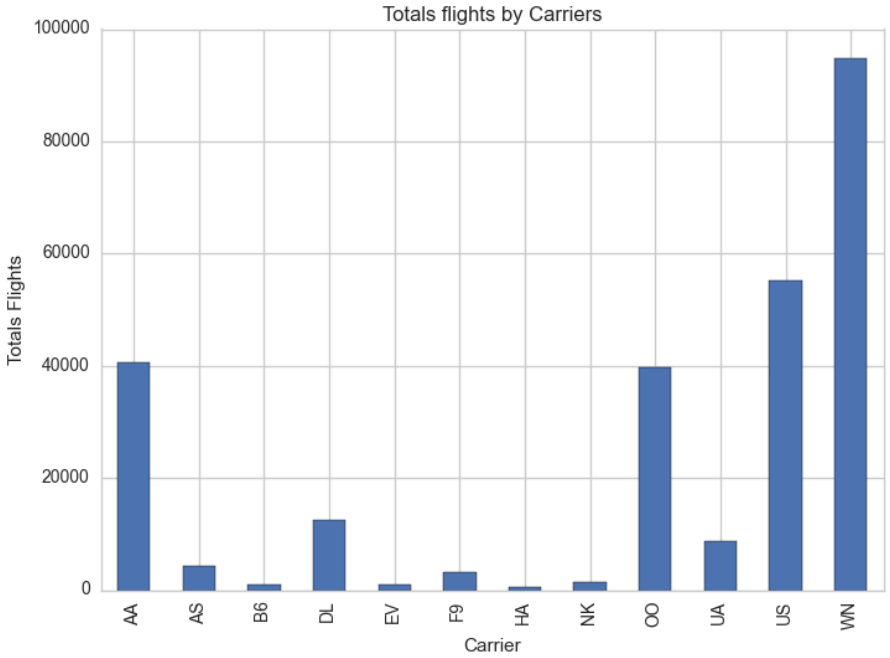
El código realizado en Pycharm para la creación de los gráficos de barras es el siguiente:

```
# funcion para datos del tipo string(caracter)
def barchart(x, y, x_title, y_title, numbins=10):
    data = pd.DataFrame()
    data[x_title] = df['CARRIER']
    data[y_title] = df['ARR_DELAY_NEW']
    carrier_group = data.groupby('CARRIER')
    delays_totals = carrier_group.mean()  # here you
may use sum() instead of mean(), when it's
appropriate
    delays_totals.sort(columns='ARR_DELAY_NEW')
    ax = delays_totals.plot(kind='bar',
title="Arrivals Delays Carrier", legend=False)
    ax.set_xlabel("Carrier")
    ax.set_ylabel("Average Arrivals Delays")
    plt.show()

barchart(df.CARRIER.values, df.ARR_DELAY_NEW.values,
"CARRIER", "ARR_DELAY_NEW", len(df.CARRIER.unique()))
print barchart
```

Con este tipo de gráficos se ha querido representar más de un escenario. Estos escenarios se explican a continuación.

En un primer escenario, se muestra el gráfico con la frecuencia de vuelos realizados por las compañías aéreas en el periodo de enero 2015 - septiembre 2015:

Ilustración 20. Gráfico de Total de los vuelos por compañía (enero 2015-sept. 2015)

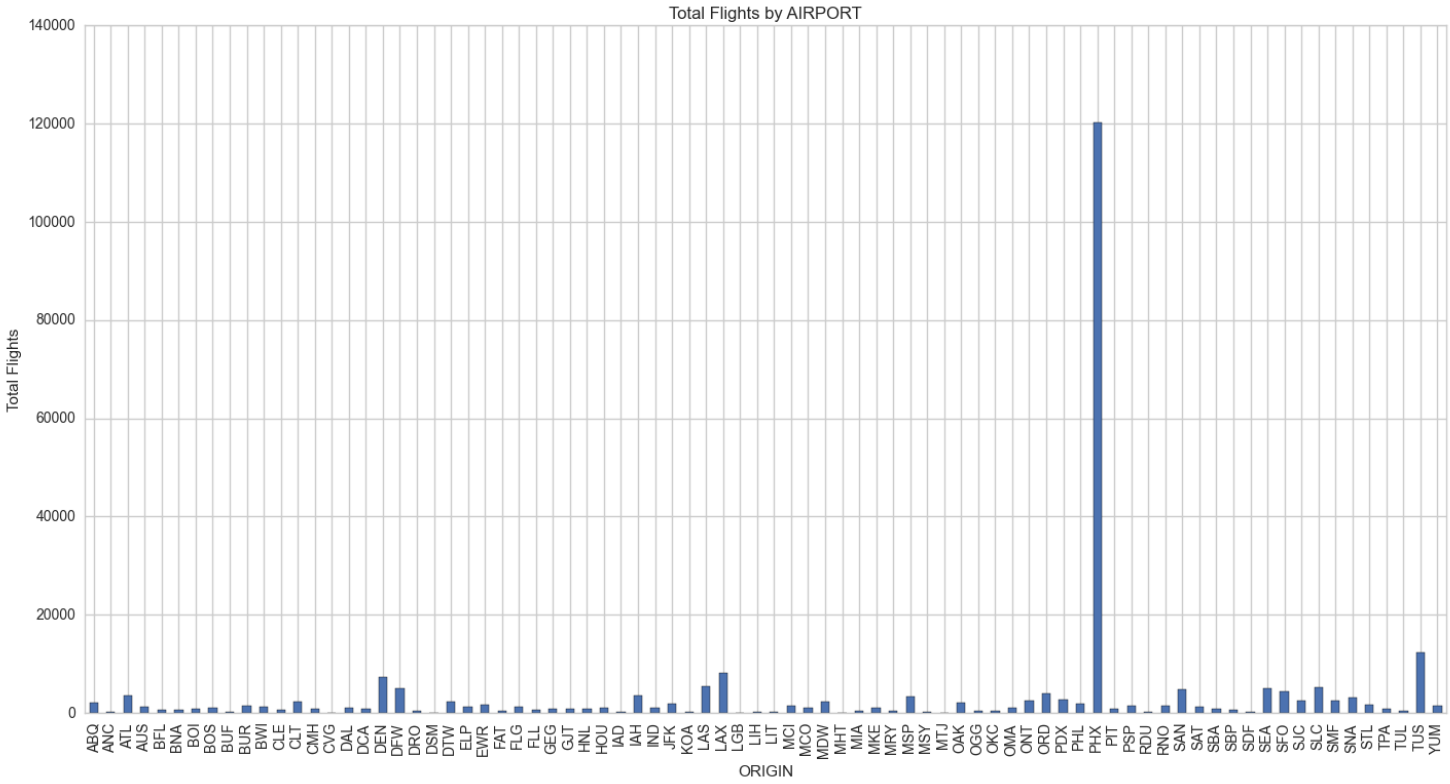


Fuente: Elaboración propia

Se puede observar como la compañía aérea *Southwest Airlines (WN)* es la que realiza un mayor número de vuelos para el periodo de estudio de enero 2015 - septiembre 2015, con cerca de 100.000 vuelos realizados. A ella le sigue *US Airways(US)* con alrededor de 60000 vuelos realizados y *SkyWest Airlines (OO)* y *American Airlines (AA)* con 40.000 vuelos realizados.

- En el siguiente gráfico se muestra el total de vuelos realizados en cada uno de los aeropuertos de la base de datos. Se observa claramente que el mayor tránsito de vuelos se realiza en el Aeropuerto Internacional de Phoenix-Sky Harbor(phx), con 120.000 vuelos realizados para el periodo de enero a septiembre de 2015.

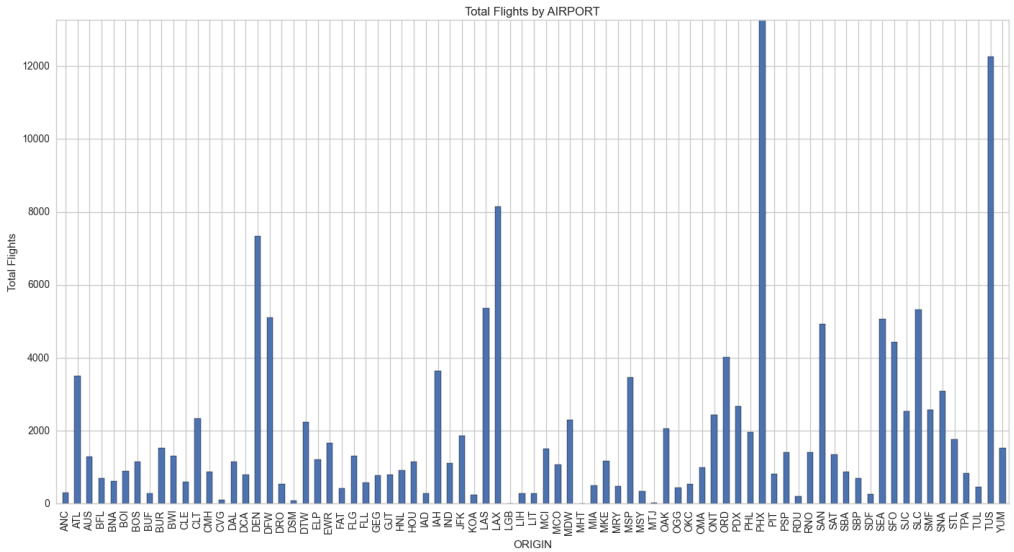
Ilustración 21. Gráfico de Vuelos totales por aeropuerto(enero 2015-sept. 2015)



Fuente: Elaboración propia

Si reducimos la escala del total de vuelos podemos observar cómo la mayoría de los aeropuertos no pasan de 4000 vuelos realizados desde enero 2015 - septiembre 2015. Esto se muestra en el siguiente gráfico:

Ilustración 22. Gráfica de Total de vuelos por aeropuerto (escala reducida)



Otro dato a destacar es el del aeropuerto de Long Beach, CA (LGB) que en esta base de datos únicamente se analiza un vuelo realizado con destino al aeropuerto de Phoenix(PHX).

Para el siguiente escenario se ha querido representar **la media de los retrasos tanto en las salidas como en las llegadas por compañía aérea** para así comparar qué compañías aéreas poseen más retrasos en sus vuelos realizados a lo largo del periodo estudiado en este proyecto (enero 2015 - septiembre 2015). El resultado ha sido el siguiente:

Ilustración 23. Gráfica de Media de retrasos en las salidas por compañía aérea

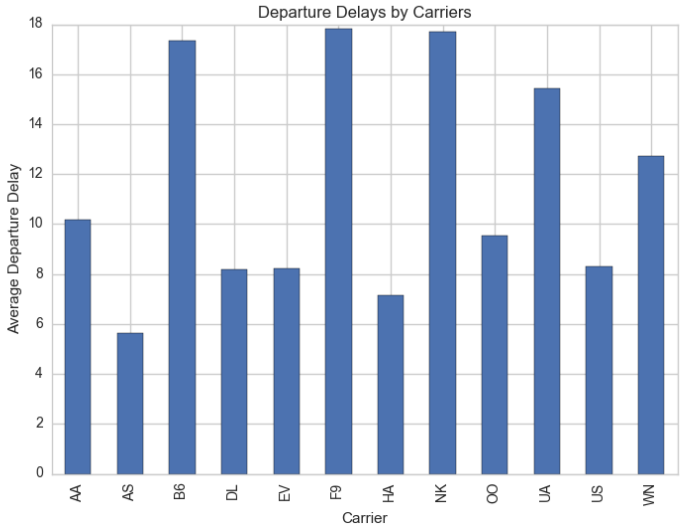
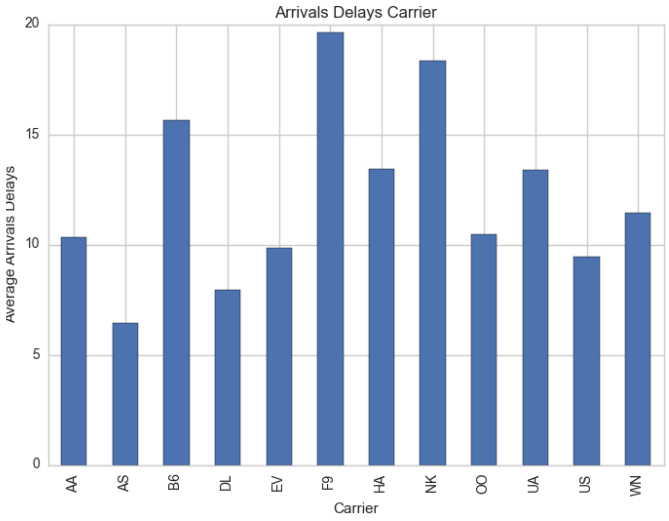


