Práctica 4 PRPA

Ainhoa Dïaz Cabrera y Claudia Gómez Alonso

May 2023

1 Motivación

El propósito de esta práctica será plantear un problema en relación con un fichero de datos sobre el itinerario de bicicletas eléctricas del servicio de BiciMAD, que proporciona el ayuntamiento de Madrid. Utilizando Spark manejaremos el dataset de manera más cómoda, de modo que podremos ir realizando las acciones que consideremos necesarias para alcanzar el objetivo propuesto y solucionar el problema planteado. Los ficheros de datos disponibles reflejan el movimiento de las bicicletas mensualmente. La información que nos dan estos datasets se corresponde, entre otros aspectos, con las estaciones de salida (*idunplug_base*) y entrada de las bicis (*idplug_base*), el rango de edad de las personas que alquilan la bici (*ageRange*), el código de usuario (*user_day_code*), el tiempo de utilización (*travel_time*).

2 Planteamiento del problema

Teniendo en cuenta los datos que se nos proporcionan, nos planteamos trabajar con los datos de entrada y salida de las bicicletas, y enfocar el problema en ello. Dado que tenemos a nuestra disposición numerosos ficheros de datos, uno para cada mes desde abril de 2017, tratamos de plantear un problema que utilice varios para ampliar la cantidad de datos empleados y, por tanto, conseguir resultados también más amplios. El estudio que vamos a realizar se centrará en las estaciones de donde salen las bicis y aquellas a las que entran. El problema planteado será contabilizar el número de bicicletas que salen y entran en cada estación, de modo que podremos observar aquellas estaciones de donde salen muchas bicicletas, y por tanto, hay una necesidad mayor de reponerlas. Análogamente comprobaremos a donde llegan la mayor parte de las bicis alquiladas para reorganizarlas y que no haya una sobrecarga en una estación. Realizaremos este estudio empleando datos a lo largo de todo un año, en nuestro caso 2018, y lo que queremos observar es la entrada y salida de las bicicletas agrupadas por épocas del año, de manera que queremos obtener los resultados según si los datos recogidos son de invierno, primavera, verano u otoño. Además, también nos planteamos, dados los datos a lo largo de todo el 2018, en qué epoca del año se utilizan mayor cantidad de bicicletas y por tanto, hace falta un mayor número de ellas disponibles para circulación.

3 Diseño e implementación de la solución

En primer lugar, para diseñar una solución correcta y comprobar que los pasos realizados están bien ejecutados más facilmente, lo que haremos será aplicar las diferentes acciones en un fichero de datos correspondiente a un mes concreto que servirá como modelo de prueba para luego poder aplicarlo a todos los meses del año. A continuación explicamos los pasos seguidos:

1. Importamos SparContext de la librería pyspark y llamamos *sc = SparkContext()*
2. Importamos json para poder leer los distintos ficheros de datos que tenemos.
3. Creamos nuestro primer rdd al que llamamos *rdd_base*, que contendrá los datos de la muestra que vamos a utilizar en forma de rdd al haber utilizado *sc.textFile*
4. Construimos la función *mapper* como hemos visto en los ejemplos realizados en clase, que nos servirá para acceder más fácilmente a los datos que nos interesan, en nuestro caso, del rdd. Aplicamos mapper a cada uno de los elementos del rdd con *rdd_base.map(mapper)*, y el resultado se denominará *rdd_util*.

5. Después filtraremos los datos para quedarnos solo con aquellos que pertenecen a *'user_type' = 1.* Para ello aplicamos *filter(lambda x: x[0] == 1)* a *rdd_util* y posteriormente transformaremos cada uno de los elementos en tuplas que contengan la estación de salida, la estación de entrada y el tiempo transcurrido, en este orden, que se corresponden con los datos a emplear en el estudio. Guardamos los datos en un nuevo rdd denominado *rdd_users.*

6. Para contar el número de bicicletas que salen de cada estación volvemos a realizar un map de manera que nos quedamos solo con el primer elemento de cada tupla y le aplicamos un *countByValue()* para calcular la cantidad de veces que aparece cada estación. Después procedemos análogamente con las estaciones de entrada, quedándonos esta vez con el segundo elemento de la tupla. En ambos casos transformamos lo obtenido en un rdd utilizando *sc.parallelize* y a cada rdd lo llamaremos *rdd_out* y *rdd_in* respectivamente.

7. Con el fin de hacer más sencilla la comprensión de los resultados, agruparemos las estaciones por zonas de 10 en 10, de modo que la zona 0 se corresponde con las estaciones del 0-9, la zona 1 con las estaciones de 10-19 y así sucesivamente. Para ello creamos la función llamada *agrupar_estaciones(par)* que tiene como parámetro de entrada cada una de las tuplas de los rdd anteriores y devuelve nuevamente una tupla en la que el primer elemento se corresponde que el número de la zona en vez de con la estación concreta. Por ejemplo, si tenemos como entrada (4,15), nos devolverá (0,15). Aplicamos esta función a cada uno de los elementos mediante un map.

