Trabajo de Fin de Master

Máster Big Data y Data Science

Aplicaciones al comercio, empresa y finanzas.

Curso 2020-2021

Índice

1. Introducción

Este trabajo se encuentra enmarcado dentro del Master de Big Data y Data Science, aplicaciones al comercio, empresa y finanzas de la Universidad Complutense. Durante Los últimos 12 meses hemos aprendido diferentes tipos de técnicas, lenguajes de programación que nos ayudan a utilizar y entender datos de manera masiva. Esta capacidad de tratamiento y comprensión de cantidades ingentes de datos suponen un avance considerable para cualquier industria que consiga utilizarlo de manera efectiva.

Precisamente una de las industrias donde más se puede utilizar esta capacidad en todos los aspectos de su operativa, es en la industria aeronáutica. Esta industria está compuesta por distintos Stakeholders, como pueden ser los fabricantes (Airbus, Boeing, Embraer etc), las aerolíneas y las empresas de servicios auxiliares (aeropuertos, handling etc.) Todo Este entorno está caracterizado por una altísima competencia así como por márgenes de beneficio muy reducidos. Esto lo convierte en el caldo de cultivo perfecto para que un arma tan útil como son los datos, se convierta en el caballo de batalla que muchas de estas compañías utilicen.

En nuestro caso, vamos a tratar el lado de las aerolíneas. Este mercado, como hemos comentado, presenta unos márgenes de beneficio extradamente reducidos. Por ello, las aerolíneas hacen uso de los datos en todos los procesos posibles. Desde la predicción de la demanda para fijar precios hasta la compra de combustible, pasando por el mantenimiento de sus aeronaves.

Por otra parte, gran parte de los ingresos de las aerolineas provienen de los viajeros de negocios, que suelen tener unas preferencias inelásticas con respecto a los precios (la decisión de compra no suele verse afectada por el precio). Sin embargo, algo que si valora este tipo de pasajero, es la puntualidad. Para ello, planteamos la prestación de un nuevo servicio, acoplado a los servicios de reserva, que, usando algoritmos de Machine Learning, prediga la posibilidad de que un vuelo pueda retrasarse. Esto puede utilizarse por dos partes:

- Aerolínea: Tener en cuenta qué vuelos suelen retrasarse y aplicar ideas innovadoras para intentar reducir el impacto de dichos retrasos en los vuelos que más ingresos generan debido a pasajeros de negocios.
- Pasajeros: Como hemos dicho antes, los pasajeros de negocios suelen ser muy cuidadosos a la hora de elegir sus vuelos, ya que cuentan con un tiempo limitado para cumplir con sus obligaciones. Estos podrían utilizar, al igual que existen buscadores de vuelos por precio, incluir un feature que incluya un índice de puntualidad para ese vuelo.

Para ello vamos a analizar los datos publicados por el Bureau of Transportation Statistics, agencia perteneciente al gobierno de los Estados Unidos que publica datos históricos sobre los vuelos realizados por aerolíneas nacionales.

2.Contexto de negocio

La industria de la aviación comercial es un gran motor económico y desde su desregulación en los años 80, ha sufrido una gran expansión. Esta expansión la podemos medir con los RPKs (Revenue Passenger Kilometers) que son, en resumen, los kilómetros viajados por pasajeros de pago. Esta métrica ha mantenido un aumento medio de 4-5% anual desde 1985.

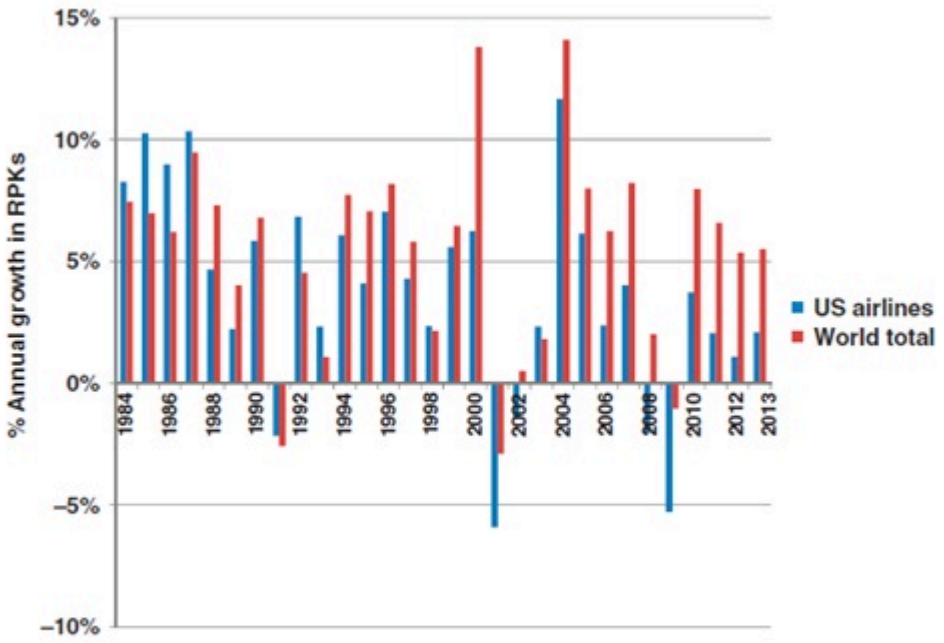
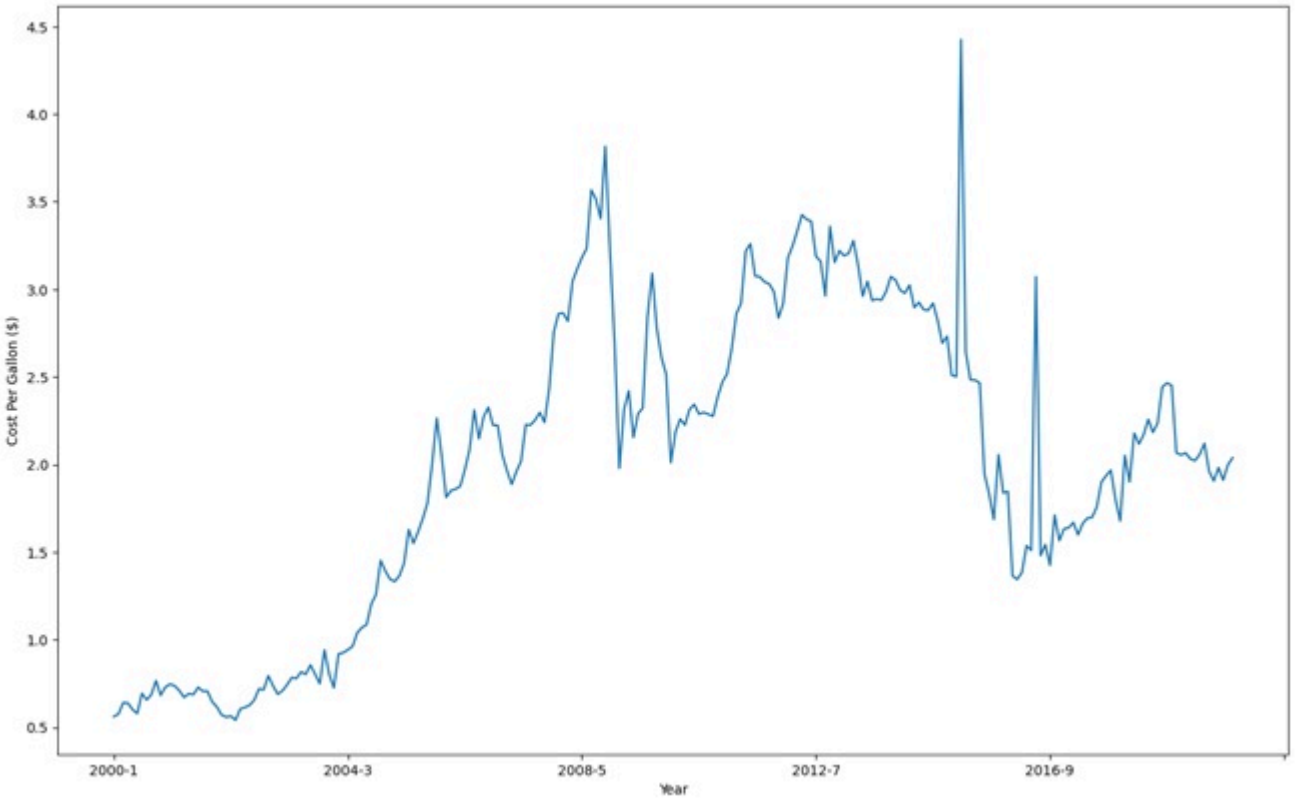


Figure 1.1 Annual RPK growth rates 1984–2013. (*Data sources:* Air Transport Association; ICAO)

Esta industria se caracteriza por una altísima competencia, márgenes de beneficio muy reducidos, alta volatilidad en los costes variables (especialmente el combustible) y la necesidad de alcanzar niveles muy altos de eficiencia en las operaciones.