Ilustración 24. Gráfica de Media de retrasos en las llegadas por compañía aérea



Fuente: Elaboración propia

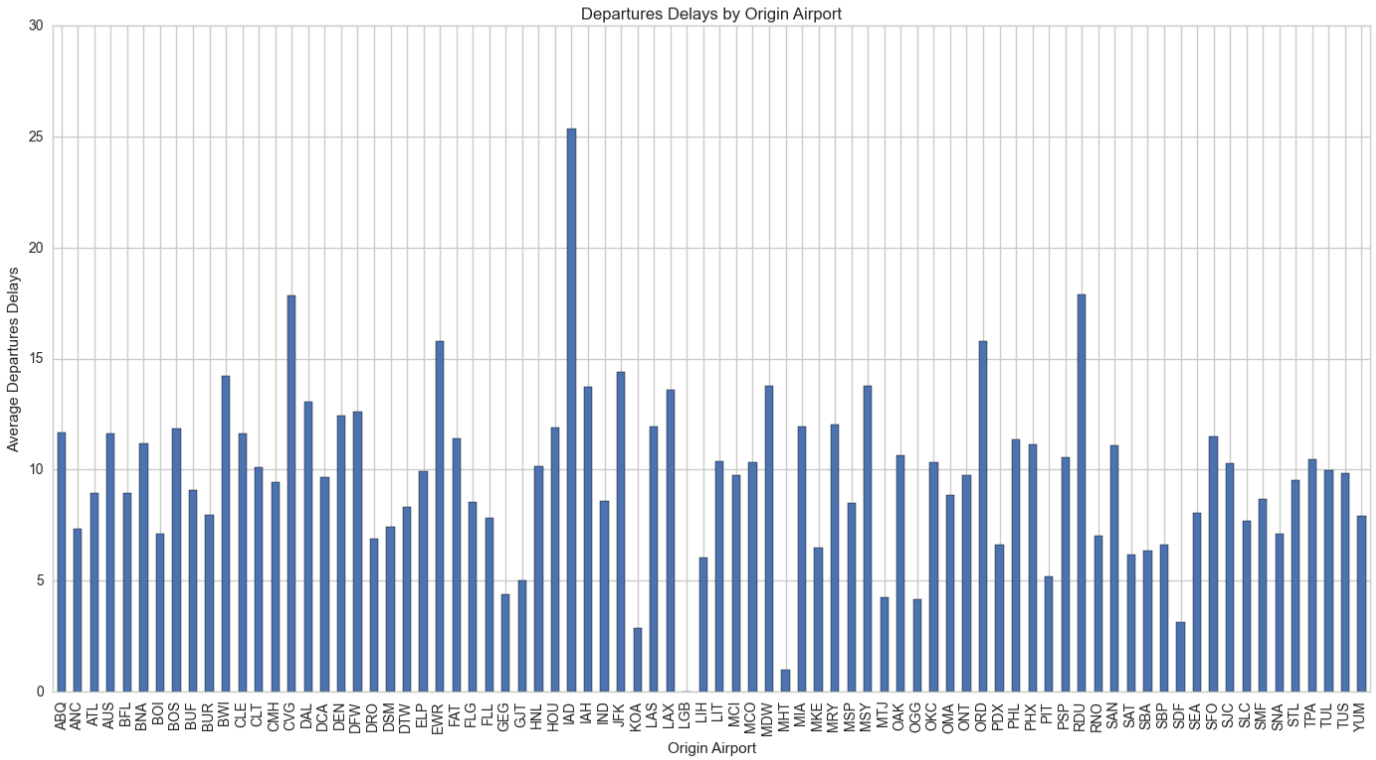
En los anteriores dos gráficos podemos ver como las compañías con mayores medias por retrasos tanto en salidas como en llegadas son:

- ***JetBlue Aiways(B6), Frontier Airlines(F9), Spirit Airlines (NK)*** con una media de alrededor de entre 15 a 20 minutos de retraso para el periodo analizado de enero 2015-septiembre2015.

En el siguiente gráfico realizado se ha querido ver cuál era la media de los retrasos en las salidas de los aeropuertos para el periodo de enero 2015-septiembre2015. El aeropuerto con la media en minutos mayor es la del aeropuerto **de Washington, DC (IAD)** con una media de alrededor de 25 minutos de retraso por vuelo. Le siguen Cincinnati, OH (CVG) y Raleigh/Durham, NC (RDU) con una media de alrededor 18 minutos de retraso.

Fuente: Elaboración propia

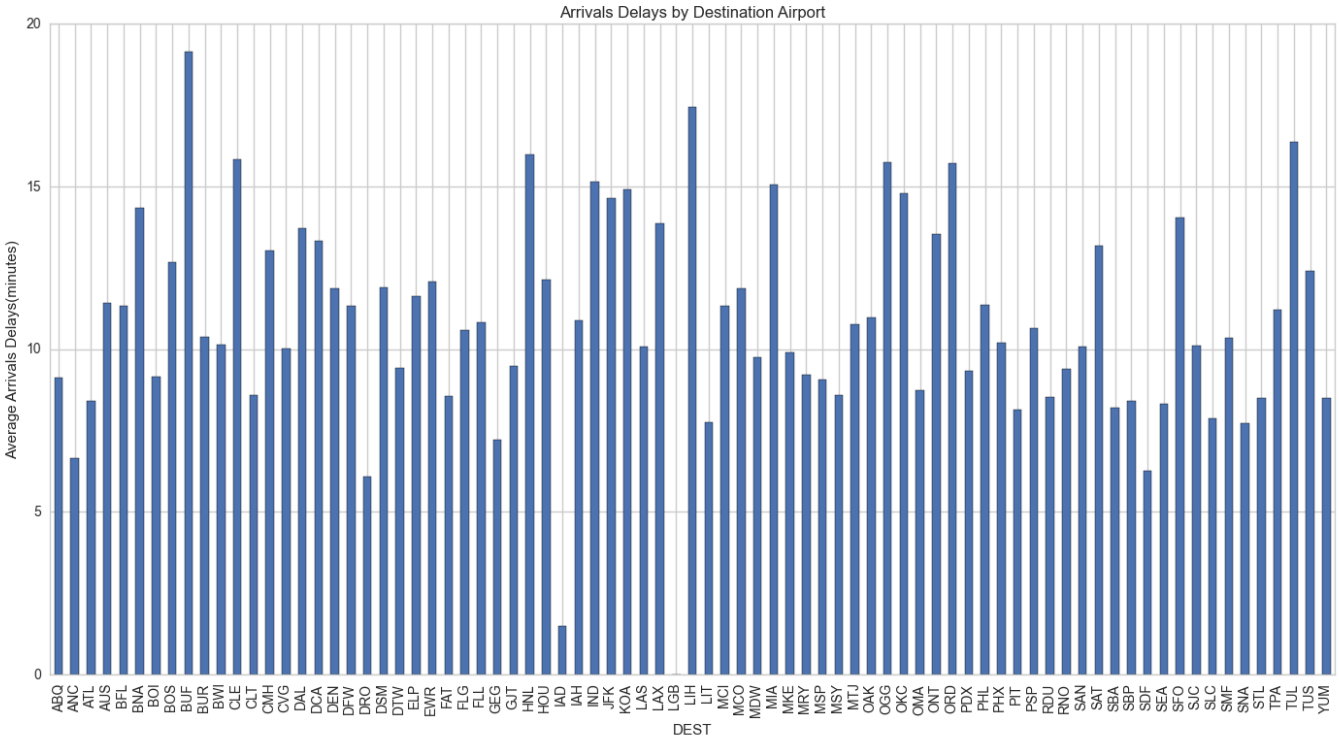
Ilustración 25.Gráfica de Media de los retrasos en las salidas del aeropuerto(en minutos)



Fuente: Elaboración propia

Para el último gráfico a analizar de a continuación , se observa la media de los retrasos en las llegadas de los aeropuertos para el periodo desde enero 2015-septiembre2015. Se puede ver que la media de los retrasos a aumentado para la mayoría de los aeropuertos respecto a la media de los retrasos en los aeropuertos de salida. En este caso el aeropuerto con mayor media en el tiempo en minutos de retraso ha sido el del aeropuerto de **Buffalo, NY (BUF)** con alrededor de una media de 20 minutos de retraso por vuelo.

Ilustración 26. Gráfica de Media de los retrasos en las llegadas del aeropuerto (en minutos)



Fuente: Elaboración propia

3.2.2.3 Transformación de los datos

3.2.2.3.1 Ingeniería de características (*feature engineering*)

Cuando el objetivo es encontrar el mejor resultado posible en un modelo de predicción, se necesita obtener el mejor rendimiento posible de las variables que uno posee y de los algoritmos que utiliza.
Ahora bien, ¿cómo se obtiene el máximo provecho de los datos utilizados para la creación del modelo predictivo?
Este es el problema que la práctica y el proceso de la **ingeniería de características (*feature engineering*)** resuelve.

Una definición para este concepto sería la siguiente:
La ingeniería de características (IC) es el proceso de transformar los datos no procesados en características que representen mejor el problema subyacente a los modelos predictivos, mejorando así la exactitud del modelo en los datos que no se ven.
De esta definición se pueden extraer otra serie de dependencias, las cuales son tareas básicas también en el proceso de ingeniería de características:

- Elección de las medidas de rendimiento (RMSE, AUC...)
- Encuadre del problema (Clasificación, Regresión...)
- Selección de los modelos de predicción que se utilizan(SVM, redes neuronales...)

Se puede decir que la IC, a *grosso modo,* es la **representación del problema.**

Por ello un buen análisis de la representación del problema y de las características de las que se posee es esencial para crear un buen modelo de predicción.
De ahí, que el éxito de todos los algoritmos de aprendizaje automático dependan de cómo se presentan los datos.

Pero, **¿qué tan importante es las ingeniería de características?**

Las características de la base de datos influyen directamente en los modelos de predicción que se utilicen y en los resultados que se puedan conseguir.

Los datos reales pueden ser impuros, pueden conducir a la extracción de patrones/reglas poco útiles. Esto se puede deber a:

- Datos Incompletos: falta de valores de atributos, …
- Datos con Ruido
- Datos inconsistentes (incluyendo discrepancias)

La preparación de datos puede generar un conjunto de datos más pequeño que el original, lo cual puede mejorar la eficiencia del proceso de Minería de Datos.
Se puede decir que, contra mejor se preparen y se elijan las características, mejores resultados se podrán conseguir. Aunque no sólo depende de la preparación de estas características, sino también, de factores en relación con el modelo que se elija, los datos de los que se dispongan, el encuadre del problema y las medidas de rendimiento que se utilizan para estimar la precisión de los resultados del modelo. Como se puede observar, los resultados del modelo de predicción dependen de muchas **propiedades interdependientes.**
Es por ello, que se necesiten de buenas características que describan las estructuras inherentes de los datos.
Las ventajas de tener buenas características se resumen en **mayor flexibilidad**, **mayor simplicidad en los modelos** y **mejores resultados,** dado que si se posee de buenas características, si se da el caso de que se elije un modelo "equivocado" (menos que óptimo), aun y así se puede obtener buenos resultados. La mayoría de los modelos pueden coger una buena estructura en datos. La flexibilidad de las buenas características y la simplicidad del modelo permitirá utilizar modelos menos complejos los cuales son más rápidos para ejecutar, más fácil de entender y más fácil de mantener. Esto es un aspecto muy deseable.

De esta manera, en los dos siguientes apartados se trabaja para la mejora de las características existentes en los datos de la muestra observando las características de los datos y haciendo transformaciones de ellas para su mejor representación del modelo. Además también se seleccionan las características más influyentes en base a la variable dependiente a predecir, que es la variable de los retrasos relacionados con las **llegadas de los vuelos a los aeropuertos.**

Transformación de la asimetría, centrado y escalado

En este apartado se trabaja con una primera selección de las características de la base de datos de Arizona, con el fin de observar sus atributos en base a sesgo, su escala y la existencia o no de valores atípicos en las características/variables de la muestra. Además también se observa la media y la desviación estándar de cada característica seleccionada.

Este paso es el pre proceso a la selección de las características más influyentes para la variable dependiente (siguiente punto).

Para empezar dividimos el proceso de transformación de características en dos partes

1) Volcado de las variables en el programa *Python*, las cuales, se utilizarán en el pre proceso/transformación (Paso 2).

