Memoria de la Práctica 4:

Manuel Pablo Bejarano Galeano, Miguel Caballero Rodríguez y Pedro Corral Ortiz-Coronado

1. Descripción del problema:

El **problema** que hemos querido tratar en la práctica parte de los datos que se encuentran en una base de datos (de formato variable: *.csv* o *.json*), de los cuales queremos extraer los resultados que nos pueden resultar interesantes y recogerlos todos en una lista para que se pueda consultar por cualquier persona, sin tener que leer toda la base de datos.

2. Descripción de las partes de la práctica:

La práctica se ha dividido en dos partes, caracterizados por las dos bases de datos que estábamos manejando:

En la primera base de datos (sample_10e4.json) encontramos información de uso de las bicicletas BiciMAD para unos ciertos días. Nos hemos centrado en obtener los datos del día con más tiempo de uso, la ruta más utilizada, el usuario que más viajes ha realizado en un mismo día, el tipo de usuario y el rango de edad que más ha usado las bicicletas y que menos han usado las mismas y la estación más y menos deficitaria. Estos resultados pueden ser de gran utilidad a la empresa competente, por ejemplo, para saber qué estación es la que en más riesgo está de quedarse sin bicicletas podemos transportar bicicletas de la menos deficitaria a la más deficitaria.

En la segunda base de datos (AccidentesBicicletas_2018.csv) se recoge la información de uso de los accidentes en bicicletas en Madrid en el año 2018 y nosotros hemos filtrado por: el tipo de accidente más común, la meteorología y el tipo de suelo en los cuales es más frecuente que ocurran los accidentes y la media de víctimas por accidente. De nuevo, esta información puede ser de gran interés para la empresa o incluso para el Ayuntamiento de Madrid para saber que tipo de suelos se deben mejorar para prevenir accidentes.

3. Implementación del problema:

En cuanto la **implementación** de los programas, hemos hecho uso de la librería pyspark, y como estructura de datos principal para tratar los datos hemos usado los RDD de dicha librería. Este tipo de datos tiene asociados numerosos métodos, de los cuales hemos usado los siguientes para nuestra implementación: map, reduce, reduceByKey, groupByKey, flatMap y join. Por ejemplo, para calcular la media de víctimas hemos utilizado .reduce(lambda x,y : ((x[0]*x[1]+y[0]*y[1])/(x[1]+y[1]),x[1]+y[1])) el cual nos devuelve la media ponderada de los elementos que componen un RDD en caso de que estos traten de números.

4. Conclusiones:

Las conclusiones que sacamos a partir de esta práctica giran en torno a las ventajas y posibilidades que nos ofrece tratar los datos con la estructura de los RDD de la librería pyspark, que aun habiendo trabajado con bases de datos pequeñas ya notamos la eficiencia gracias a su comportamiento perezoso y paralelo.

También ha sido muy útil el gran abanico de métodos que ofrece este tipo de datos, y la posibilidad de aplicarlos secuencialmente y de manera concatenada.

Finalmente, ha sido de gran uso que los RDD tienen son compatibles con múltiples fuentes de datos, véase los archivos json o csv que hemos usado como bases de datos, pueden ser leídos

sin problema y transformados en un RDD. Esto brinda flexibilidad en la elección de la fuente de datos y facilita la integración con diferentes tipos de almacenamiento.

Estas características hacen de RDD una opción poderosa para el procesamiento distribuido de datos en entornos de big data.