Como hemos comentado, uno de los grandes problemas a los que se enfrenta esta industria, es el precio del combustible, que desde el año 2000 se han cuadruplicado, únicamente dando un respiro en la crisis del 2008.

Los picos que podemos observar en la serie, son debidos, entre otros factores a los conflictos entre los oligopolios productores de petróleo como la OPEP, los cuales para favorecer sus intereses, reducen la producción con el objetivo de hacer que los precios suban artificialmente.



Fuente: Elaboración propia (Datos: Bureau of Transportation Statistics)

3. Análisis Exploratorio de los datos

3.1 Análisis Principal

Contamos con un Dataset elaborado por el Bureau of Transportation Statistics que recoge datos de todos los vuelos operados en territorio estadounidense. En este Dataset se recoge información sobre el retraso sufrido por los vuelos, así como un desgloses de por qué ocurren dichos retrasos:

- Carrier Delay : Retrasos causados por la gestión de la aerolinea.
- Weather Delay : Retrasos causados por condiciones meteorológicas adversas.
- NAS Delay : Retrasos causados por el control aéreo (National Air System).
- Security Delay : Retrasos causados por causas de control de seguridad.
- Late Aircraft Delay : Retrasos causados por la tardanza del propio aeronave.

Los Campos que contine nuesto dataset son los siguientes:

- YEAR: Año del vuelo en cuestión.
- QUARTER: Trimestre del año en el que el vuelo se realizó.
- MONTH: Mes del año en el que el vuelo se realizó.
- Day_OF_WEEK: Día de la semana en que el vuelo se realizó.
- FL_DATE: Fecha completa en que el vuelo se realizó.
- OP_CARRIER: Código identificador de la aerolínea.
- ORIGIN_AIRPORT_ID: Código de identificación del aeropuerto de origen.
- ORIGIN: Nombre completo del aeropuerto de origen.
- DEST_AIRPORT_ID: Código de identificación de aeropuerto de destino.
- DEP_DELAY_NEW: Retraso en la salida en minutos.
- DEP_DELAY_GROUP: GRupos de retraso en la salida (Entre 15 y 180 minutos).
- ARR_DELAY_NEW: Retraso en la llegada en minutos.
- ARR_DELAY_GROUP: Grupos de retraso en la llegada (Entre 15 y 180 minutos).
- CANCELLED: Campo binario que indica si el vuelo ha sido cancelado.

- DIVERTED: Campo binario que indica si el vuelo ha sido desviado.
- AIR_TIME: Tiempo de vuelo en minutos.
- FLIGHTS: Número de vuelos.
- DISTANCE: Distancia recorida.
- DISTANCE_GROUP: Grupos de distancia.
- Carrier_Delay : Retrasos causados por la gestión de la aerolinea.
- Weather_Delay : Retrasos causados por condiciones meteorológicas adversas.
- NAS_Delay: Retrasos causados por el control aéreo (National Air System).
- Security_Delay: Retrasos causados por causas de control de seguridad.
- Late _Aircraft_Delay: Retrasos causados por la tardanza del propio aeronave.

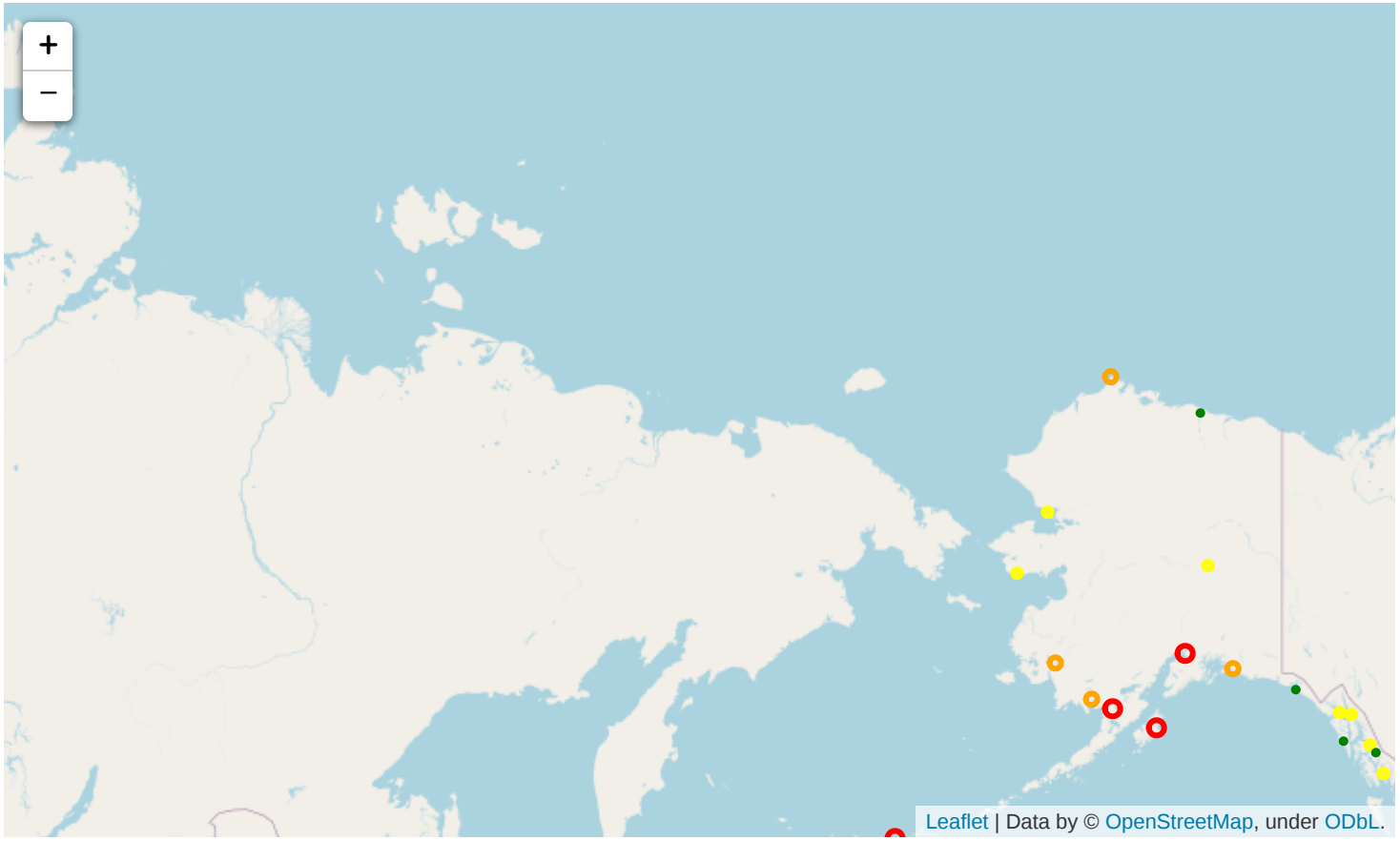
Para comenzar vamos a eliminar los vuelos que han sido cancelados o redirigidos a otro aeropuerto ya que no son relevantes para nuestro análisis, además de comprobar nuestro estado de valores missing en los datos.

	YEAR	QUARTER	MONTH	DAY_OF_WEEK	FL_DATE	OP_CARRIER	ORIGIN	DEST	DEP_TIME	DEP_DELAY_NE
Data Type	int64	int64	int64	int64	object	object	object	object	float64	float
Null values	0	0	0	0	0	0	0	0	119405	1194
Null Values (%)	0	0	0	0	0	0	0	0	1.76541	1.765

Observamos que tenemos una gran cantidad de datos con valores missing, tendremos que tener cuidado con dichos datos, ya que pueden no permitir el correcto funcionamiento de nuestro modelo. Los trataremos al final de este apartado.

Cargamos ahora un dataset con los datos geográficos de los aeropuertos que nuestro dataset cubre. Vamos a representarlos para visualizar espacialmente nuestro tráfico. Para calcular el tráfico que cada aeropuerto tiene, hacemos un join entre nuestros datos de vuelos y nuestro dataset geográfico de aeropuertos. Una vez que tenemos las frecuencias junto con los datos geográficos, los ploteamos en nuestro mapa de Folium.

' '



En el mapa podemos ver todos los aeropuertos caracterizados por su importancia dentro de la red:

- Verde : Vuelos < 1000
- Amarillo: 100 < Vuelos < 10.000
- Naranja: 10.000 < Vuelos < 100.000
- Rojo: Vuelos > 100.000

Podemos ver cómo existe una gran concentración de vuelos en la costa este, donde se encuentran las capitales financieras de los EEUU, como pueden ser Nueva York, Washington, Atlanta etc. También comprobamos que en sur existen grandes ciudades que sirven de aeropuertos de conexión como pueden ser Houston y Dallas. Además podemos comprobar la extensa red de conexiones que sirven las aerolineas norteamericanas.