- Primero se ha guardado todo el trabajo realizado en las practicas anteriores en un archivo CSV separado:

```
df.to_csv("dataARIZONA/flightsArizona_2015.csv", index=False)
```

- Seguidamente se ha creado un nuevo archivo Python llamado **"featureengineering"** y se ha llamado al archivo csv guardado en el primer paso:

```
import pandas as pd

df = pd.read_csv("dataARIZONA/flightsArizona_2015.csv")
```

- Finalmente se seleccionan las variables que se usaran en el pre proceso del escalado y centrado:

```
def build_features(features,data):
    #Firts we add numeric variables
    features.extend(['YEAR', 'QUARTER', 'MONTH',
   'DAY_OF_MONTH', 'DAY_OF_WEEK', 'FL_NUM',
   'ORIGIN_AIRPORT_ID', 'DEST_AIRPORT_ID', 'DEP_TIME',
   'DISTANCE'])
    #Secondly we add categorical variables and transform them
   into the numerical ones
    features.append('CARRIER')
    le = LabelEncoder().fit(data['CARRIER'])
    data['CARRIER'] = le.transform(data['CARRIER'])

 return data

features = []
build_features(features, df)
print df[features].head()
```

2) Transformación del conjunto de características:
Para realizar este paso se ha llevado a cabo el siguiente procedimiento:

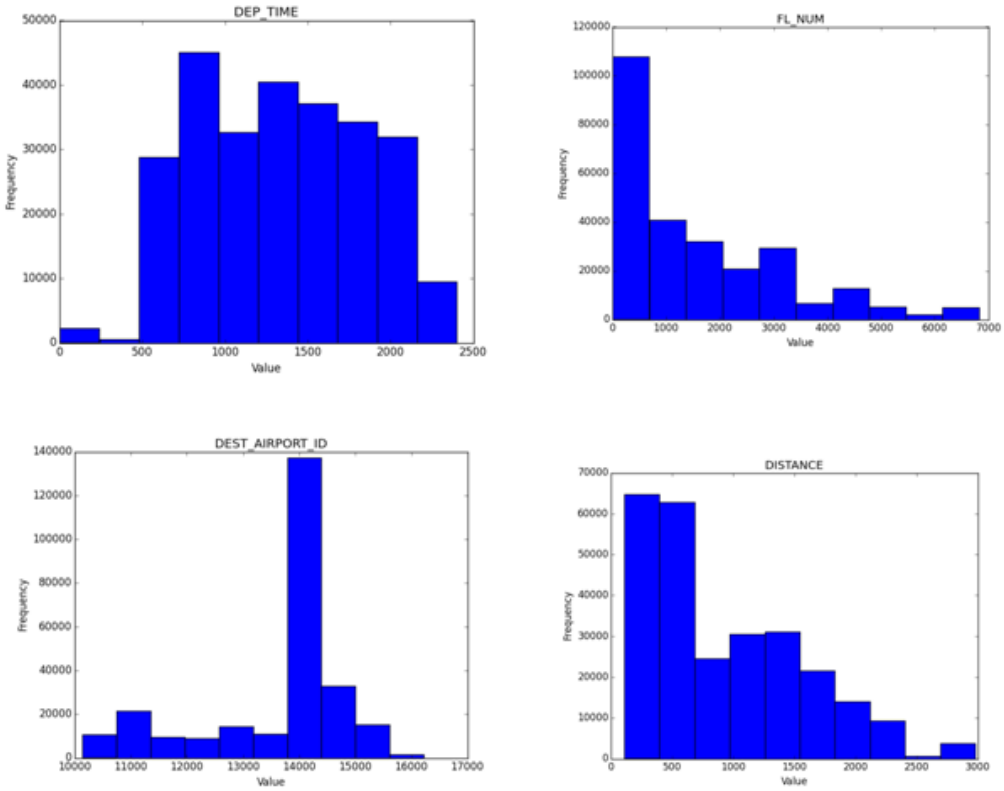
- Primero se han tenido que instalar dos librerías más para poder crear el código en Pycharm. Estas librerías son **"sklearn"** y **"xlwt"**.
- Después de instalarlas se importan al programa Pycharm:

```
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing
import matplotlib.pyplot as plt
from sklearn import ensemble
import scipy.stats as stats
import random
```

- Seguidamente se crean gráficos en relación con las variables seleccionadas en el primer punto, para investigar su **sesgado**, su **escala** y los **datos/valores atípicos** que puedan haber en cada uno de ellos. Aquí solo se mostraran algunos de ellos, para ver únicamente su forma.

Además de cada una de las variables, **aún no procesadas,** también se analiza su **media** y su **desviación estándar**.
Estos resultados se muestran en la siguiente ilustración::

Ilustración 27. Gráficos de Variables no procesadas: gráfico, medias y desviaciones

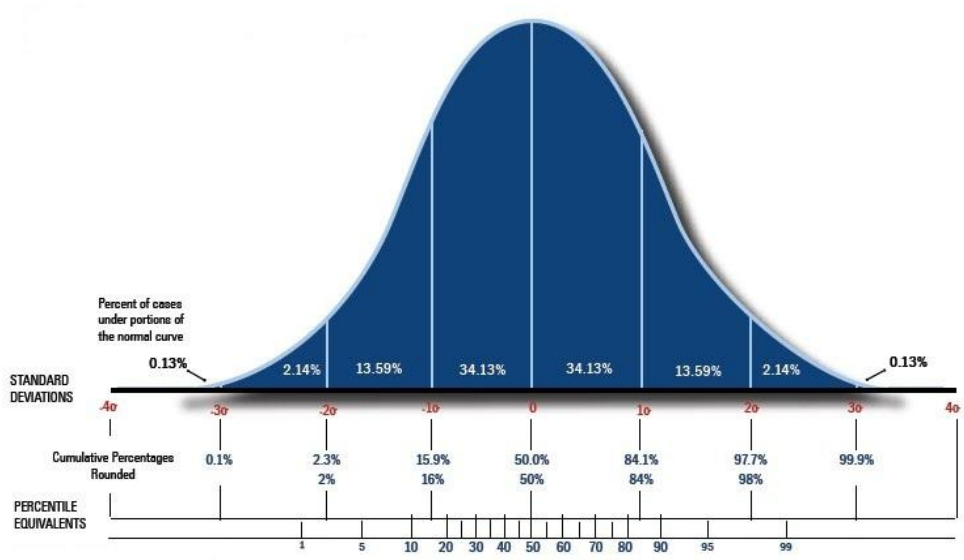


	Non-processed Data	
	Mean	Std. Deviation
YEAR	2015.0	0.0
QUARTER	1.99822277361	0.810787590649
MONTH	4.98763933341	2.53308455624
DAY_OF_MONTH	15.6782725512	8.76899176443
DAY_OF_WEEK	3.96777778201	1.99229415498
FL_NUM	1659.58065069	1476.88507961
ORIGIN_AIRPORT_ID	13628.8258699	1278.50908029
DEST_AIRPORT_ID	13628.8448485	1278.4086313
DEP_TIME	1334.20839216	500.178735327
DISTANCE	948.060912817	619.41129556
CARRIER	7.86125456199	4.0369972964

Fuente: Elaboración propia

Se puede observar como en la representación de los gráficos el sesgo de alguno de ellos está o muy desviado para la derecha o para la izquierda.
Además podemos observar que hay una gran **dispersión de las variables** seleccionadas respecto al valor promedio. Es decir hay gran **variabilidad** en las características.
Con todo esto podemos ver que existe una **asimetría.** Esta asimetría se va a corregir mediante una transformación (*skewness transformation*), con el fin de **estandarizar** el conjunto de datos de la muestra llevándolos hacia una **distribución normal.**
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.

Ilustración 28. Imagen de Distribución Normal (de Gauss)



La distribución normal tiene una asimetría cero. Pero en realidad, los valores no son nunca perfectamente simétricos y por ello la asimetría de la distribución proporciona una idea sobre si las desviaciones de la media son positivas o negativas, es decir, de si poseen valores distintos a los de la media.

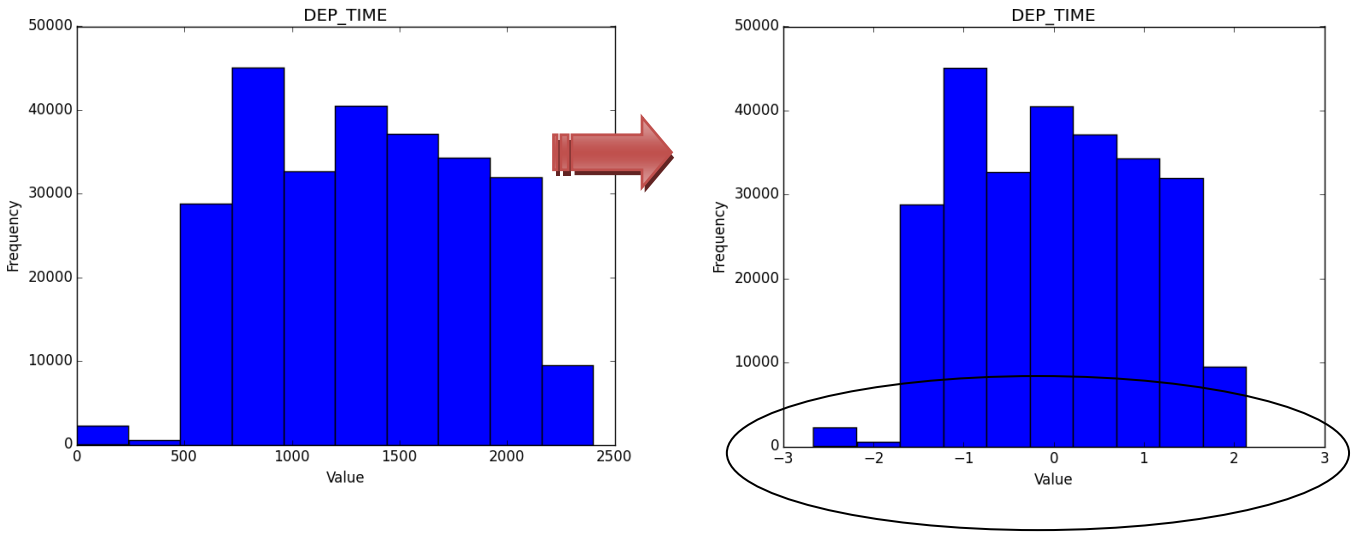
Las medidas de asimetría, sobre todo **el coeficiente de asimetría de Fisher**, se utilizan para contrastar si se puede aceptar que una distribución estadística sigue la distribución normal.

¿Cómo se realiza esta transformación?
Mediante transformaciones aplicando **un centrado** y un **escalado** de todas las características seleccionadas.

En la práctica, a menudo ignoramos la forma de la distribución y sólo se transforman los datos para centrarlos quitando el valor medio de cada función, y a continuación, se ajusta su escala dividiendo las características no constantes para su desviación estándar.
Por ejemplo, muchos de los elementos utilizados en la función objetivo de un algoritmo de aprendizaje asumen que todas las funciones están centradas en torno a cero y tienen varianza en el mismo orden. Si una característica tiene una varianza que es varios órdenes de magnitud más grandes que otros, podría dominar la función objetivo y hacer que el estimador sea incapaz de aprender de otras características correctamente como se esperaba.

El escalado es una alternativa a la estandarización haciendo que los valores de las características estén entre un mínimo y un valor máximo dado, a menudo entre cero y uno, o de modo que el valor absoluto máximo de cada característica esté a escala de tamaño unitario.
La motivación para utilizar esta transformación del escalado es que ofrece una **robustez** de muy pequeñas desviaciones estándar de las características y la **preservación de cero entradas** en los datos dispersos.

Mediante las transformaciones realizadas se pueden ver los siguientes resultados visuales y en relación numérica, con las medias y las desviaciones estándar de las características:



Fuente: Elaboración propia

Como se puede observar en los gráficos anteriores, **la transformación del escalado,** no afecta a la forma de la frecuencia de estos. Por lo tanto, la única diferencia entre el "antes" y el "después" de la transformación esta únicamente en los valores de la variable x, los cuales después de la transformación se han centrado en base al 0, con una desviación estándar a 1, acercándose así más a los datos de una **distribución normal.** Esta transformación también se puede apreciar en la tabla siguiente dónde se muestran la media y la desviación estándar de cada una de las características antes de ser procesadas y después de ser procesadas.

	Non-processed Data		**Processed Data**	
	Mean	Std. Deviation	Mean	Std. Deviation
YEAR	2015.0	0.0	0.0	0.0
QUARTER	1.99822277361	0.810787590649	-4.75914287811e-17	1.0
MONTH	4.98763933341	2.53308455624	7.48483379921e-17	1.0
DAY_OF_MONTH	15.6782725512	8.76899176443	-2.85548572687e-17	1.0
DAY_OF_WEEK	3.96777778201	1.99229415498	-3.13670780603e-17	1.0
FL_NUM	1659.58065069	1476.88507961	-2.97446429882e-19	1.0
ORIGIN_AIRPORT_ID	13628.8258699	1278.50908029	5.20044521772e-16	1.0
DEST_AIRPORT_ID	13628.8448485	1278.4086313	-5.71719078817e-16	1.0
DEP_TIME	1334.20839216	500.178735327	2.45663710498e-17	1.0
DISTANCE	948.060912817	619.41129556	-1.04917467995e-17	1.0
CARRIER	7.86125456199	4.0369972964	-4.08853638165e-17	1.0

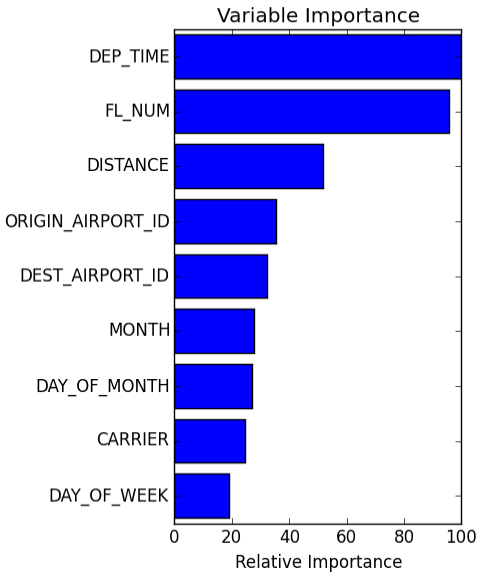
Selección de características

Este es el último proceso antes de la creación del modelo de predicción. En este punto se seleccionan las variables/características mas influyentes para la variable dependiente, que en este caso como bien ya se ha comentado anteriormente es la variable de **retraso en las llegadas al aeropuerto *(Arr_Delay_New).***

Una característica puede ser importante si está altamente correlacionada con la variable dependiente (variable a predecir).
Para ver que característica de entre las seleccionadas en el anterior apartado es la más influyente o tiene una mayor correlación con la variable dependiente se utiliza el **modelo *Random Forest*** mediante la creación de código en el programa Pycharm(*).