8. Ahora tendremos un rdd con tuplas en la que la clave coincide en numerosos casos y nos interesa sumar sus valores, así que, realizamos un *reduceByKey(lambda x,y: x + y)* al *rdd_out.* El resultado se denominará *rdd_out_agrup.* Análogamente aplicamos todo lo descrito anteriormente a *rdd_in* y se llamará *rdd_in_agrup.*

9. Por último, ordenaremos cada uno de los rdds para poder visualizar más fácilmente aquellas zonas de las que salen, o a las que entran, más bicicletas. Para ello, aplicamos un *sortBy(lambda x : x[1], ascending = False).*

Tras seguir todos estos puntos, hemos conseguido dos rdds que nos informan sobre las zonas en las que hay más bicicletas de salida y de entrada. Así, podemos tratar de seguir el mismo procedimiento con todos los meses del año 2018, pero agrupados por épocas del año, para realizar, de este modo, el estudio que realmente nos interesa.

El siguiente paso que hemos de realizar, por tanto, es el diseño de nuestra solución, tratando de replicar el proceso recién explicado pero usando todos los ficheros de datos de todo un año. Para ello, haremos lo siguiente:

1. Lo primero que realizaremos será leer todos los ficheros de datos. Los agruparemos en cuatro listas que representan las cuatro estaciones del año, invierno, primavera, verano y otoño. A continuación transformamos las distintas listas en rdds mediante la función *trimestres(sc, lista_filename)* que nos devolverá los rdds agrupados de tres en tres meses, según se lo aplicamos a cada lista. El rdd final obtenido en cada caso, tendrá la misma forma que *rdd_users* del caso prueba explicado anteriormente. Por ejemplo, si metemos la lista invierno formada por los ficheros de datos correspondientes a los tres primeros meses del año obtendremos *rdd_inv* que será un rdd con las tuplas que tienen los datos de las estaciones de salida, de entrada y el tiempo de uso. Además aplicamos un filter para que aisle los casos el número de la estación de bicicletas se mayor que 300, puesto que hemos observado un dato anómalo que no nos interesa estudiar.

2. Luego crearemos las funciones *mas_bicis_salida(rdd_util)* y *mas_bicics_entrada(rdd_util).* Estas contienen las diferentes acciones de los pasos 6,7,8,9 del caso prueba explicado, concentradas dentro de una misma función (una para los datos de salida y otra para los de entrada, respectivamente) para poder aplicársela a cada uno de los cuatro rdds correspondientes con las distintas épocas del año.

3. Aplicamos las dos funciones nombradas en el paso anterior a cada uno de los cuatro rdds para obtener los resultados deseados y poder así realizar el estudio planteado inicialmente. Volviendo al ejemplo de *rdd_inv* obtendremos *salida_inv* y *entrada_inv* con las soluciones correspondientes.

Por último, ya que tenemos agrupados los datos por estaciones del año, no es muy complicado obtener el número exacto de bicicletas que se alquilan por temporadas, para así, poder observar en cuál de ellas hay mayor demanda y por tanto, mayor necesidad de número de bicicletas a disposición de circulación. Esto se realizará mediante un *reduce(lambda x,y: x + y)* sobre el segundo elemento de cada uno de los datos de los rdds finales de salida (como *salida_inv*).

También hemos querido comprobar, a modo de curiosidad, si hay algún extravío de alguna bicicleta comparando las bicicletas totales que salen de las distintas estaciones con las bicicletas que llegan de nuevo. Lo haremos realizando exactamente lo mismo que en el párrafo anterior pero sobre los rdds con los datos de entrada y después se comparará con los resultados anteriores.

4 Evaluación de resultados

En esta sección estudiaremos los resultados obtenidos por nuestro código. Tras llevar a cabo todos los pasos explicados anteriormente, obtenemos la solución a nuestro problema. Al final de la ejecución de todo el programa conseguimos una lista de rdds ordenados según la cantidad de bicicletas que han sido extraídas de cada zona de mayor a menor, así como la cantidad de bicicleta que llegan a las distintas zonas. Las listas obtenidas para cada una de las estaciones del año, contabilizando el número de bicicletas de salida son las siguientes:

Invierno	Primavera	Verano	Otoño
[(16, 55346),	[(16, 73685),	[(16, 87612),	[(16, 78305),
(5, 49003),	(5, 64230),	(5, 80581),	(5, 71127),
(4, 48081),	(4, 62163),	(4, 70174),	(4, 58787),
(13, 39592),	(13, 55205),	(13, 69334),	(13, 55428),
(0, 37466),	(0, 49792),	(0, 59755),	(1, 51704),
(7, 37034),	(1, 48994),	(1, 59483),	(0, 51244),
(1, 36513),	(7, 47661),	(9, 56425),	(7, 49092),
(9, 36141),	(9, 46376),	(7, 52718),	(9, 48697),
(12, 32883),	(8, 43975),	(8, 52255),	(8, 47091),
(8, 32683),	(12, 43876),	(12, 51469),	(12, 46785),
(3, 32433),	(6, 40483),	(6, 50354),	(6, 45189),
(11, 29996),	(3, 40445),	(11, 48783),	(3, 43370),
(6, 29952),	(11, 40361),	(3, 47710),	(14, 41577),
(14, 28561),	(15, 38034),	(14, 46801),	(11, 41335),
(10, 28465),	(14, 37818),	(15, 44496),	(15, 41251),
(15, 27774),	(10, 36746),	(10, 44264),	(10, 38212),
(2, 27540),	(17, 32027),	(17, 40087),	(17, 33826),
(17, 23737)]	(2, 28605)]	(2, 31734)]	(2, 27147)]