Esta distribución es muy importante a la hora de entender el modelo de negocio de las aerolineas norteamericanas. Cada aerolínea tiene sus grandes centros, como Atlanta Hartsfield Jackson para Delta, Newark para United Airlines o John Fitzgerald Kennedy para American Airlines. Estas suelen tener subsidiarias regionales, que operan sus vuelos regionales y de corto radio dentro de Estados Unidos, como American Eagle para American Airlines. Estas operan los vuelos procedentes de pequeños aeropuertos y suponen gran parte del tráfico interno de los Estados Unidos. Posteriormente, todo ese tráfico es distribuído internacionalmente desde los grandes aeropuertos en la costa Este, o el Medio Oeste americano.

Por último, se hace muy evidente la cantidad de tráfico que produce Alaska, debido a su desconexión con el resto de la zona continental norteamericana.

Tenemos una cantidad muy grande de datos, por lo tanto, para realizar el análisis preliminar utilizaremos todos los datos del mes de Junio, pero para nuestro modelo de predicción utilizaremos solamente los vuelos de Delta Air Lines, la aerolínea más relevante en los Estados Unidos.

Para el análisis preliminar del mercado, vamos a utilizar los datos de Junio, uno de los meses con mayor tráfico.

3.2 Limpieza de Datos:

Vamos a comentar los tratamientos que hemos realizado a los datos:

- Hemos diseñado una función que convierta las horas de salida y llega a formato time para poder utilizarlas, ya que originalmente provienen en formato float y con caracteres que ensucian la usabilidad de los datos.
- Al comprobar la cantidad de missings presentes en los datos, hemos decidido eliminar las siguientes columnas:

```
* CARRIER_DELAY
* NAS_DELAY
* WEATHER_DELAY
* SECURITY_DELAY
* LATE_AIRCRAFT_DELAY
* Unnamed: 24
```

- Al haber eliminado al principio del tratamiento los vuelos cancelados y desviados, no va a a presentar el dataset gran cantidad de valores missings en nuestras variables principales, lo cual es gran punto positivo a tener en cuenta.

	Variable	Missing Values	% Missings
0	QUARTER	0	0.000000
1	MONTH	0	0.000000
2	DAY_OF_WEEK	0	0.000000
3	FL_DATE	0	0.000000
4	OP_CARRIER	0	0.000000
5	ORIGIN	0	0.000000
6	DEST	0	0.000000
7	DEP_TIME	0	0.000000
8	DEP_DELAY_NEW	0	0.000000
9	DEP_DELAY_GROUP	0	0.000000
10	ARR_TIME	0	0.000000
11	ARR_DELAY_NEW	0	0.000000
12	ARR_DELAY_GROUP	0	0.000000
13	CANCELLED	0	0.000000
14	DIVERTED	0	0.000000
15	AIR_TIME	0	0.000000
16	DISTANCE	0	0.000000
17	DISTANCE_GROUP	0	0.000000
18	CARRIER_DELAY	467529	75.300175
19	WEATHER_DELAY	467529	75.300175
20	NAS_DELAY	467529	75.300175
21	SECURITY_DELAY	467529	75.300175
22	LATE_AIRCRAFT_DELAY	467529	75.300175
23	Unnamed: 24	620887	100.000000

Las columnas que nos explican por qué se producen los retrasos tienen una cantidad de missings que no es admisible, así que las eliminamos. El resto de columnas no cuentan con valores missing por lo que las dejamos como

están.

3.3 Eliminación de Outliers

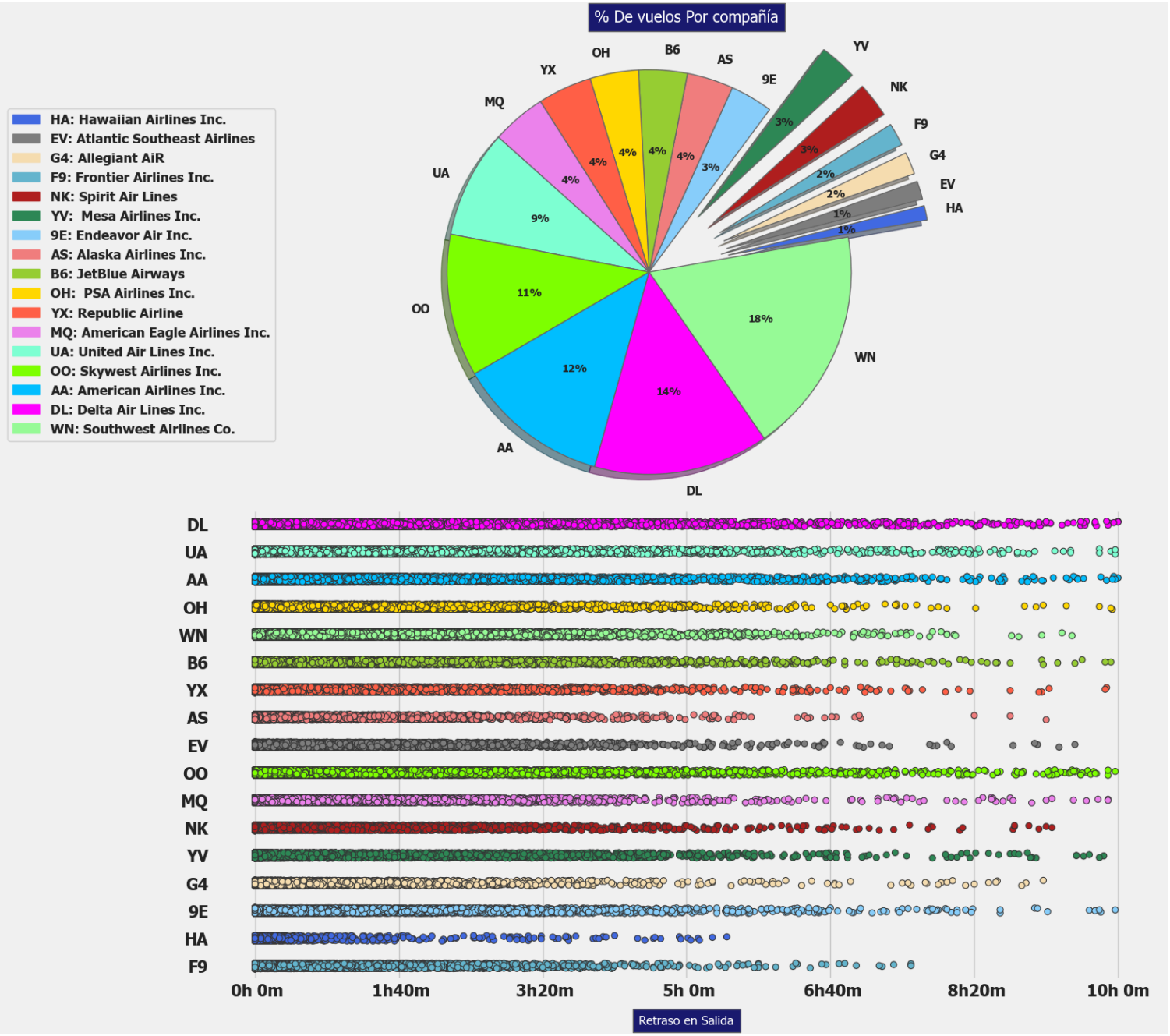
Vamos a eliminar los vuelos que tengan un retraso mayor de 10 horas, ya que los vamos a considerar un outlier. El tratamiento por rangos intercuartílicos no ha sido posible ya que la distribución está muy sesgada hacia un lado. Por ello, hemos elegido nosotros tomar una decisión basada en el negocio. Un vuelo con más de 10 horas de retraso, es considerado una rareza. Por ello procedemos de esta manera.

3.4 Comparación de Aerolineas

' '

	mean	min	max	count
OP_CARRIER				
HA	5.230232	0.0	328.0	7158.0
EV	28.518637	0.0	570.0	9068.0
G4	16.730464	0.0	548.0	10967.0
F9	23.932343	0.0	456.0	11307.0
NK	18.485845	0.0	554.0	17485.0
YV	23.057516	0.0	590.0	18760.0
9E	22.273487	0.0	598.0	21182.0
AS	9.468382	0.0	550.0	23009.0
B6	24.334247	0.0	595.0	24066.0
OH	16.542498	0.0	596.0	24072.0
YX	16.343689	0.0	592.0	26812.0
MQ	16.972572	0.0	593.0	26980.0
UA	21.132060	0.0	598.0	53362.0
OO	16.903160	0.0	598.0	70725.0
AA	19.847306	0.0	600.0	75956.0
DL	13.794198	0.0	600.0	86763.0
WN	15.330553	0.0	568.0	112463.0

Los datos nos confirman algo que ya sospechábamos, hay una grandísima disparidad entre la cantidad de vuelos realizados por unas aerolíneas y otras. Pero, además de esto, observamos cómo existe también una gran disparidad en la media de retraso en los vuelos. Con un rango de medias desde 5 a 28 minutos, queda claro que tener en cuenta la aerolínea a la que pertenece cada vuelo nos sería muy útil a la hora de tener un modelo que nos predijera a gran escala el retraso de dichos vuelos.