El resultado del código creado ha sido el siguiente:

Ilustración 29. Gráfica de Importancia de las variables en relación con la variable a predecir



Fuente: Elaboración propia

9 Important features(> 15 % of max importance):

['MONTH' 'DAY_OF_MONTH' 'DAY_OF_WEEK' 'FL_NUM' 'ORIGIN_AIRPORT_ID'

'DEST_AIRPORT_ID' 'DEP_TIME' 'DISTANCE' 'CARRIER']

Features sorted by importance (ASC):

['DAY_OF_WEEK' 'CARRIER' 'DAY_OF_MONTH' 'MONTH' 'DEST_AIRPORT_ID'

'ORIGIN_AIRPORT_ID' 'DISTANCE' 'FL_NUM' 'DEP_TIME']

Como se puede observar las características más importantes, con una importancia mayor al 15%, son 9. Las variables **de hora de salida** (*dep_time*) y número de vuelo (*fl_num*) encabezan esta lista con el mayor porcentaje de importancia y correlación con la variable dependiente de los retrasos en la llegada del aeropuerto.

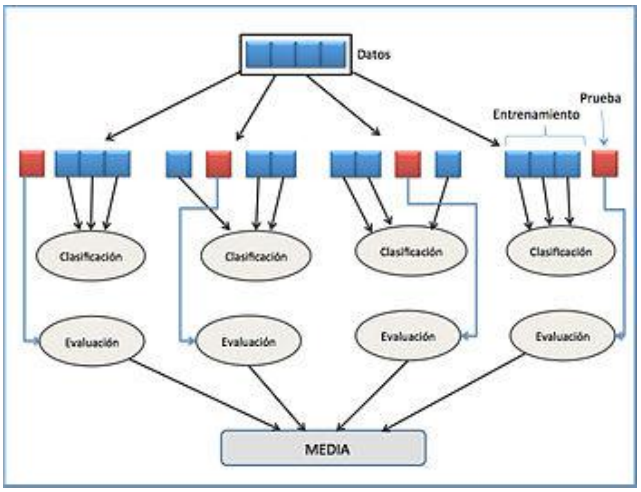
3.2.3 Creación de los modelos *Random Forests* y Gradiente de Árboles *Boosting* con la estructura de datos *Training with Cross-Validation and Testing.*

En este apartado es donde se crean los 2 modelos de predicción seleccionados, de los cuales se analizarán y compararán los resultados obtenidos para los datos disponibles de los vuelos realizados en los aeropuertos de Arizona.

Para la creación y evaluación de estos 2 modelos de predicción, primero, se organizan los datos de los que se disponen en dos conjuntos:

- Conjunto de entrenamiento (*Training set*): En este subconjunto se entrenan el **70%** del total de los datos disponibles. Es un subconjunto del conjunto de datos utilizados para construir modelos predictivos. Se puede decir que es donde el algoritmo aprende mediante el entrenamiento del visionado de los datos. En este conjunto de entrenamiento existe un proceso **intermedio de validación** del modelo con el fin de probar la calidad de dicho modelo y seleccionar el modelo con el mejor comportamiento. A este proceso se le llama **validación cruzada.**
    - **Validación Cruzada:**

      Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Se utiliza para estimar cómo de preciso es el modelo que se llevará a cabo a la práctica.

      La validación cruzada es una manera de predecir el

ajuste de un modelo a un hipotético conjunto de datos de prueba cuando no disponemos del conjunto explícito de datos de prueba

- Conjunto de prueba(*Testing set*): En este subconjunto se validan el **30%** del total de los datos disponibles. Es un subconjunto del conjunto de datos para evaluar el posible rendimiento futuro de un modelo. Si un modelo se ajusta al conjunto de entrenamiento mucho mejor de lo que se ajusta al conjunto de prueba, es posible que se esté dando un sobreajuste (*overfitting*).

Esta forma de división del conjunto de los datos es debido a que, evaluar el rendimiento del modelo únicamente con los datos utilizados para el entrenamiento no es aceptable en la minería de datos, ya que puede generar fácilmente modelos más optimistas y sobreajuste. Por ello se utiliza el **conjunto de prueba** para evaluar el rendimiento del modelo.

Para la creación del modelo de predicción se ha querido construir dos modelos con diferentes métodos para así, poder analizar y comparar los resultados de cada uno con el fin de evaluar el modelo que mejor se adapta a los datos disponibles.

Estos dos modelos seleccionados son:

- **Random Forest**
- **Gradiente de Arboles Boosting**

Estos 2 modelos se engloban dentro de los métodos denominados ***ensemble methods.*** Estos métodos combinan predicciones de varios estimadores base construidos con algoritmos de aprendizaje con el fin de mejorar la generalización y la robustez del estimador.

Se dividen en dos grupos:

- Métodos de promedio (*bagging*), donde la función principal es construir muchos estimadores independientes y promediar las predicciones.
  Bagging ofrece un incremento sobre la precisión de cualquiera de los clasificadores individuales utilizados. Además es muy robusto porque el modelo compuesto reduce la varianza de los clasificadores individuales a diferencia de utilizar un único estimador base.
  Un ejemplo de este modelo es el método de **Random Forest.**

- Métodos *Boosting*: los estimadores base se construyen secuencialmente y en cada iteración se reduce la desviación de los estimadores combinados. El objetivo es combinar varios modelos débiles para producir un conjunto fuerte.

Un ejemplo de este modelo es el método de **Gradiente de Árboles *Boosting.***

Ambos métodos proporcionan una manera de reducir el **sobreajuste,** aunque los métodos de ***bagging*** funcionan mejor con los modelos fuertes y complejos (por ejemplo, árboles de decisión plenamente desarrollados), en contraste con los métodos ***boosting*** que por lo general funcionan mejor con los modelos débiles (por ejemplo, árboles de decisión de poca profundidad).
Los métodos de conjuntos (*ensemble methods*) se utilizan tanto para los problemas de **clasificación** como de **regresión.**
En este caso de estudio, para la implementación de los dos tipos de modelos de predicción se utilizan **problemas de regresión.**

Desarrollo de los modelos de predicción en Pycharm

A la hora de la realización del código en Pycharm de los dos modelos de predicción, primero se ha creado una nueva carpeta donde irá el desarrollo de los modelos de predicción. Esta carpeta se ha nombrado **"prediction.py".**
Seguidamente se ha construido el código en Pycharm de los modelos de predicción. En esta parte el código se estructura de la forma siguiente:

- Primero se han pasado todos los datos transformados y guardados en el proceso anterior de *feature engineering* a esta nueva carpeta. De esta forma se ha tenido que llamar al archivo *.csv* con nombre **"flightsARIZONA_2015_transformed"** guardado en el anterior archivo Python "featuresengineering.py" y cargar los datos.
- Seguidamente se han establecido todas las variables predictivas, que son todas las características de las columnas de nuestro archivo en la base de datos (exceptuando la variable a predecir) y después se ha seleccionado la variable a predecir, que en este caso son los **minutos de retraso en las llegadas al aeropuerto**("ARR_DELAY_NEW").
- El tercer paso consiste en la creación de los modelos de regresión *Random Forest* y Gradiente de Árboles *Boosting*. Las dos iteraciones de código en Python utilizadas para su creación son las siguientes:

```
print ("Creating Random Forest model")
forest = ensemble.RandomForestRegressor(n_estimators=10,
min_samples_split=2, bootstrap=True, verbose=True, random_state=111)

print ("Creating Gradient Boosting Trees model")
gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=100,
max_depth=50, min_samples_split=150, verbose=True, random_state=111)
```

Como se puede observar cada modelo tiene una serie de campos/parámetros, los cuales, tienen gran importancia ya que si se varia el número de parámetros variará el resultado final de predicción. Esto es lo que se tratará de estudiar seguidamente. Antes de esto se describen los significados de los parámetros de cada modelo y se recuerda la función de cada uno:

- Parámetros modelo *Random Forest(RF)*:

  Función modelo RF:

  RF es un estimador meta que ajusta un numero de clasificación de arboles de decisión en varias sub-muestras del conjunto de datos y utiliza el promedio para mejorar la exactitud de la predicción y el control del sobreajuste.

  El tamaño sub-muestra es siempre el mismo que el tamaño original de la muestra de entrada, pero las muestras se extraen con el reemplazo en caso de que el bootstrap= True (predeterminado).

  n_estimators:
  integer, optional / (default =10)
  Es el número de árboles en el bosque o modelo.
  min_samples_split:
  integer, optional (default=2)
  Es el número mínimo de muestras requeridas para dividir un nodo interno. Nota: este parámetro es específico de los modelos de árbol.
  bootstrap:
  boolean, optional (default=True)
  En el caso de que las muestras bootstrap se utilicen cuando se construyen los arboles.
  verbose:
  int, optional (default=0)
  Controla el nivel de detalle del proceso de generación de árboles.
  random_state:
  int, RandomState instance or None, optional (default=None)
  Si es un int, random_state es la semilla usada para generar el numero aleatorio; Si RandomState es una *instance,* random_state es el generador de números aleatorios; Si es *None,* el generador de números aleatorios es la instancia de RandomState usada por np.random.

- Parámetros modelo Gradient Boosting Trees(GBT):

Función modelo GBT:

GBT construye un modelo aditivo hacia adelante por etapas; esto permite la optimización de las funciones de pérdida diferenciables arbitrarias. En cada etapa de un árbol de regresión se ajusta en el gradiente negativo de la función de pérdida dada.

loss:
{'ls', 'lad', 'huber', 'quantile'}, optional (default='ls')
La función loss se va optimizando.
'ls' se refiere a la regresión de mínimos cuadrados. 'lad' (menor desviación absoluta) es una función de pérdida muy robusta basada en información ordenada de las variables de entrada. 'huber' es una combinación de los dos. 'quantile' permite la regresión cuantil.
n_estimators:
int (default=100)
Es el número de etapas boosting a realizar. GB es bastante robusto frente a un sobreajuste (over-fitting), por lo que, **un gran número normalmente ofrece un mejor rendimiento.**
max_depth:
integer, optional (default=3)
Es la profundidad máxima de los estimadores de regresión individuales. La profundidad máxima limita el número de nodos en el árbol. Este parámetro se ha de **ajustar para obtener un mejor rendimiento**; el mejor valor depende de la interacción de las variables de entrada. Se ignora si max_leaf_nodes no es *None*.
min_samples_split:
integer, optional (default=2)
Es el número mínimo de muestras requeridas para dividir un nodo interno.
verbose:
int, default: 0
Habilita la salida detallada(verbose). Si es 1 entonces imprime el progreso y el rendimiento de vez en cuando (contra mas árboles la frecuencia es menor). Si es mayor que 1, entonces se imprime el progreso y rendimiento para cada árbol.
random_state:
int, RandomState instance or None, optional (default=None)
Si es un *int*, random_state es la semilla usada para generar el numero aleatorio; Si RandomState es una *instance,* random_state es el generador

de números aleatorios; Si es *None,* el generador de números aleatorios es la instancia de RandomState usada por np.random.

- Un cuarto paso es la división de todos los datos en dos partes: en un 70% de los datos para el entrenamiento(*Training et*), donde como antes se ha explicado, es donde se entrenan los datos de los que se disponen para que el algoritmo pueda aprender y así luego aplicar estos conocimientos en la parte de prueba(*Testing Set*), que es la otra parte que utiliza el 30% restantes del total de los datos, en relación a los vuelos realizados en los aeropuertos de Arizona. Esta última parte tratará de predecir (sin conocer los datos reales de la variable a predecir), mediante los conocimientos adquiridos en la parte de entrenamiento, el tiempo de retraso de los vuelos de nuestra base de datos.

- Finalmente se extraen los resultados, tanto los reales como los predichos, en referencia a la variable a predecir de **minutos de retraso de los vuelos en las llegadas a los aeropuertos,** en relación con los dos modelos utilizados en el proceso de predicción.
  La explicación del procedimiento, los métodos de evaluación y la comparación de los resultados entre modelos se detalla en el siguiente punto " 3.3 Análisis de los resultados".

3.3 Análisis de los resultados

3.3.1 Evaluación de los modelos: *Root Mean Square Error* (RMSE) y análisis visual de los retrasos reales y predichos en las llegadas

Para analizar los resultados de los modelos de predicción utilizados, primero se analizan individualmente los resultados de cada modelo y luego se comparan. Estos resultados se analizan mediante el **error medio cuadrático** *(Root Mean Square Error, RMSE)* y un **análisis visual** donde se comparan los resultados en tiempo de los retrasos reales y los resultados de los retrasos predichos en el modelo de predicción para las llegadas:

- **Root Mean Square Error(RMSE):**
  RMSE es una medida que cuantifica la calidad de las predicciones. Es decir mide las diferencias entre los valores predichos por un modelo o un estimador y los valores realmente observados.
  Estas diferencias individuales se denominan **residuos** cuando los cálculos se realizan sobre la muestra de datos que se utilizó para la estimación, y se denominan **errores de predicción** cuando se calcula fuera de la muestra. El

RMSE sirve para sumar las magnitudes de los errores en las predicciones para varios tiempos dentro de una única medida de poder predictivo. RMSD es una buena medida de la precisión, pero sólo para comparar los errores de predicción de los diferentes modelos para una variable particular y no entre las variables, ya que es dependiente de la escala. La fórmula es la siguiente:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}}$$

a = actual target
p = predicted target

Contra menor sea el valor del error de la anterior ecuación mejor será la predicción, es decir, la diferencia entre el modelo real y el de predicción será menor, por lo tanto el modelo de predicción se adaptara/asemejará mejor a los datos reales.

