# Documento de diseño ejercicio Spark

Manuel Jesús Jiménez Navarro

Marzo 2020

## 1 Introducción

Con el objetivo de aclarar ciertos aspectos de diseño realizados durante la implementación del ejercicio de Spark Streaming, se ha decidido realizar un documento que especifique las decisiones tomadas y dar una vista general al flujo de trabajo.

## 2 Decisiones de implementación

El ejercicio proponía transformar/filtrar un único flujo de datos y que de este se obtuvieran 4 flujos de datos diferentes. A continuación se detallarán las decisiones más importantes de diseño tomadas en cada uno de estos flujos de datos.

### 2.1 Facturas erroneas

Se ha considerado como factura erronea aquellas lineas que presenten algún problema, es decir, una factura puede ser válida pero contener alguna linea que contenga algún error. Estas lineas son consideradas como erroneas si: no contienen 8 elementos separados por comas, algún valor está vacío o los datos no siguen el formato adecuado (controlados por expresiones regulares). En el caso de la descripción, existen casos donde el contenido se encuentra entre doble comillas y con los productos separados por comas. En este caso se ha considerado que las comas dentro de las dobles comillas no son válidas. Como salida se obtendrá la tupla recibida como entrada.

#### 2.1.1 Facturas completas

Las facturas completas se han calculado mediante la función mapWithState y un timeout. En el momento que una factura no recibe una nueva linea en el timeout asignado se devuelve el identificador de la factura y sus propiedades, en caso contrario devolverá nulo. Las propiedades que contiene no son las que originalmente tenía (Precio, unidades, descripción...), son aquellas propiedades que son posteriormente introducidas al modelo (Precio mínimo, máximo, suma

unidades...). Se ha considerado que como mejora en la eficiencia de memoria era preciso no almacenar todo el conjunto de lineas de factura hasta el timeout, si no calcular y/o actualizar el estado de la factura respecto a las propiedades que espera el modelo.

### 2.2 Cancelaciones

Las cancelaciones parten de las facturas erroneas, posteriormente aplica una ventana de 8 minutos y cuenta los elementos que hay en este. La salida se compondrá del número de cancelaciones en sí.

### 2.3 Anomalías Kmeans y BisectKmeans

Como entrada recibirán aquellas facturas completas, ya que como salida de las facturas completas se obtiene la entrada del modelo, se coge esta tal cual y se clasifica si es anómalo o no. Como salida se obtiene el identificador de la tupla erronea.

## 3 Flujo de trabajo

Como flujo de trabajo general se ha seguido el siguiente esquema. Partiendo de unos modelos y unos parámetros que identifican el mínimo y el máximo del conjunto de datos de entrenamiento de dichos modelos se realiza los siguientes pasos:

1. Se filtran las entradas válidas y no válidas.
2. Se divide por comas sin tener en cuenta las comas dentro de las dobles comillas.
3. Se obtienen las propiedades usadas por el modelo: UnitPrice, Time y Quantity.
4. Se escalan las propiedades respecto los parámetros antes mencionados.
5. Se aplica el mapWithState con un timeout de 40 segundos.
6. Se filtran aquellas facturas que no están completas.
7. Se calculan las facturas completadas Canceladas.
8. Se calculas las anomalías de los dos modelos de aquellas facturas no canceladas.