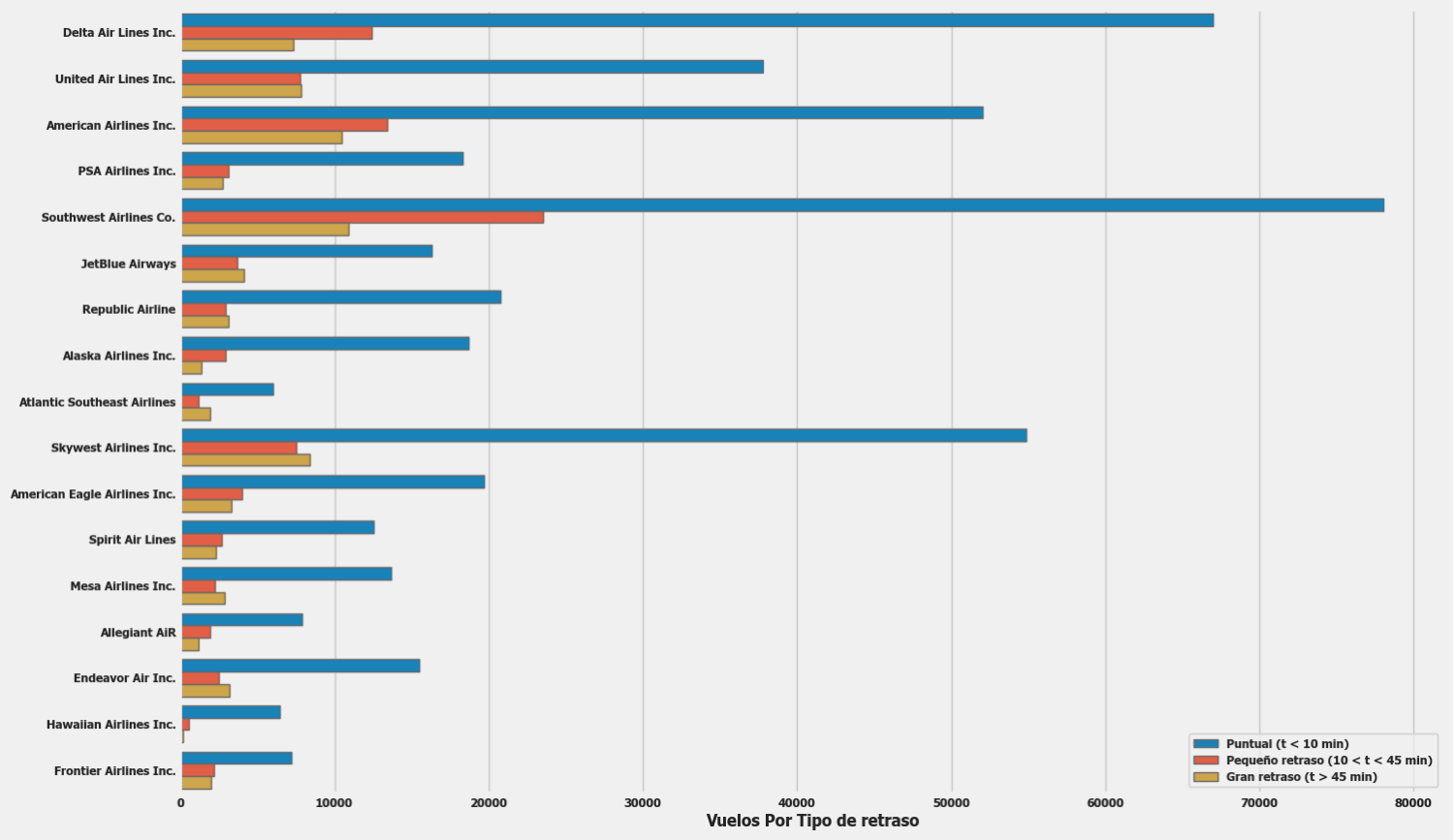


En estos gráficos podemos observar ciertas características del mercado de aerolíneas estadounidense. Para comenzar, existe una gran concentración del tráfico en unas pocas aerolineas, como pueden ser Delta, American Airlines y Southwest.

Sin embargo, en cuanto a los retrasos, vemos que la distribución es mucho más simétrica, auqnue se puede apreciar que las aerolineas con menos tráfico suelen tener concentraciones más cercanas a cero. Esto puede explicarse debido a distintos factores:

- Las aerolineas con concentraciones de retrasos más bajas suelen coincidir con aerolineas regionales, con vuelos más cortos, que dan servicio a aeropuertos más pequeños en los que los retrasos suelen ser menores.
- Las aerolineas de gran tamaño suelen tener procedimientos y necesidades más complejas que las aerolineas regionales o de mediano tamaño (tripulaciones más grandes, catering, mantenimiento etc.).

A continuación vamos a agrupar los retrasos y a ver que tipo de comportamiento tienen las aerolíneas.



En este caso comprobamos que existe grandes diferencias entre las cantidades de vuelos operados por cada aerolinea, así como una gran disparidad en los porcentajes que suponen cada uno de los grupos de retraso dentro del total de vuelos. Pongamos un ejemplo paradigmático: Southwest Airlines.

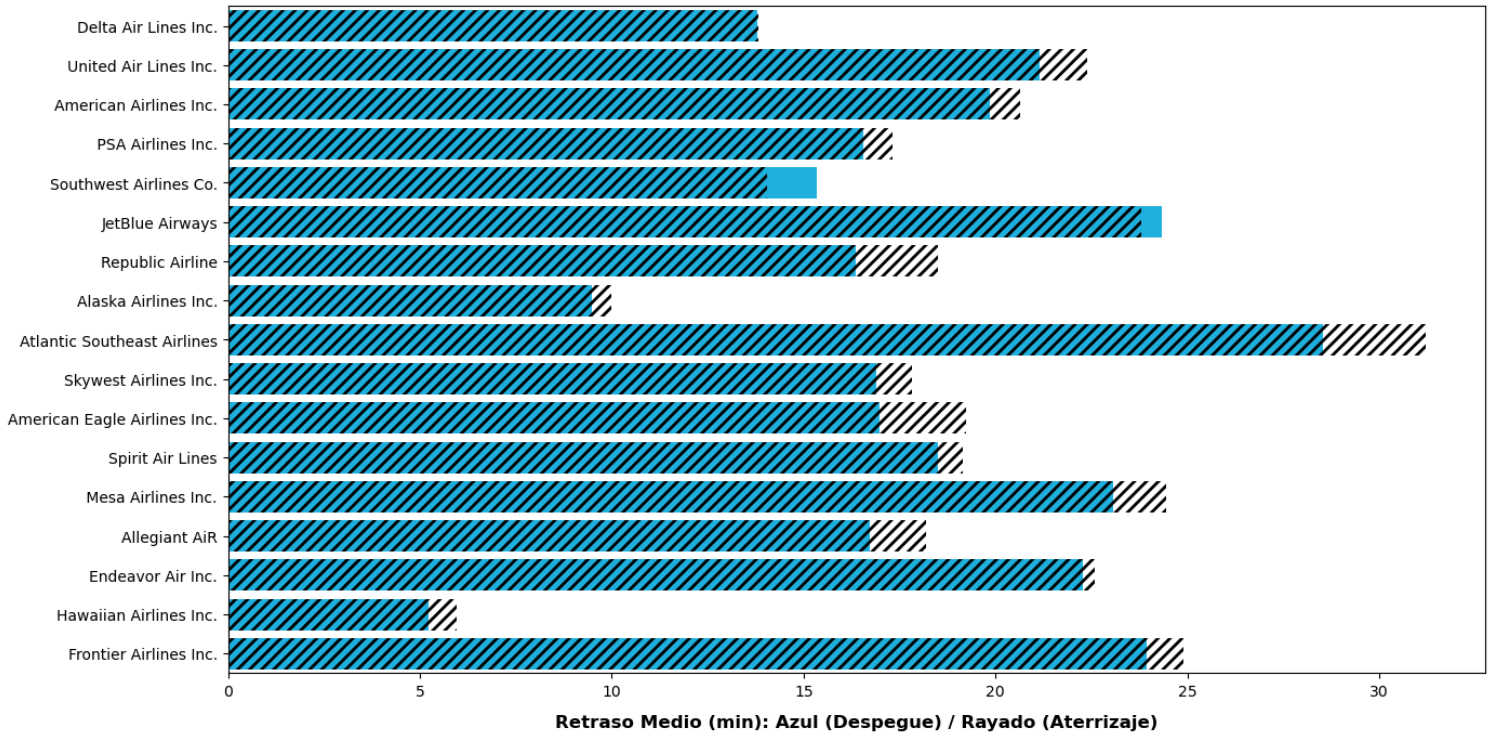
Southwest Airlines es la compañía pionera en Low Cost en Estados Unidos, su homóloga europea sería Ryanair. Southwest cuenta con una masiva flota de unos 750 aviones. Como sabemos, existen grandes diferencias entre las compañías tradicionales y las de bajo coste. Una de las principales diferencias con las aerolíneas tradicionales, es la cantidad de vuelos que un solo avión puede operar a lo largo del día. Normalmente, una aerolínea de bajo coste opera sus primeros vuelos a las 6 de la mañana y deja de operar a medianoche. Por lo tanto, un solo avión puede realizar hasta 8 ciclos (vuelos) en un solo día. Por lo tanto, un sólo retraso en uno de esos vuelos, tiene un efecto cascada con el resto de vuelos que esa aeronave debe operar durante el día.