Como podemos observar en la tabla, en cada estación del año coincide que las bicicletas se cogen con mayor frecuencia en las zonas 16,5 y 13, por ejemplo. De este modo, podemos utilizar los resultados para saber cuáles son las zonas donde, cada mes, se deben proveer con mayor cantidad de bicis a las estaciones. De igual modo, se puede apreciar que las zonas de las que se extraen bicicletas con menos frecuencia también coinciden en muchos casos a lo largo de los meses, lo cual también puede ser útil para la redistribución de las bicis.

Luego, las listas obtenidas para cada una de las estaciones del año, contabilizando el número de bicicletas de entrada son las siguientes:

Invierno	Primavera	Verano	Otoño
[(16, 55486),	[(16, 73961),	[(16, 87007),	[(16, 78647),
(4, 49936),	(4, 64365),	(5, 80116),	(5, 71141),
(5, 48421),	(5, 63721),	(13, 78926),	(13, 63516),
(13, 45761),	(13, 63375),	(4, 73035),	(4, 61175),
(7, 37858),	(7, 48984),	(1, 58698),	(1, 50997),
(0, 35768),	(1, 48219),	(0, 57349),	(7, 50339),
(1, 35734),	(0, 48116),	(8, 56426),	(8, 50291),
(8, 35671),	(8, 47411),	(7, 54973),	(0, 49883),
(9, 35228),	(12, 45535),	(9, 53793),	(12, 48485),
(12, 35014),	(9, 44565),	(12, 53218),	(9, 46421),
(3, 31561),	(6, 40707),	(6, 50536),	(6, 45218),
(6, 30304),	(3, 39787),	(3, 47036),	(3, 42720),
(11, 28326),	(11, 37801),	(11, 45788),	(11, 39228),
(14, 26928),	(14, 35480),	(14, 44176),	(14, 38894),
(10, 26101),	(15, 34908),	(17, 41240),	(15, 38211),
(2, 25901),	(10, 34052),	(10, 41000),	(10, 34963),
(15, 25661),	(17, 32444),	(15, 40950),	(17, 34249),
(17, 23541)]	(2, 27045)]	(2, 29768)]	(2, 25789)]

De nuevo podemos ver que muchos de los datos coinciden de una estación del año a otra. Cabe destacar que se devuelven una gran cantidad de bicis a las zonas 16, 5 y 13, así que estas zonas recuperan muchas de las bicis que se extrayeron de ellas. Sin embargo, también aparecen nuevas zonas no consideradas anteriormente, como la 4, donde podemos deducir que habrán llegado una gran cantidad de bicis que estarán disponibles para poder distribuirse a otras zonas donde haya menos bicis al final de los meses.

Por otro lado, como ya se indicó, también podemos observar los resultados al sumar la cantidad total de bicis que se extraen cada época del año para ser capaces de averiguar en qué momentos es más popular el alquiler de bicis. Obtenermos la siguiente tupla, que nos indicará el número total de bicis que salieron de alguna de las estaciones de Madrid en invierno, primavera, verano y otoño, en ese orden: (633200, 830476, 994035, 870167). De aquí podemos concluir que es en verano cuando hay mayor demanda general de bicis (en 2018 se alquilaron 994035 bicicletas) y menor en invierno, como ya era de esperar.

Además, queríamos comprobar si había alguna bici que se perdía o extraviaba, para lo cual hacemos exactamente lo mismo y compramos el número de bicis total que llegan a las estaciones con los resultados que ya tenemos. Sorprendentemente, comprobamos que los datos coinciden, de modo que ninguna bici se pierde.

5 Conclusión

En conclusión, hemos utilizado una serie de técnicas vistas en clase, a través de las cuales hemos obtenido un código de python, empleando el entorno Spark, que nos permite estudiar un conjunto de datos. Tras plantear un problema, que en nuestro caso ha sido, observar el número de bicicletas que salen y llegan a cada estación, agrupadas de 10 en 10, y observadas por épocas del año. Así podemos hacer un estudio sobre en qué estaciones hay más número de salidas y si vuelven a llegar la misma cantidad o no, y por tanto, es necesario reponerlas. Y también, en qué época del año hay mayor uso de bicicletas y se necesitan, por ello, un mayor número en circulación. Además a modo de curiosidad, hemos observado si hay estravíos de bicicletas, haciendo un estudio del número que salen y el número que llegan. Hemos concluido que no ocurren desapariciones.