El mejor valor para un modelo de predicción seria el de con RMSE=0.

- **Análisis visual de los retrasos reales y predichos en las llegadas:**
  Se comparan los resultados obtenidos mediante una gráfica con dos funciones de variables, una función para los datos reales y otra función para los datos predichos en el modelo de predicción en relación al tiempo en minutos de retrasos en las llegadas.

Para la realización del análisis individual de los resultados de cada modelo el procedimiento y los resultados han sido los siguientes:

3.3.1.1 Procedimiento y resultados de Random Forest en Python:

Como bien se ha comentado anteriormente para la creación de los modelos, en este caso el de *Random Forest*, se necesita especificar una serie de parámetros los cuales, dependiendo del valor que se les asignen, pueden influir en el resultado de predicción final tanto positiva como negativamente. A continuación se ven las pruebas realizadas:

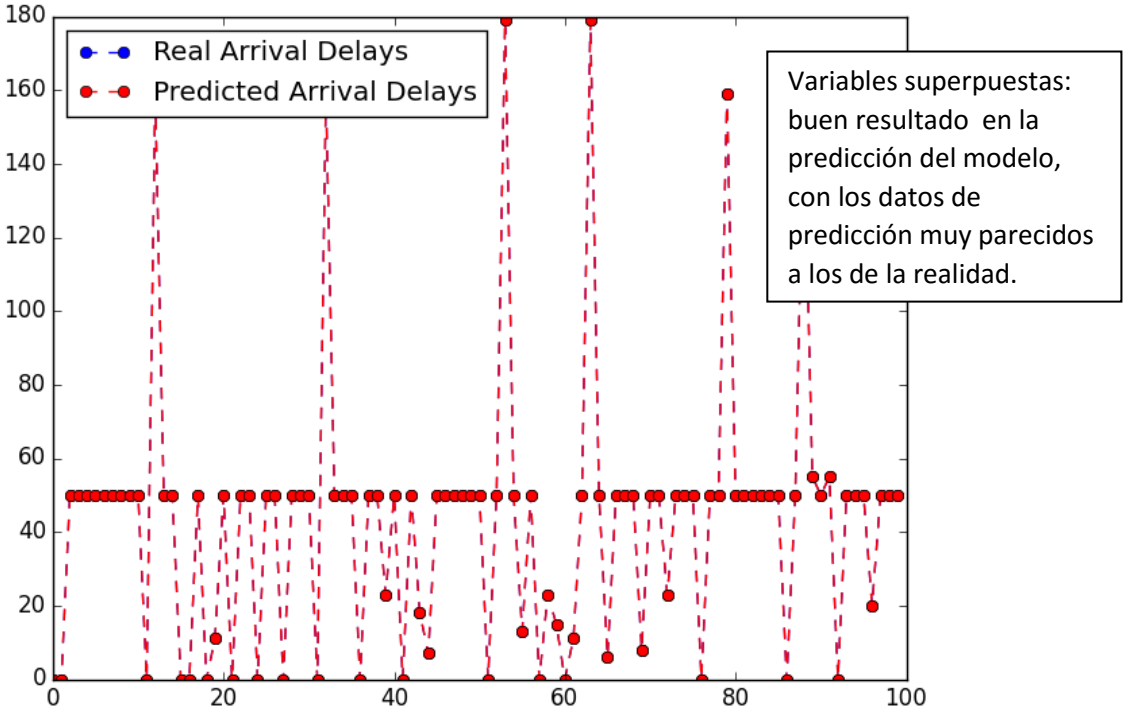
Prueba 1: valor bajo en los parámetros de n_estimators y min_samples_split

PARÁMETROS SELECCIONADOS:

```
forest = ensemble.RandomForestRegressor(n_estimators=12, min_samples_split=2,
bootstrap=True, verbose=True, random_state=111)
```

RESULTADOS:
Creating Random Forest model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Random Forest model using 'train_fold'
[Parallel(n_jobs=1)]: Done 12 out of 12 | elapsed: 4.8s finished
Testing Random Forest model using 'test_fold'
[Parallel(n_jobs=1)]: Done 12 out of 12 | elapsed: 0.0s finished
Estimating prediction error **Root Mean Squared Error**
1.8673720728 ⟶ Mejor cualificación del error respecto a las pruebas realizadas.

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 2: valores elevados en relación con los parámetros 'n_estimators' y 'min_samples_split'

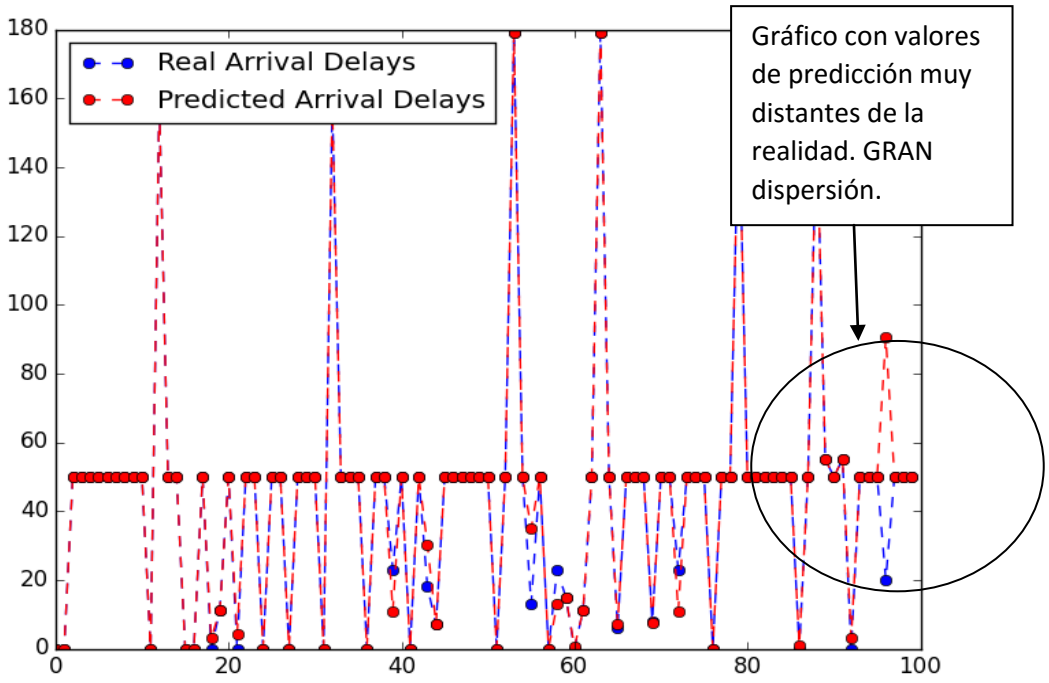
PARÁMETROS SELECCIONADOS:

forest = ensemble.RandomForestRegressor(n_estimators=500, min_samples_split=600, bootstrap=True, verbose=True, random_state=111)

RESULTADO:
Creating Random Forest model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Random Forest model using 'train_fold'
[Parallel(n_jobs=1)]: Done 49 tasks | elapsed: 18.7s
[Parallel(n_jobs=1)]: Done 199 tasks | elapsed: 1.3min
[Parallel(n_jobs=1)]: Done 449 tasks | elapsed: 2.9min

[Parallel(n_jobs=1)]: Done 500 out of 500 | elapsed: 3.2min finished
Testing Random Forest model using 'test_fold'
[Parallel(n_jobs=1)]: Done 49 tasks | elapsed: 0.0s
[Parallel(n_jobs=1)]: Done 199 tasks | elapsed: 0.4s
[Parallel(n_jobs=1)]: Done 449 tasks | elapsed: 1.1s
[Parallel(n_jobs=1)]: Done 500 out of 500 | elapsed: 1.2s finished
Estimating prediction error **Root Mean Squared Error**
6.13409658464

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 3: elevado valor de n_estimators y bajo valor en min_samples_split

PARÁMETROS SELECCIONADOS:

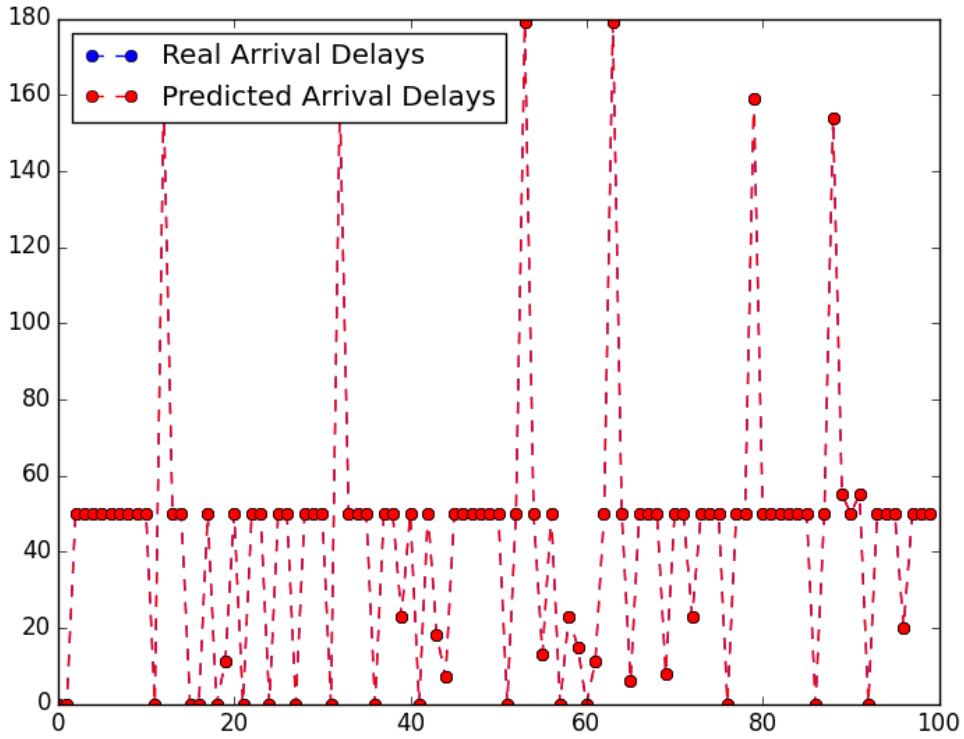
forest = ensemble.RandomForestRegressor(n_estimators=200, min_samples_split=2, bootstrap=True, verbose=True, random_state=111)

RESULTADOS:

[Parallel(n_jobs=1)]: Done 49 tasks | elapsed: 19.4s
[Parallel(n_jobs=1)]: Done 199 tasks | elapsed: 1.3min
Testing Random Forest model using 'test_fold'
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed: 1.3min finished
[Parallel(n_jobs=1)]: Done 49 tasks | elapsed: 0.0s

[Parallel(n_jobs=1)]: Done 199 tasks | elapsed: 0.5s
[Parallel(n_jobs=1)]: Done 200 out of 200 | elapsed: 0.5s finished
Estimating prediction error **Root Mean Squared Error**
1.900786022

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 4: valor pequeño de n_estimators y alto valor en min_samples_split

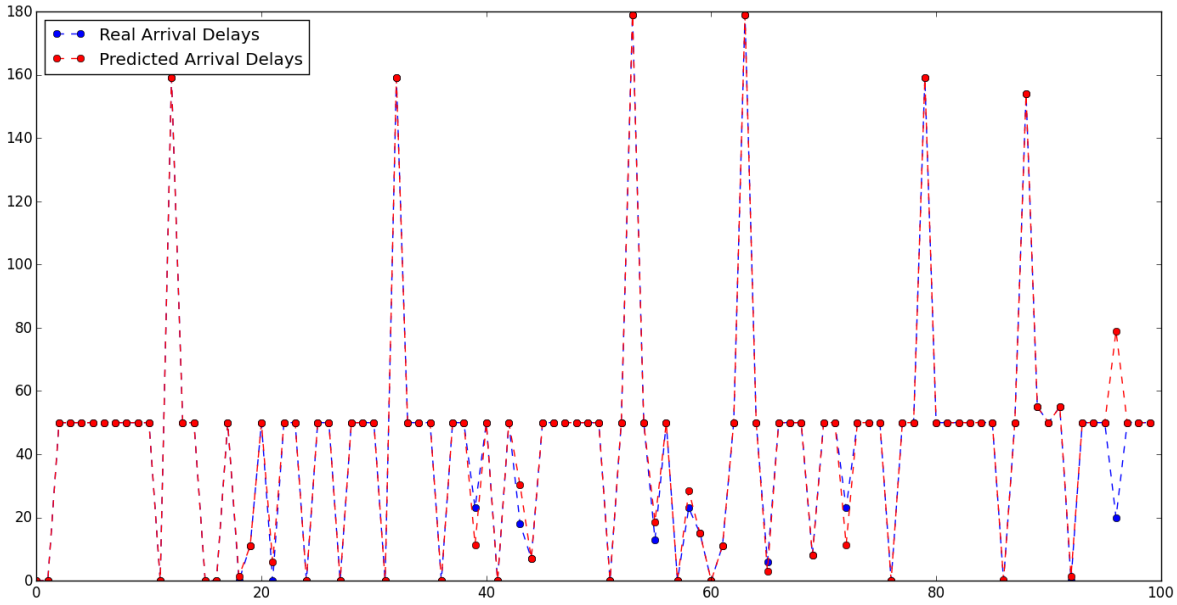
PARÁMETROS SELECCIONADOS:

forest = ensemble.RandomForestRegressor(n_estimators=12, min_samples_split=400, bootstrap=True, verbose=True, random_state=111)

RESULTADOS

Creating Random Forest model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Random Forest model using 'train_fold'
[Parallel(n_jobs=1)]: Done 12 out of 12 | elapsed: 4.5s finished
Testing Random Forest model using 'test_fold'
[Parallel(n_jobs=1)]: Done 12 out of 12 | elapsed: 0.0s finished
Estimating prediction error **Root Mean Squared Error**
5.28275344842

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Conclusiones de los resultados de las pruebas en el modelo Random Forest:

Podemos observar como valores muy elevados en los parámetros de **n_estimators** y **min_samples_split** ofrecen una baja calidad y rendimiento en relación al tiempo de computación del modelo y en base a los resultados de predicción obtenidos. Esto se observa en el alto resultado en el estimador del **error medio cuadrático** *(Root Mean Square Error, RMSE),* el cual, muestra que el modelo realizado de predicción difiere notablemente con el modelo real, aspecto este último negativo.