Precisamente por este motivo podemos observar cómo el número de vuelos con un retraso medio de Southwest, es el más alto con diferencia (incluso comparando con aerolíneas con similar nivel de vuelos).

3.5 ¿Mejoran las aerolineas sus retrasos en el aterrizaje?

En ocasiones, aunque los vuelos se retrasen, ese retraso puede ser compensado una vez en el aire, gracias a las corrientes favorables de aire, menor congestión de la esperada o directamente una gran eficiencia en las operaciones. Por ello, queremos ver si esto es un factor relevante a la hora de saber cual es el retraso final de un vuelo.

```
Text(0.5, 0, 'Retraso Medio (min): Azul (Despegue) / Rayado (Aterrizaje)')
```



Podemos observar como por lo general, el comportamiento de los vuelos en el aterrizaje es igual o peor que en el despegue. Esto tiene sentido, ya que los retrasos se acumulan de manera lineal, es decir, cuando un vuelo se retrasa en origen, lo lógico es que esos retrasos causen más retrasos en el camino por motivos de tráfico o en los aeropuertos de llegada por exceso de tráfico de llegada. Descartamos así que los retrasos al aterrizar sean un factor a tener en cuenta.

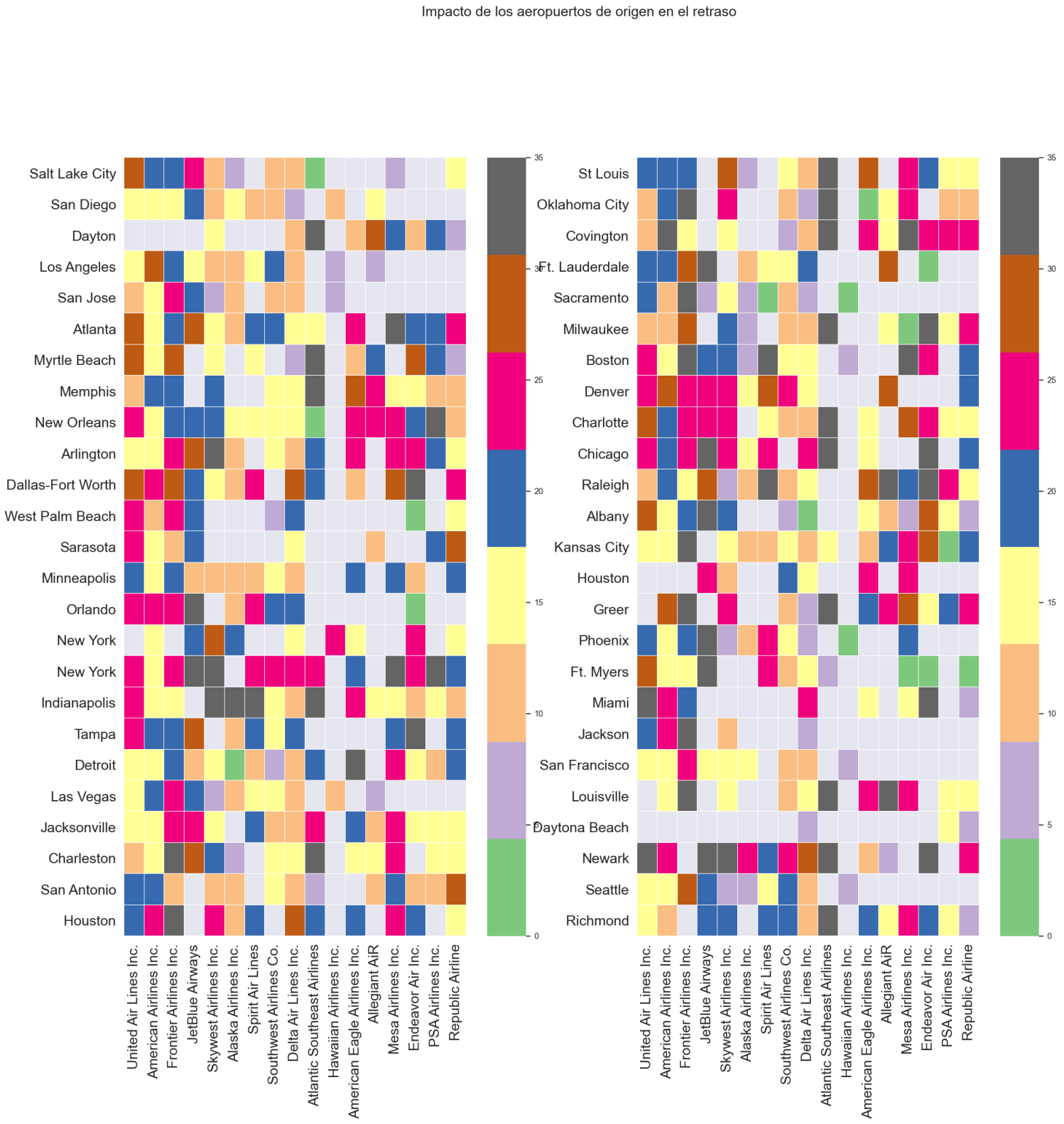
En nuestro modelo de Machine Learning, por lo tanto, tendremos en cuenta el retraso al despegue.

3.6 ¿Tienen los aeropuertos de Origen impacto en los retrasos?

Otro factor a tener en cuenta, es si los aeropuertos de origen pueden tener un impacto en qué vuelos muestran retrasos o no, para ello, estudiaremos los retrasos medios por aerolínea y aeropuerto de salida.

```
''
```

```
Text(0.5, 1.02, 'Impacto de los aeropuertos de origen en el retraso')
```