3.3.1.2 Procedimiento y resultados de Gradiente de Árboles Boosting en Python

Al igual que en el caso de *Random Forest*, se necesita especificar una serie de parámetros los cuales, dependiendo del valor que se les asignen, pueden influir en el resultado de predicción final tanto positiva como negativamente. A continuación se ven las pruebas realizadas:

Prueba 1: valor elevado en los parámetros de n_estimators, max_depth min_samples_split

PARÁMETROS SELECCIONADOS:

gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=100, max_depth=50, min_samples_split=150, verbose=True, random_state=111)

RESULTADOS

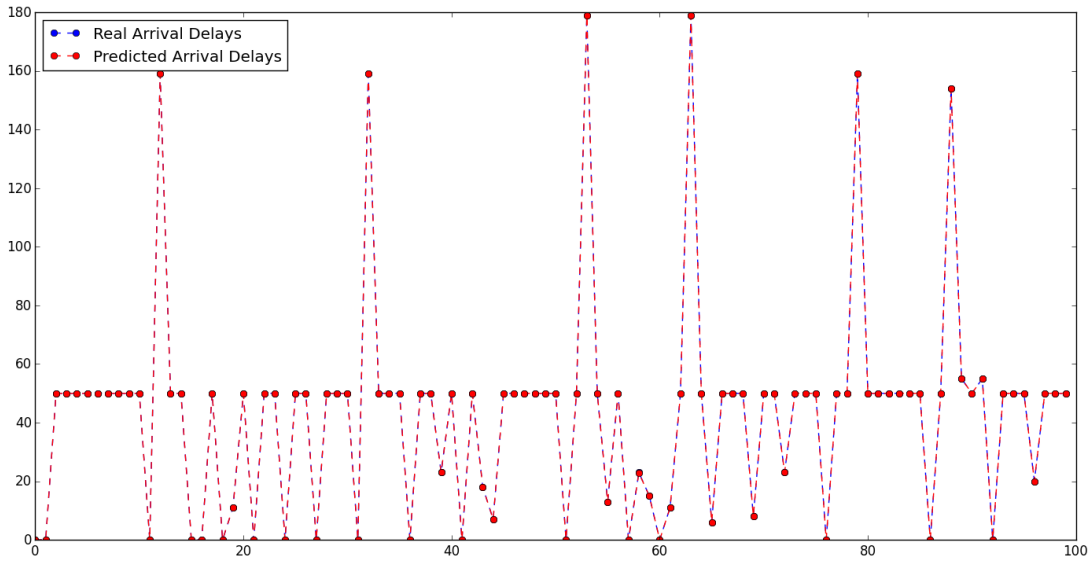
Creating Gradient Boosting model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Gradient Boosting model using 'train_fold'

Iter	Train Loss	Remaining Time
1	884.5176	1.47m
2	717.2084	1.65m
3	581.6734	1.57m
4	471.8840	1.54m
5	382.9415	1.49m
6	310.7928	1.46m
7	252.3415	1.43m
8	204.9912	1.40m
9	166.6057	1.39m
10	135.4609	1.37m
20	18.0789	1.23m
30	2.9232	1.09m
40	0.6782	57.14s
50	0.2244	48.03s
60	0.1051	39.70s
70	0.0637	31.07s
80	0.0381	21.83s
90	0.0233	12.01s
100	0.0159	0.00s

Se puede observar como la función de perdida(*Train Loss*) va disminuyendo a medida que se va construyendo el árbol. Ya que el algoritmo busca la optimización de estos.

Testing Gradient Boostong model using 'test_fold'
Estimating prediction error **Root Mean Squared Error**
0.806225481327

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 2: valor bajo en los parámetros de n_estimators, max_depth min_samples_split

PARÁMETROS SELECCIONADOS:

gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=10, max_depth=50, min_samples_split=15, verbose=True, random_state=111)

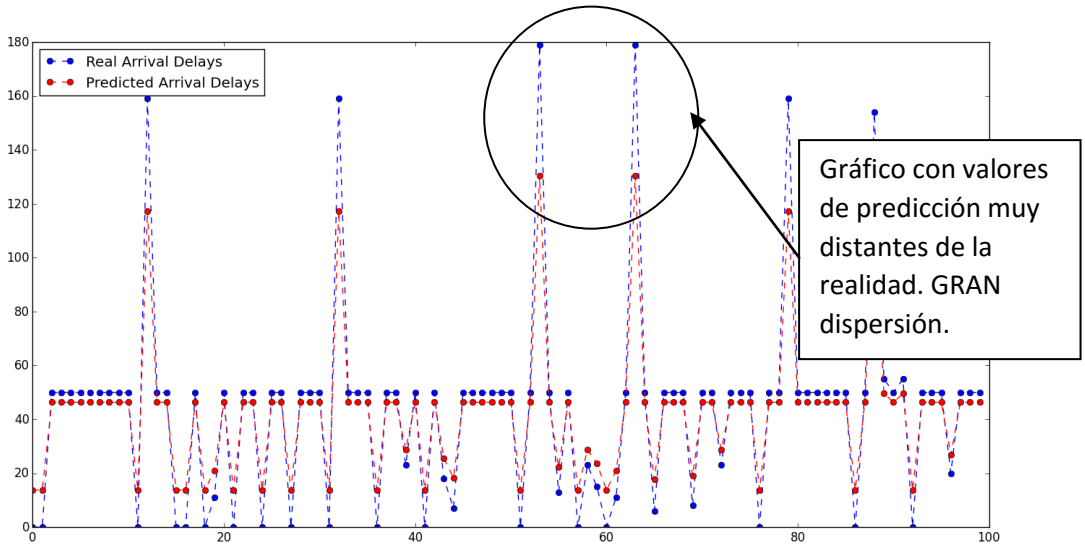
RESULTADOS:

Creating Gradient Boosting model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Gradient Boosting model using 'train_fold'

Iter	Train Loss	Remaining Time
1	883.9241	10.09s
2	716.2160	9.04s
3	580.3837	7.95s
4	470.3448	6.85s
5	381.0678	5.76s
6	308.8424	4.63s
7	250.2318	3.51s
8	202.7565	2.35s
9	164.3091	1.20s
10	133.1590	0.00s

Testing Gradient Boostong model using 'test_fold'
Estimating prediction error **Root Mean Squared Error**
11.7607584506

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 3: valor bajo en los parámetros de n_estimators y min_samples_split y valor elevado en max_depth

PARÁMETROS SELECCIONADOS:

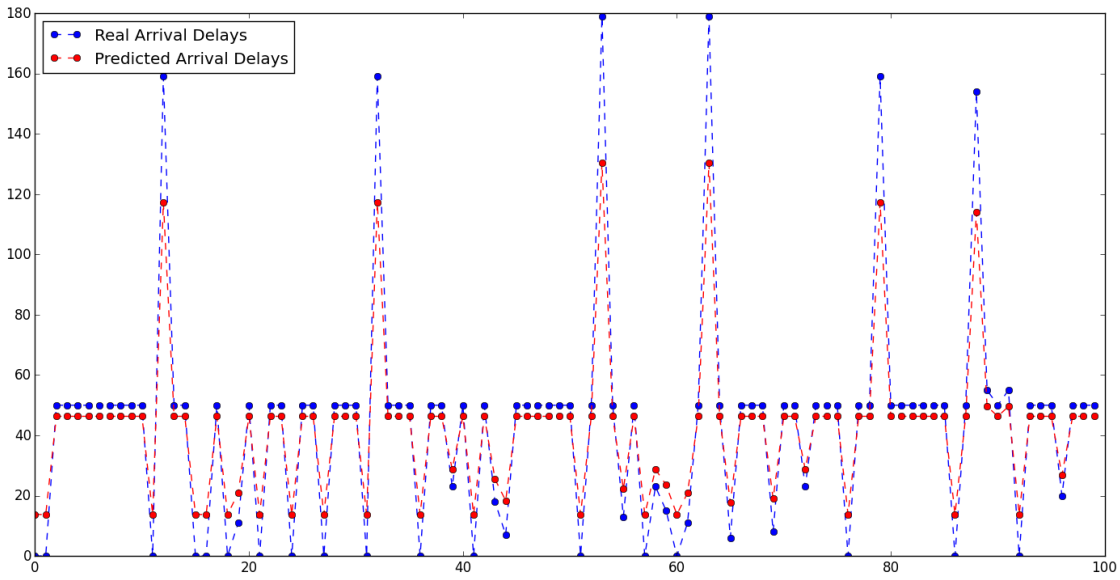
gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=10, max_depth=500, min_samples_split=15, verbose=True, random_state=111)

RESULTADOS:
Creating Gradient Boosting model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Gradient Boosting model using 'train_fold'

Iter	Train Loss	Remaining Time
1	883.9241	10.71s
2	716.2160	9.36s
3	580.3837	8.20s
4	470.3448	7.06s
5	381.0678	6.03s
6	308.8424	4.83s
7	250.2318	3.63s
8	202.7565	2.42s
9	164.3091	1.21s
10	133.1590	0.00s

Testing Gradient Boostong model using 'test_fold'
Estimating prediction error **Root Mean Squared Error**
11.7607584506

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 4: valor bajo en los parámetros de max_depth y min_samples_split y valor elevado en n_estimators

PARÁMETROS SELECCIONADOS:

gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=500, max_depth=10, min_samples_split=15, verbose=True, random_state=111)

RESULTADOS:

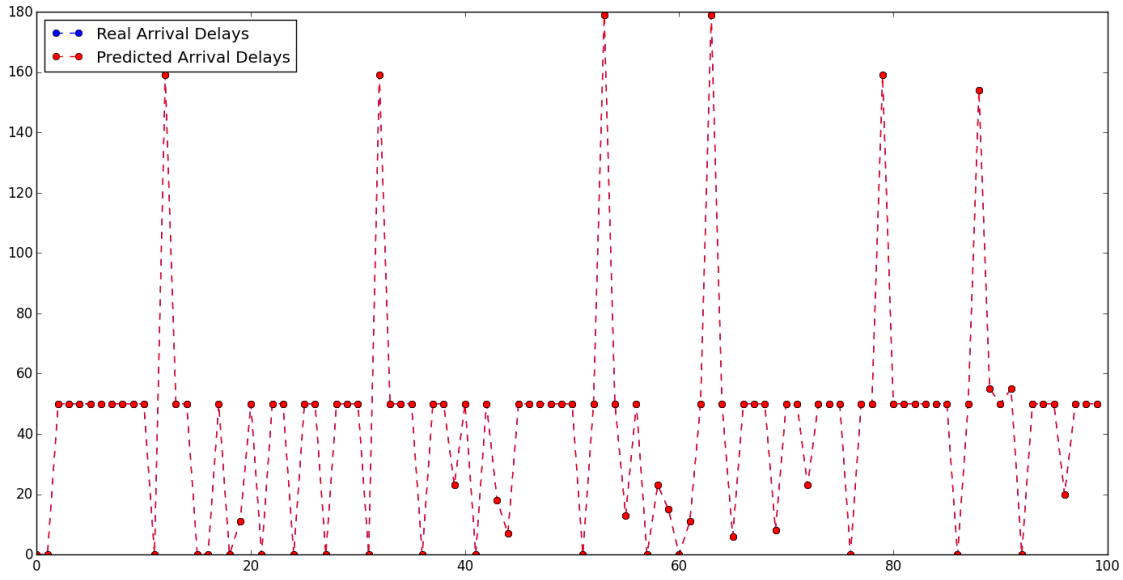
Creating Gradient Boosting model
Splitting 'alldata' into two sets: 70% Training and 30% Testing
Training Gradient Boosting model using 'train_fold'

Iter	Train Loss	Remaining Time
1	887.7999	4.57m
2	723.2880	4.55m
3	589.9508	4.54m
4	481.9737	4.52m
5	394.4905	4.50m
6	323.6563	4.50m
7	265.9884	4.49m
8	219.3998	4.48m
9	181.4262	4.48m
10	149.9825	4.47m
20	30.4971	4.38m
30	11.8855	4.30m

40	5.7766	4.32m
50	2.8819	4.36m
60	1.7145	4.34m
70	1.1360	4.30m
80	0.7414	4.22m
90	0.4945	4.12m
100	0.3764	4.03m
200	0.0295	3.08m
300	0.0037	2.07m
400	0.0007	1.05m
500	0.0001	0.00s

Testing Gradient Boostong model using 'test_fold'
Estimating prediction error **Root Mean Squared Error**
1.33604168168

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Prueba 5: valor más elevado en el parámetros de n_estimators en comparación con la "Prueba 1".

PARÁMETROS SELECCIONADOS:

gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=500, max_depth=50, min_samples_split=150, verbose=True, random_state=111)

RESULTADOS

Creating Gradient Boosting model

Splitting 'alldata' into two sets: 70% Training and 30% Testing

Training Gradient Boosting model using 'train_fold'

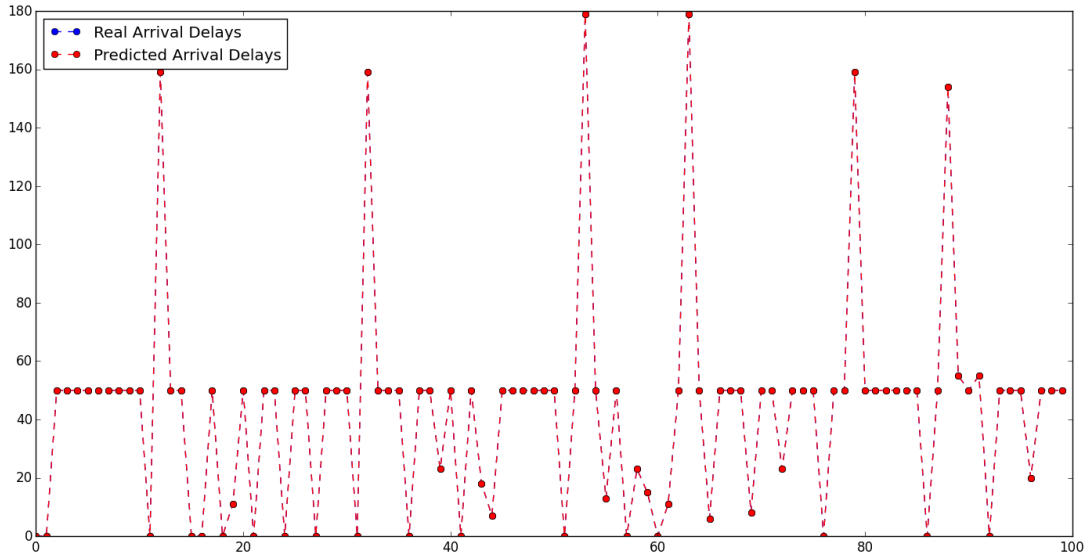
Iter	Train Loss	Remaining Time
1	884.5176	6.88m
2	717.2084	7.18m
3	581.6734	7.17m
4	471.8840	7.10m
5	382.9415	7.08m
6	310.7928	7.09m
7	252.3415	7.07m
8	204.9912	7.06m
9	166.6057	7.07m
10	135.4609	7.08m
20	18.0789	7.15m
30	2.9232	7.22m
40	0.6782	7.22m
50	0.2244	7.08m
60	0.1051	7.15m
70	0.0637	7.31m
80	0.0381	7.47m
90	0.0233	8.04m
100	0.0159	8.35m
200	0.0033	8.30m
300	0.0017	6.03m
400	0.0010	3.15m
500	0.0007	0.00s

Testing Gradient Boostong model using 'test_fold'

Estimating prediction error **Root Mean Squared Error**

0.805627244272

GRÁFICO ENTRE LOS RETRASOS PREDECIDOS Y LOS RETRASOS REALES:



Conclusiones de los resultados de las pruebas en el modelo Gradiente de Árboles Boosting:

Se puede observar que para el modelo de Gradiente de Árboles Boosting los mejores resultados en relación con el **error medio cuadrático** *(Root Mean Square Error, RMSE)* se han mostrado cuando el número en las etapas de creación del árbol es **elevado** (n_estimators). El mínimo RMSE conseguido ha sido en la última prueba 5 con 0.805627244272.
Esto quiere decir, que contra más iteraciones se hagan en el modelo mejores resultados se obtienen en función del rendimiento en los tiempos estimados de retraso en las llegadas de los vuelos a los aeropuertos.
Un aspecto a destacar es el tiempo en la realización del modelo de predicción que tiende a ser elevado en cuanto aumentamos el numero de iteraciones a realizar. A demás, si comparamos los resultados del problema 1 y del problema 5 veremos que la diferencia de error no es mucho mejor para el 5 y sí que hay una diferencia notable para el tiempo de procedimiento entre uno y otro.

3.3.2 Comparación de los modelos

De entre los resultados de los dos modelos podemos observar que el que ha conseguido un menor **error medio cuadrático** *(Root Mean Square Error, RMSE)* en las diferentes realizaciones de las pruebas con los parámetros ha sido el de **Gradiente de Árboles *Boosting*(GB)** con un RMSE de 0.805627244272, aunque para este modelo el tiempo de ejecución es mayor que para el modelo *Random Forest*(RF) ya que el primero

necesita de un mayor nombre de iteraciones para conseguir un buen resultado del modelo de predicción.

Para Random Forest, en cambio el mejor modelo analizado muestra un RMSE de 1.8673720728. Este modelo se caracteriza por utilizar un nombre bajo en sus parámetros de n_estimators y min_samples_split, es decir, en los parámetros en relación al número de árboles del modelo y el numero mínimo d muestras requeridas para dividir un nodo interno. Por lo que es un buen estimador del rendimiento en relación con el tiempo de ejecución y los resultados obtenidos de la predicción de los tiempos de retraso en las llegadas de los vuelos a los aeropuertos de Arizona.

CONCLUSIONES

Para este último apartado de conclusiones se quiere hacer referencia principalmente a los objetivos planteados en el inicio de este proyecto. Se puede decir que el resultado en verso a la realización de éstos es exitoso, ya que se han cumplido la totalidad de ellos.

Dicho esto se puede empezar explicando los resultados que se han ido obteniendo a lo largo de este proyecto.

En la parte inicial se han estudiado los impactos que los retrasos aéreos generan en el transporte aéreo. Primero se ha podido ver como estos retrasos son consecuencia de diversos factores que influyen drásticamente en su desarrollo. Se han visto factores tales como la **congestión** de los vuelos en los aeropuertos debido a la planificación inadecuada de las compañías aéreas al no tener en cuenta los vuelos planificados del resto de las compañías. Esto conlleva a un exceso de vuelos programados en las horas punta provocando entonces los retrasos tanto en los vuelos propios como en el resto.

De aquí también se extrae otra consecuencia implícita en los retrasos que es la de el efecto de propagación a lo largo de sus operaciones. Este efecto es debido al modelo **secuencial** que poseen las operaciones en el transporte aéreo, donde si un vuelo de una compañía aérea se retrasa, si no se elimina este retraso, afectará tanto a los vuelos posteriores de la misma compañía como a los vuelos y operaciones de todo el entorno aeroportuario, dado que el aeropuerto es una estructura limitada donde no se concibe una circulación sin permiso de entrada con una hora(slot) programada para cada vuelo.

Otras consecuencias se han podido ver mediante el estudio comparativo de los vuelos en los Estados Unidos y la Unión Europea. De estos análisis se extrae una clasificación precisa la cual es dividida en 5 grupos:

- Aerolínea + *Turnaround* Local
- Tiempo Extremo
- Aeronaves que llegan tarde (o retraso reaccionario)
- Seguridad
- Sistema ATM(retrasos ATFM / NAS) donde se incluyen los retrasos debidos a condiciones no extremas en el tiempo.

De los datos más relevantes en el periodo para 2013 se traducen con que, para la Unión Europea las 2 principales causas de retraso se debe a un 5,9% por causas de la aerolínea + *turnaround* Local y un 7, 7% a retrasos reaccionarios, a diferencia que las causas en los vuelos de los Estados Unidos donde sus dos mayores porcentajes son en relación al sistema ATM con causas debidas mayoritariamente al mal tiempo y causas por retrasos reaccionarios, con un 6,8% y un 7,6% respectivamente.

También destacar que alrededor del 80% de los vuelos realizados se han llevado a cabo dentro del tiempo planificado tanto para UE como para US.

Seguidamente, en otro punto de estudio, se han podido investigar las consecuencias de estos retrasos.
En este punto se ha podido ver como éstos repercutían tanto a compañías aéreas y a aeropuertos en base a mayores costos económicos en relación de la ineficiencia operativa en los tiempos de operaciones, como a los usuarios que utilizan este medio de transporte en relación con el tiempo y los costes de oportunidad perdidos por los retrasos sufridos en los vuelos, lo cual ha significado una percepción menor de la calidad del servicio final para éstos.

También se han podido ver los diferentes tipos de **enfoques para minimizar** los retrasos aéreos. Éstos son junto con los modelos de predicción una herramienta indispensable para la mejora del problema en el sistema. Muchos de estos enfoques descritos se basan en **técnicas del intercambio de datos** (*data sharing)* para mejorar la previsibilidad en el transporte aéreo y así poder controlar y disminuir los problemas de los retrasos en los vuelos.

Otro objetivo de este proyecto era conocer los diferentes modelos de predicción mediante su análisis. Respecto a este punto hemos podido desarrollar los modelos de *Random Forest,* Regresión Lineal, redes Neuronales y Gradiente de Arboles *Boosting*. Hemos podido ver como cada uno de sus algoritmos poseía de una caracterizada composición y forma de construcción del modelo en cuestión, donde en esta diferenciación, radica el poder de elección frente a un modelo o otro el cual se pueda adaptar al problema o modelo a desarrollar de la mejor forma.

Para el correcto funcionamiento y construcción del modelo de predicción también se proponía analizar los diferentes pasos claves de **minería de datos** que se llevan a cabo para una correcta creación del modelo de predicción. Estos pasos han sido desarrollados y llevados a la practica en el caso de estudio de predicción de los tiempos de retrasos en los vuelos realizados en los aeropuertos de Arizona para el periodo de enero a septiembre de 2015.
Respecto a los resultados y conclusiones mas importantes de estos puntos se muestran a continuación:
En relación con los análisis realizados de las 12 compañías aéreas y de los 80 aeropuertos de la base de datos se han podido extraer que las compañías con más retrasos para el periodo de estudio de enero a septiembre de 2015 han sido ***JetBlue Aiways(B6), Frontier Airlines(F9), Spirit Airlines (NK)*** con una media de alrededor de entre 15 a 20 minutos de retraso.

En relación con el porcentaje mayor de retrasos en los aeropuerto de salida el que concibe un porcentaje mayor es el de **de Washington, DC (IAD)** con una media de alrededor de 25 minutos de retraso por vuelo.

Y en relación con el aeropuerto con el mayor porcentaje de retrasos en los aeropuerto de llegada es el de **Buffalo, NY (BUF)** con una media de alrededor de 20 minutos de retraso por vuelo.

En relación con el último apartado de los resultados en la creación de los modelos de predicción con *Random Forest* y Gradiente de Arboles *Boosting,* se han extraído las siguientes conclusiones:

Los dos modelos han sido capaces de predecir los tiempos de retrasos en las llegadas a los aeropuertos de los vuelos de Arizona con unos resultados predichos muy semejantes a los datos reales. Esto se puede observar mediante los resultados obtenido con la medida de evaluación de la predicción en base al error cuadrático medio RMSE. El modelo de Gradiente de Árboles Boosting ha obtenido un RMSE de **0.80562724427,** mientras que Random Forest ha obtenido un RMSE peor de **1.8673720728.**

A demás de estos datos anteriores y de todas las pruebas realizadas en base a las modificaciones en los parámetros de cada modelo, se ha podido extraer que el modelo de Gradiente de Árboles Boosting tiene unos mejores resultados en relación con el error cuadrático medio RMSE en relación con la predicción de los tiempos de retrasos, pero necesita de mayores pruebas e iteraciones para encontrar el mejor modelo, con lo cual el rendimiento no es del todo óptimo.
En contrapartida a esto, el Random Forest aún y tener un número más elevado de RMSE y no tener una predicción que se ajusta tan bien como el Gradiente de Árboles Boosting, el RF tiene la capacidad de encontrar un buen resultado de predicción en un menor tiempo de ejecución que el de GB, dado que este puede trabajar en **paralelo** a diferencia que el modelo de trabajo **secuencial** GB, de ahí que el GB necesite un numero grande en sus parámetros para conseguir unos buenos resultados de predicción.