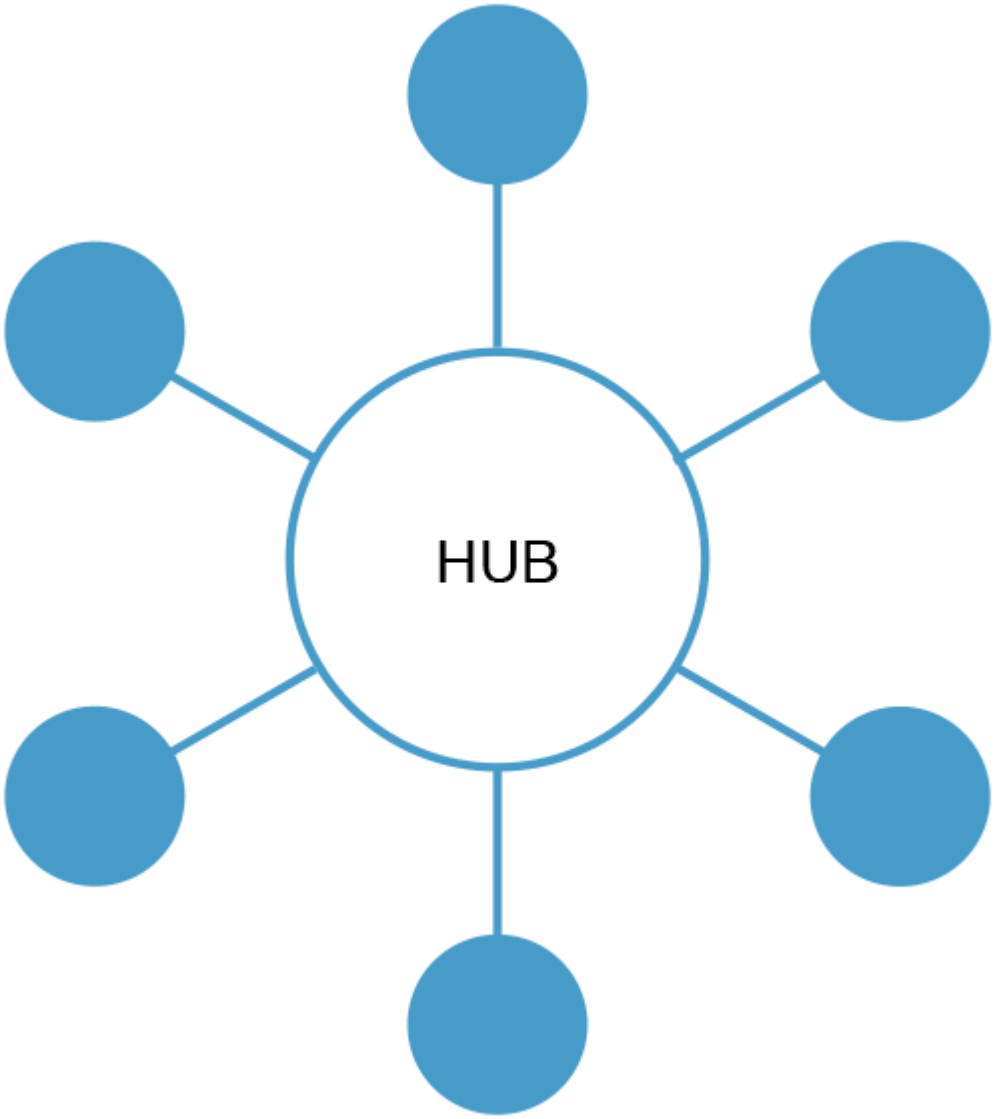


Hemos escogido un subset pequeño de aeropuertos, ya que en el dataset se incluyen más de 30 destinos dentro de los Estados Unidos. Podemos comprobar que los aeropuertos centrales, Como Newark (New York City), Chicago, New York, Dallas - Fort Worth, Indianapolis etc. concentran más altos niveles de retraso de media que los demás. Esto es relativamente evidente, ya que los grandes aeropuertos que sufren de gran congestión de tráfico tienden a sufrir mayores retrasos en cadena. Es decir, cuando un eslabón falla, por ejemplo un avión sufre una avería, es más fácil que esto afecte al resto del tráfico en un aeropuerto grande que en un aeropuerto pequeño.

También podemos darnos cuenta de que cada aerolinea sufre grandes retrasos en sus aeropuertos centrales o "Hubs".

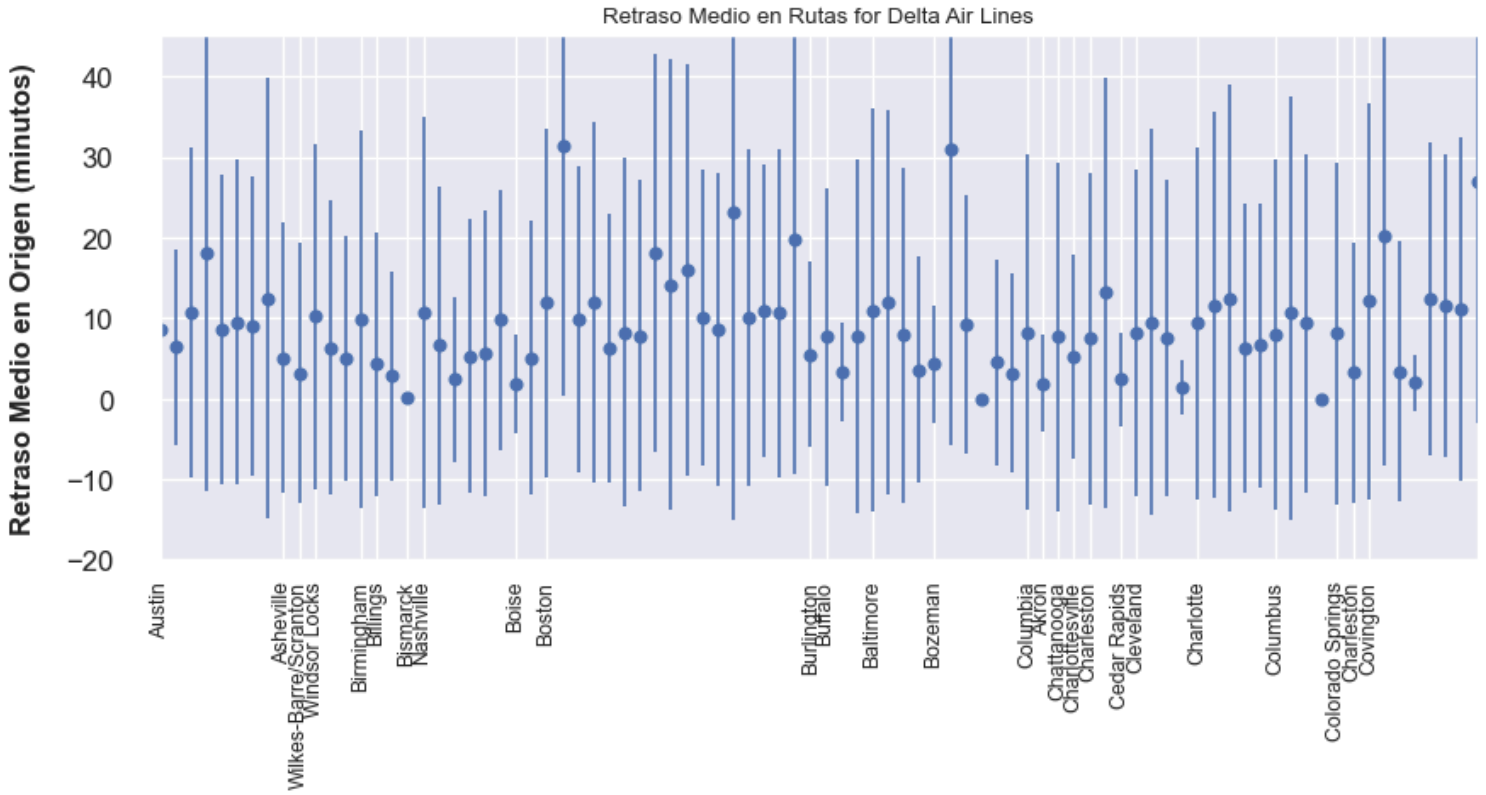
- Delta -> Atlanta
- United -> Newark
- American Airlines -> New York
- Jet Blue -> New York

Esto ocurre principalmente para las llamadas "legacy carriers" o compañías aéreas tradicionales (lo que en España vendrían a ser Iberia o Air Europa), ya que dichas aerolineas utilizan el denominado "Hub and Spoke Model". Este modelo consiste en que las aerolineas cuentan con uno (o más) aeropuertos principales a los que dirige todo el tráfico regional, y desde los que opera los vuelos de largo radio e internacionales. Esto les permite concentrar sus operaciones, tener gran parte de sus trabajadores y mantenimiento en un lugar y utilizar las economías de escala que les otorga ser operadores mayoritarios dentro de un aeropuerto.



3.7 ¿Hay rutas que siempre lleguen tarde o siempre pronto?

Como dijimos, vamos a utilizar los vuelos de Delta Air Lines como ejemplo, en este caso vamos a usarlos para analizar las rutas, para comprobar si existe alguna ruta que sistemáticamente lleguen tarde o a tiempo.



Podemos observar que existe una gran variabilidad entre las rutas, tenemos rutas con medias de cero minutos de retraso, mientras que otras tienen una media por encima de 30 minutos. Esto nos deja claro, que tanto el origen como el destino van a ser variables relevantes a la hora de crear nuestro modelo de predicción de retraso.

4. Predicción de retraso en vuelos

A continuación vamos a preparar los datos, para introducirlos en nuestro modelo de machine learning. Comenzamos por eliminar las variables que no necesitamos y a tratar las que si nos vamos a quedar:

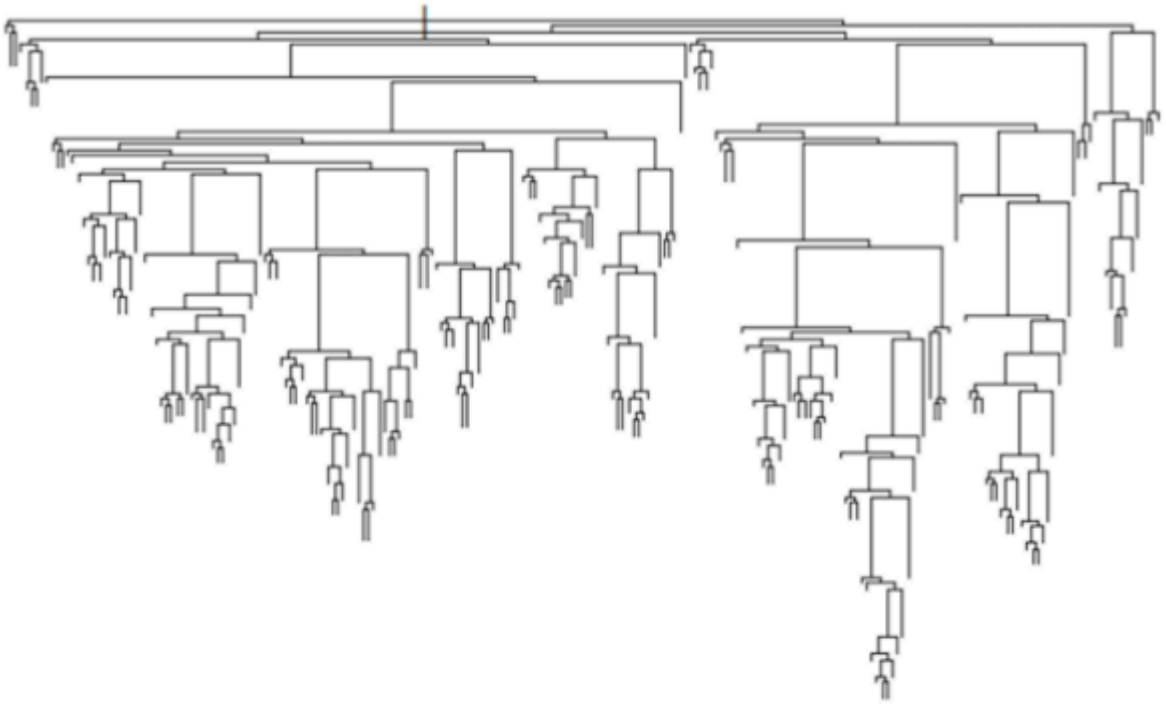
- Eliminamos las variables de mes, trimestre, aerolinea y las que nos indican si los vuelos han sido cancelados o desviados, ya que las hemos filtrado ya.
- Utilizamos las variables Arrival time y Departure Time para agruparlas e indicarle al modelo las horas orientativas en las que ese vuelo ha tomado lugar. Tendremos grupos de hora de llegada y de salida ya que los vuelos pueden llegar a durar 7 u 8 horas, información que queremos que el modelo tenga.
- Utilizaremos One Hot Encoding para los días de la semana, ya que esto no genera demasiadas variables dummy, y no añade una excesiva distorsión a nuestro modelo.
- Utilizaremos Label Encoding para nuestros aeropuertos de destino y origen, ya que contamos con hasta 200 valores únicos. Además probaremos modelos incluyendo solo los aeropuertos de origen o de destino para comprobar la relevancia que pueden tener para el modelo.

4.1 One Hot Encoding

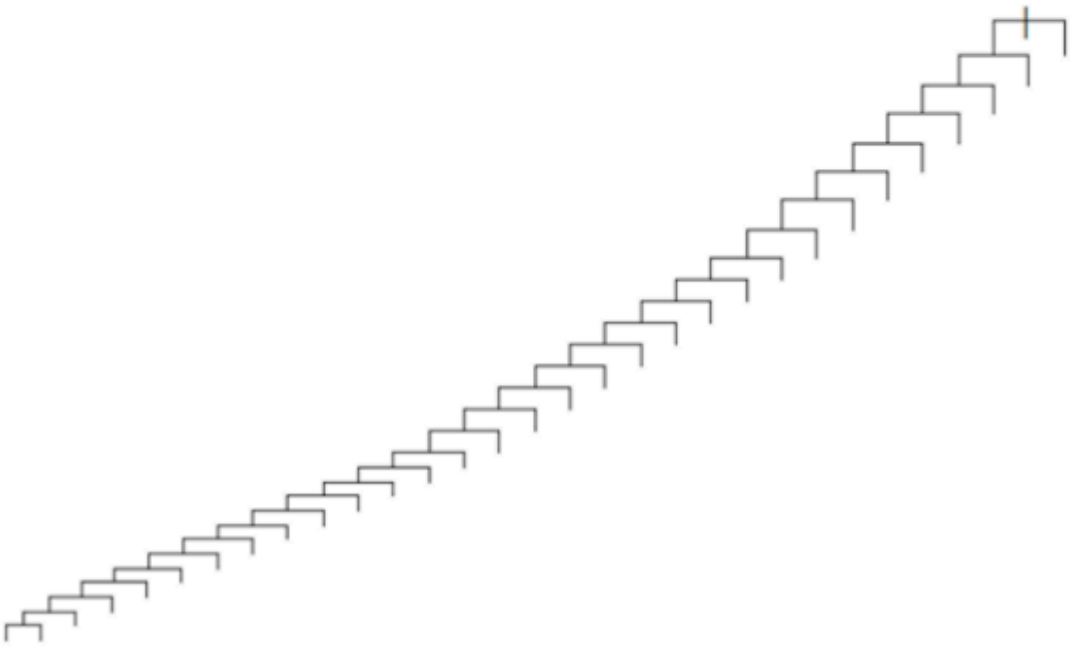
Las variables categóricas con pocos niveles (como los días de la semana) podríamos categorizarlas con OneHotEncoding ya que tienen pocos niveles y no nos añadiría mucha sparsity al modelo. Sin embargo, después de comprobar los modelos, con ambas opciones, observamos que manteniendo la variable días de la semana observamos que los modelos sin One Hot Encodign obtienen mejores puntuaciones.

Cuando a un modelo de Random Forest o ensemble como los que vamos a utilizar, le damos variables dummie como las que vamos a crear con One Hot Encoding, se crea un problema, y es que el modelo sólo tiene dos valores por los

que separar a través de esa variable los árboles de decisión. (Adjuntamos imagen para visualizar mejor el problema).



Dense Decision Tree (Model without One Hot Encoding)



Sparse Decision Tree (Model with One Hot Encoding)

Fuente: One-Hot Encoding is making your Tree-Based Ensembles worse, here's why?

4.2 Label Encoding

Para transformar nuestras variables categóricas que hemos visto que son relevantes para nuestro modelo (como los aeropuertos de llegada y salida), vamos a utilizar la técnica de Label Encoding con la función incluída de sklearn LabelEncoder. Esto proporcionará al modelo de una información que hemos visto es muy relevante a la hora de considerar el retraso que un vuelo va a tener.

	ORIGIN	DEST	AIR_TIME	DISTANCE	DISTANCE_GROUP	Delay_group	ARR_TIME_GROUP	DEP_TIME_GROUP
2997647	315	185	55.0	368.0	2	0	6	6
2997648	295	177	294.0	2446.0	10	0	7	5
2997649	89	20	69.0	432.0	2	0	2	1
2997650	187	177	292.0	2475.0	10	0	7	4
2997651	312	315	88.0	584.0	3	1	3	1

4.3 Variable objetivo y Train Test Split

Por último, hacemos nuestro Train-Test Split para tener nuestros datos de entrenamiento y de validación, y comprobamos que las dimensiones son las correctas.

```
Las dimensiones de las variables de entrenamiento es: (496108, 7)
Las dimensiones de la variable objetivo de entrenamiento es: (496108,)
Las dimensiones de las variables de validación es: (124027, 7)
Las dimensiones de las variable objetivo de entrenamiento es: (124027,)
```

4.4 Modelos

Vamos a utilizar 3 modelos de clasificación multiclase implementados en scikitlearn, como son:

- Naive Bayes: Supone una relación lineal
- K-Nearest_Neighbors: Es muy pesado al calcular las distancias
- Random Forest Classifier: Creemos puede ser el mejor fit para nuestro modelo
- XGBoost Classifier

4.5 Naive Bayes Classifier

El clasificador Bayesiano es un modelo probabilístico basado en el teorema de Bayes, que incluye además hipótesis simplificadoras, como son la independencia de las variables que le aportamos. Otra ventaja de este modelo, es que se necesita muy poca cantidad de datos de entrenamiento para estimar los parámetros (media y varianza) necesarios para la clasificación.

Como desventajas podemos citar las siguientes:

- El cálculo asume que los datos se distribuyen normalmente, algo que no tiene por que cumplirse en todas las ocasiones.
- Los predictores se consideran independientes entre si, suposición que no tiene por qué ser cierta.

```
La métrica Accuracy del modelo Naive Bayes Gaussiano es: 0.73

              precision    recall  f1-score   support

           0      0.749     0.974     0.846     90513
           1      0.263     0.019     0.035     18752
           2      0.415     0.139     0.209     14762
```

```
    accuracy                           0.730    124027
   macro avg       0.475     0.377     0.363    124027
weighted avg       0.635     0.730     0.648    124027
```

4.6 K Neighbors Classifier

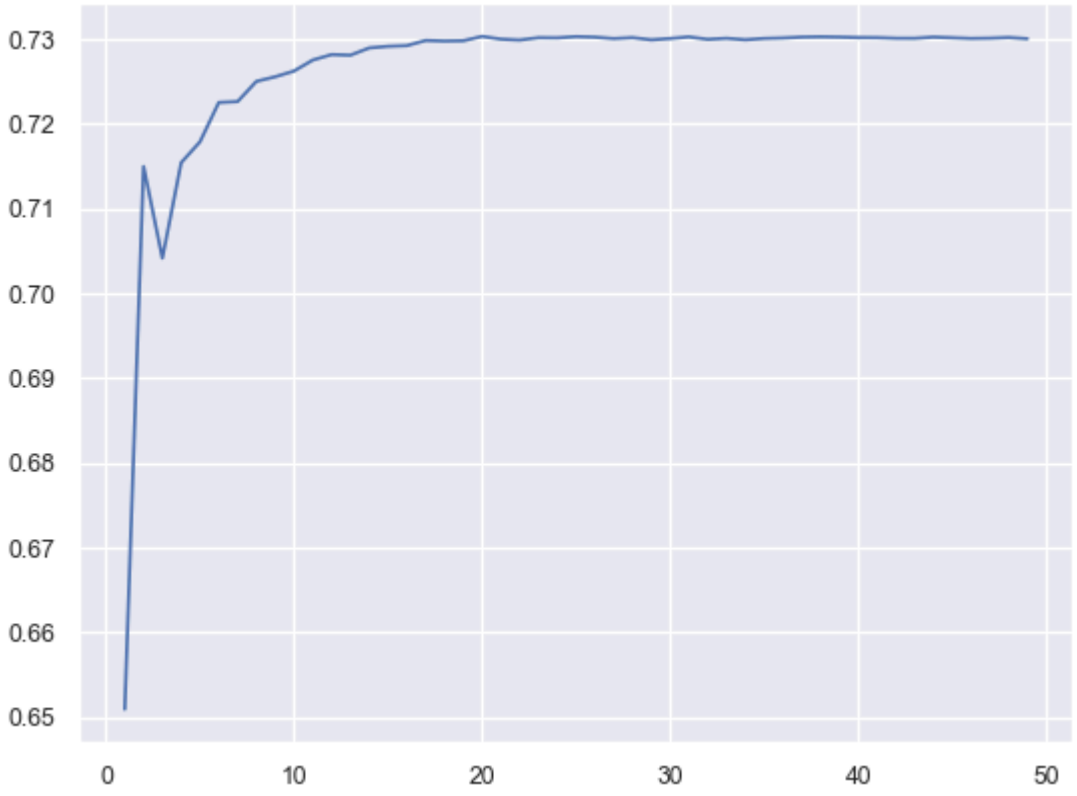
Método de clasificación supervisada que estima las funciones de densidad de las variables predictoras por cada clase. Es un método no paramétrico que no hace ninguna suposición sobre la distribución de las variables predictoras. Sus ventajas son las siguientes:

- No paramétrico.
- Simple y fácil de interpretar.
- Alta precisión.
- No es sensible a los valores atípicos.

Sus desventajas son:

- No crea un modelo, si no que a la hora de la predicción utiliza las instancias de entrenamiento para darnos la predicción.
- Alto coste computacional.
- Alto requisito de memoria.

```
[<matplotlib.lines.Line2D at 0x23d1c87b3d0>]
```




```
La precisión del modelo con k = 25 es : 0.7303
```

Podemos ver que con k = 25 se consigue la mejor precisión del modelo.

```
''
```

```
              precision    recall  f1-score   support

           0       0.749     0.974     0.846     90513
           1       0.263     0.019     0.035     18752
           2       0.415     0.139     0.209     14762
```

```
    accuracy                           0.730    124027
   macro avg      0.475     0.377     0.363    124027
weighted avg      0.635     0.730     0.648    124027
```

4.7 Random Forest Classifier

Random Forest Classifier es una combinación de árboles predictores. Este tipo de modelo presenta muchas ventajas:

- Es uno de los algoritmos de ML más certeros con grandes datasets.
- Corren eficientemente en bases de datos grandes.
- No es necesaria la normalización de los datos, ni que las variables sean del mismo tipo.
- Nos permite estimar qué variables son importantes en la clasificación.

Sin embargo, estos modelos tienen a presentar overfitting en ciertos datos en tareas de clasificación y presenta las siguientes desventajas:

- A diferencia de los árboles de decisión básicos, la clasificación de los modelos Random Forest es complicada de interpretar.
- En los datos que incluyen variables categóricas con muchos niveles, este algoritmo se vuelca a favor de estos atributos con muchos niveles.
- Si existen atributos correlacionados con relevancia similar para el rendimiento del modelo, los grupos pequeños son más favorecidos que los grandes.

```
El modelo básico nos da una accuracy de: 0.711

              precision    recall  f1-score   support

           0      0.800     0.874     0.835     90513
           1      0.277     0.205     0.236     18752
           2      0.467     0.355     0.403     14762

    accuracy                           0.711    124027
   macro avg      0.514     0.478     0.491    124027
weighted avg      0.681     0.711     0.693    124027


GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=123),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [10, 20, 30, 40, 50],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'n_estimators': [100, 200, 500]})

El mejor criterio para medir la separación de los árboles de decisión es: entropy
La mejor profundidad para un arbol de decisión es: 20
El mejor método para medir el número de variables a tener en cuenta a la hora de separar los ár
boles es: auto
El mejor número de estimadores de los árboles de decisión es: 500

La Accuracy de nuestro modelo mejorado es: 0.7300829658058327

              precision    recall  f1-score   support

           0      0.769     0.974     0.860     90513
           1      0.378     0.072     0.120     18752
           2      0.643     0.254     0.364     14762

    accuracy                           0.752    124027
   macro avg      0.597     0.433     0.448    124027
weighted avg      0.695     0.752     0.689    124027
```

4.8 XGBoost Classifier

XGBoost es un modelo de combinación de árboles de decisión que usa un elemento de gradient boosting. Los principales beneficios por los que se diferencia con respecto al resto de algoritmos son:

- Amplio rango de aplicaciones: Regresión, clasficación etc.
- Disponible y ejecutable en todo tipo de plataformas.
- Es soportado por todos los princiales lenguajes de programación como C++, Java, R, Scala y evidentemente Python.
- Optimización del algoritmo de Gradient Boosting con regularización, procesamiento parelelo, detección de secciones irrelevantes de los árboles de decisión etc.

Sin embargo, este algoritmo presenta las siguientes contraindicaciones:

- Dificil Interpretación.
- Posible aparición de overfitting si no usamos los parámetros correctos.
- Mas complejo de afinar al contar con mayor número de hiperparámetros.

```
[12:49:02] WARNING: D:\bld\xgboost-split_1631904903843\work\src\learner.cc:1095: Starting in XG
Boost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed
from 'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavi
or.
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=8, num_parallel_tree=1,
              objective='multi:softprob', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)

La Accuracy del modelo con XGBClassifier es: 0.7493
              precision    recall  f1-score   support

           0      0.756     0.988     0.857     90513
           1      0.464     0.018     0.034     18752
           2      0.629     0.214     0.319     14762

    accuracy                          0.749    124027
   macro avg      0.617     0.407     0.403    124027
weighted avg      0.697     0.749     0.668    124027
```

4.9 Modelo Elegido

Observamos que nuestro modelo de XGBoost es el que obtiene los mejores resultados con nuestro dataset. A través de la búsqueda en Grid de hiperparámetros con Random Forest conseguimos mejorar nuestro modelo inicial, que llega a superar por muy poco al modelo de XGBoost, aunque en comparación con este, hace un mejor trabajo a la hora de predecir las clases menos representadas en el dataset.

También es necesario comentar, que nuestros modelos de XGBoost han tardado mucho menos tiempo en converger que los modelos de Random Forest, por lo que lo hace una elección más ideal a la hora de productivizar el modelo y convertirlo en una herramienta como la que hemos planteado al inicio de este trabajo.

Por lo tanto, para un entorno en el que la velocidad de predicción fuera importante, nos decantaríamos por el Random Forest. Sin embargo, para un entorno de producción en el que al velocidad es la clave, utilizaríamos nuestro modelo de XGBoost.

5. Conclusiones

Obtenemos una accuracy relativamente alta, pero sin embargo, cabe mucho camino por recorrer para mejorarlo. Podría ser útil añadir más variables, como la antigüedad de los aviones, indicadores de congestión el los aeropuertos de origen y destino, etc. Además, como hemos comprobado en apartados anteriores, entre las distintas aerolíneas existen diferencias muy grandes. Como en nuestro caso hemos contado sólo con datos de una aerolinea en particular, podemos asumir, que añadir la variable aerolínea podría ayudar a mejorar el modelo.

Por último, contamos con un dataset muy desbalanceado, ya que tenemos un 20-30% de vuelos con retraso y casi un 70% en la categoría sin retraso. Por lo tanto, sería necesaria también crear un ajuste para convertir el dataset en más balanceado y conseguir un ajuste mejor.