BIBLIOGRAFÍA

- Identifican las cinco causas fundamentales de los retrasos aéreos: http://www.hosteltur.com/57156_identifican-cinco-causas-fundamentales-retrasos-aereos.html

- Analysis of aircraft arrival and departure delay characteristics: http://chadalong.com/Education/Taxi%20Network/mueller_10_02.pdf

- Understanding Flight Delays at U.S. Airports in 2010, Using Chicago O'Hare International Airport as a Case Study: http://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=1033&context=masters_theses

- Joint Economic Committee, Your Flight Has Been Delayed Again: http://www.jec.senate.gov/public/_cache/files/47e8d8a7-661d-4e6b-ae72-0f1831dd1207/yourflighthasbeendelayed0.pdf

- Airlines of America, Per-Minute Cost of Delays to U.S. Airlines: Understanding Flight Delays at U.S. Airports in 2010, Using Chicago O'Hare International Airport as a Case Study: http://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=1033&context=masters_theses

- Nextor, Total Delay Impact Study: http://www.isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf

- Flight Delays, Capacity Investment and Social Welfare under Air Transport Supply-Demand Equilibrium: http://ses.wsu.edu/wp-content/uploads/2014/09/TRA_Flight-delays_capacity-investment-and-social-welfare-under-air-transport-supply-demand-equilibrium.pdf

- AESA, Información sobre los derechos de los pasajeros: http://www.seguridadaerea.gob.es/lang_castellano/particulares/derechos_pax/info_derechos/default.aspx

- EUROCONTROL, EUROCONTROL Annual Report 2014 http://www.eurocontrol.int/publications/eurocontrol-annual-report-2014

- AENA, Fases del vuelo: http://www.aena.es/csee/Flash/html/controlAereo02_13.jsp

- EUROCONTROL, The propagation of air transport delays in Europe https://www.eurocontrol.int/sites/default/files/content/documents/official-documents/facts-and-figures/coda-reports/propagation-delays-2009.pdf

- Federal Aviation Administration https://www.faa.gov/nextgen/snapshots/stories/?slide=9

- Sesar, Annual Report 2014: http://www.sesarju.eu/sites/default/files/documents/reports/SESAR-annual-report-2014.pdf

- Scikit Learn: http://scikit-learn.org/stable/modules/ensemble.html

- FDA, Bioanalytical Method Validation, Cross Validation http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/ucm070107.pdf

- Leo Breiman, Statistics Department University of California, Random Forest: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

- Saed Sayad, data mining http://www.saedsayad.com/data_mining_map.htm

ANEXO (código en *Python*)

El código se divide en 3 diferentes archivos: "dataanalysis.py", "featuresengineering.py" y "prediction.py". Estas 3 archivos se muestran a continuación:

Archivo "dataanalysis.py":

```
import pandas as pd
import seaborn as sbn
import matplotlib.pyplot as plt
from numpy import arange
import bisect

#plt.style.use('ggplot')
sbn.set_style("whitegrid")

_author_ = 'Nerea'

df1 = pd.read_csv("dataARIZONA/ARIZONA_ENERO.csv")
df2 = pd.read_csv("dataARIZONA/ARIZONA_FEBRERO.csv")
df3 = pd.read_csv("dataARIZONA/ARIZONA_MARZO.csv")
df4 = pd.read_csv("dataARIZONA/ARIZONA_ABRIL.csv")
df5 = pd.read_csv("dataARIZONA/ARIZONA_MAYO.csv")
df6 = pd.read_csv("dataARIZONA/ARIZONA_JUNIO.csv")
df7 = pd.read_csv("dataARIZONA/ARIZONA_JULIO.csv")
df8 = pd.read_csv("dataARIZONA/ARIZONA_AGOSTO.csv")
df9 = pd.read_csv("dataARIZONA/ARIZONA_SEPTIEMBRE.csv")

result = (df1, df2, df3, df4, df5, df6, df7, df8, df9)
df = pd.concat(result)

df = df[df.CANCELLATION_CODE.isnull()]
#print df

#print df.isnull().any()

#Ejercicio 2.3
delay_column_names = ["ARR_DELAY_NEW", "ARR_DEL15", "CARRIER_DELAY",
                      "WEATHER_DELAY", "NAS_DELAY", "SECURITY_DELAY",
"LATE_AIRCRAFT_DELAY"]

df[delay_column_names] = df[delay_column_names].fillna(0)

#print df.isnull().any()

#Ejercicio 2.4
selected_delay_column_names = ["CARRIER_DELAY", "WEATHER_DELAY",
"NAS_DELAY", "SECURITY_DELAY", "LATE_AIRCRAFT_DELAY"]

mask = (df['ARR_DELAY_NEW'] > 0) &
(df[selected_delay_column_names].sum(axis=1) == 0)
df.ix[mask, 'CARRIER_DELAY'] = df.ix[mask, 'ARR_DELAY_NEW']

#print df


df = df.drop("Unnamed: 38", 1)
df = df.drop("CANCELLATION_CODE", 1)
```

```
df = df.drop("CANCELLED", 1)
#print df.head()

df = df.dropna()

#print df.isnull().any()
#print df
#print df.head()

df.to_csv("flights3.csv")

def scatterplot(x, y, x_title, y_title):
    plt.plot(x, y, 'b.')
    plt.xlabel(x_title)
    plt.ylabel(y_title)
    plt.xlim(min(x)-1, max(x)+1)
    plt.ylim(min(y)-1, max(y)+1)
    plt.show()

def barplot(labels, data, x_title, y_title):
    pos = arange(len(data))
    plt.xlabel(x_title)
    plt.ylabel(y_title)
    plt.xticks(pos+0.4, labels)
    plt.bar(pos, data)
    plt.show()

def histplot(data,x_title, y_title,bins= None,nbins= 2):
    minx, maxx = min(data), max(data)
    space = (maxx-minx)/float(nbins)
    if not bins:
        bins = arange(minx, maxx, space)
    binned = [bisect.bisect(bins, x) for x in data]
    l = ['%i' % x for x in list(bins)+[maxx]]\
        if space < 1 \
        else [str(int(x))
              for x in list(bins)+[maxx]]
    displab = [x+'-'+y for x, y in zip(l[:-1], l[1:])]

    barplot(displab, [binned.count(x+1)for x in range(len(bins))],
x_title, y_title)

# funcion para datos del tipo string(caracter)
def barchart(x, y, x_title, y_title, numbins=10):
    data = pd.DataFrame()
    data['DEST'] = df['DEST']
    data['ARR_DELAY_NEW'] = df['ARR_DELAY_NEW']
    carrier_group = data.groupby('DEST')
    delays_totals = carrier_group.sum()  # here you may use sum()
instead of mean(), when it's appropriate
    delays_totals.sort(columns='ARR_DELAY_NEW')
    ax = delays_totals.plot(kind='bar', title="Arrivals Delays by
Destination Airport", legend=False)
    ax.set_xlabel("DEST")
    ax.set_ylabel("Total Arrivals Delays(minutes)")
    plt.show()


# funcion para datos de tipo integer (numeros)
def barchart(x, y, x_title, y_title, numbins=10):
    datarange = max(x)-min(x)
```

```
    bin_width = float(datarange)/numbins
    pos = min(x)
    bins = [0 for i in range(numbins+1)]

    for i in range(numbins):
        bins[i] = pos
        pos += bin_width
        bins[numbins] = max(x)+1
        binsum = [0 for i in range(numbins)]
        bincount = [0 for i in range(numbins)]
        binaverage =[0 for i in range(numbins)]

    for i in range(numbins):
        for j in range(len(x)):
            if x[j]>=bins[i] and x[j]<bins[i+1]:
                bincount[i] += 1
                binsum[i] += y[j]
    for i in range(numbins):
        binaverage[i] = float(binsum[i])/bincount[i]

    barplot(range(numbins), binaverage, x_title, y_title)

scatterplot(df.DEP_TIME, df.ARR_DELAY_NEW, "DEP_TIME",
"ARR_DELAY_NEW")
print scatterplot

histplot(df.DEP_DEL15, 'DELAY > 15', 'FRECUENCIA')
print histplot

barchart(df.DEST.values, df.ARR_DELAY_NEW.values, "DEST",
"ARR_DELAY_NEW", len(df.DEST.unique()))
print barchart

df.to_csv("dataARIZONA/flightsArizona_2015.csv", index=False)
```

Archivo "featuresengineering.py":

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing
import matplotlib.pyplot as plt
from sklearn import ensemble
import scipy.stats as stats
import random

__author__ = 'Nerea'

df = pd.read_csv("dataARIZONA/flightsArizona_2015.csv")

# Ejercicio 4.1.3
def build_features(features,data):
    #Firts we add numeric variables
    features.extend(['YEAR', 'QUARTER', 'MONTH', 'DAY_OF_MONTH',
'DAY_OF_WEEK', 'FL_NUM',
                     'ORIGIN_AIRPORT_ID', 'DEST_AIRPORT_ID',
'DEP_TIME', 'DISTANCE'])
    #Secondly we add categorical variables and transform them into the
numerical ones
```

```
    features.append('CARRIER')
    le = LabelEncoder().fit(data['CARRIER'])
    data['CARRIER'] = le.transform(data['CARRIER'])

    return data

features = []
build_features(features, df)
print df[features].head()

# Ejercicio 4.2.3.
for x in range(0, len(features)):
    plt.figure(x)
    plt.hist(df[features].values[:, x])
    plt.title(features[x])
    plt.xlabel("Value")
    plt.ylabel("Frequency")
    plt.show()

# Ejercicio 4.2.4
for x in range(0, len(features)):
    print "%s Mean:" % features[x]
    print df[features].values[:, x].mean(axis=0)
    print "%s Std.Dev.:" % features[x]
    print df[features].values[:, x].std(axis=0)

df_scaled = preprocessing.scale(df[features])

for x in range(0, len(features)):
    print "%s Mean:" % features[x]
    #print stats.skew(df_scaled[:,x])
    print df_scaled[:,x].mean(axis=0)
    print "%s Std.Dev.:" % features[x]
    print df_scaled[:, x].std(axis=0)

delayed_minutes = df['ARR_DELAY_NEW']

#Fit a random forest with(mostly) default parameters to determine
feature importance
random.seed(111)
forest = ensemble.RandomForestRegressor(n_estimators=10,
                                        min_samples_split=2,
                                        n_jobs=-1)
forest.fit(df_scaled, delayed_minutes)
feature_importance = forest.feature_importances_

#Make importances relative to max importance
feature_importance = 100.0 * (feature_importance /
                              feature_importance.max())

#A threshold below which to drop features from the final data set.
#Specifically, this number represents the percentage of the most
important feature's importance valeu
fi_threshold = 15

# Get the indexes of all features over the importance threshold
important_idx = np.where(feature_importance > fi_threshold)[0]

#Create a list of all the feature names above the importance threshold
features = np.array(features)
important_features = features[important_idx]
```

```
print "\n", important_features.shape[0], "Important features(>",
fi_threshold, "% of max importance):\n",\
    important_features

#Get the sorted indexes of important features
sorted_idx = np.argsort(feature_importance[important_idx])
print "\nFeatures sorted by importance (ASC):\n",
important_features[sorted_idx]

#Plot the importance of features
pos = np.arange(sorted_idx.shape[0]) + .5
plt.subplot(1, 2, 2)
plt.barh(pos, feature_importance[important_idx][sorted_idx],
align='center')
plt.yticks(pos, important_features[sorted_idx])
plt.xlabel('Relative Importance')
plt.title('Variable Importance')
plt.draw()
plt.show()

#Remove non-important features from the feature set, and reorder those
remaining
df_scaled = df_scaled[:, important_idx][:, sorted_idx]

df_scaled = pd.DataFrame(df_scaled)
df_scaled['ARR_DELAY_NEW'] = delayed_minutes
important_features = np.append(important_features, ['ARR_DELAY_NEW'])
sorted_idx = np.append(sorted_idx, len(important_features)-1)

#Save final processed features in the csv file
df_scaled.to_csv("dataARIZONA/flightsARIZONA_2015_transformed.csv",
header=important_features[sorted_idx], index=False)


for x in range(0, len(df_scaled.columns)):
    plt.figure(x)
    plt.hist(df_scaled.values[:, x])
    plt.title(features[x])
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.show()
```

Archivo " prediction.py ":

```
import pandas as pd
import tabulate as t
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn import ensemble
from sklearn import cross_validation


def RMSE(y, yhat):
    rmspe = np.sqrt(np.mean((y - yhat)**2))
    return rmspe
```

```
df = pd.read_csv("dataARIZONA/flightsARIZONA_2015_transformed.csv")

alldata = df.ix[:, 0:(len(df)-1)] # selecting predictors, i.e. all
columns except 'ARR_DELAY_NEW'
target = df['ARR_DELAY_NEW'] # selecting variable to be predicted

#print ("Creating Random Forest model")
print ("Creating Gradient Boosting model")
#forest = ensemble.RandomForestRegressor(n_estimators=12,
min_samples_split=400, bootstrap=True, verbose=True, random_state=111)
gbm = ensemble.GradientBoostingRegressor(loss='ls', n_estimators=500,
max_depth=50, min_samples_split=150, verbose=True, random_state=111)

print ("Splitting 'alldata' into two sets: 70% Training and 30%
Testing")
train_fold, test_fold, train_y, test_y =
cross_validation.train_test_split(alldata, target, test_size=0.3,
random_state=123)

#print ("Training Random Forest model using 'train fold'")
print ("Training Gradient Boosting model using 'train_fold'")
#m = forest.fit(train_fold, target[train_y])
m = gbm.fit(train_fold, target[train_y])

#print ("Testing Random Forest model using 'test fold'")
print ("Testing Gradient Boostong model using 'test_fold'")
predicted_y = m.predict(test_fold)

results = pd.DataFrame()
results['Real ARR_DELAY_NEW'] = target[test_y]
results['Predicted ARR_DELAY_NEW'] = predicted_y
results.to_csv("dataARIZONA/predictionresults1.csv", index=False)

print ("Estimating prediction error Root Mean Squared Error")
print RMSE(target[test_y], [y for y in predicted_y])

# VISUALIZE PREDICTED AND REAL ARRIVAL DELAYS
plt.plot(target[test_y].iloc[0:100], marker='o', linestyle='--',
color='b')
plt.plot(predicted_y[0:100], marker='o', linestyle='--', color='r')
plt.legend(['Real Arrival Delays', 'Predicted Arrival Delays'],
loc='upper left')
plt.show()
```