Práctica obligatoria

SPARK

UNIVERSIDAD
COMPLUTENSE
MADRID

Teresa Carrasco Pérez, María Garcia Gutierrez y Naia Daza Arteche
Facultad de Ciencias Matemáticas

Abril de 2022

Índice general

1 Motivación y explicación de la práctica

En esta páctica obligatoria de programación paralela hemos diseñado e implementado tres soluciones para tres problemas de análisis de datos utilizando Spark. El dataset sobre el que hemos trabajado es el que proporciona el ayuntamiento de Madrid del uso del sistema de bicicletas de préstamo BICIMAD. Los tres problemas planteados serán los siguientes, cada uno de ellos realizado en un archivo *.py*:

1. *Bicimad_frecuencia.py*. En este primer archivo hemos realizado un analisis de los 5 viajes o rutas más frecuentes y los 5 viajes menos frecuentes de un mes en función de los días de lunes a viernes o de sábado a domingo, es decir nos devolverá: top 5 viajes más frecuentes en un mes de lunes a viernes, top 5 viajes menos frecuentes en un mes de lunes a viernes, top 5 viajes más frecuentes en un mes en sábado o domingo y top 5 viajes menos frecuentes en un mes en sábado o domingo.

2. *Bicimad_por_meses.py*. En este segundo archivo hemos realizado un estudio de las rutas mas empleadas en función de cada hora y cada día de la semana (lunes, martes ...) pero también si se accede a varios ficheros de distintos meses, se obtiene esta clasificación en función de cada mes.

3. *Bicimad_sin_separar_por_meses.py*. En este último archivo, hemos realizado el mismo estudio de las rutas que en el archivo anterior pero si se accede a varios ficheros de meses no se hace distinción entre ellos, es decir, realizamos un análisis en su conjunto.

2 Implementación e instrucciones para ejecutar los programas

En primer lugar, detallaremos las funciones que hemos implementado en cada uno de los archivos anteriores. Posteriormente indicaremos una breve guía para llevar a cabo la ejecución de cada uno de ellos.

Funciones usadas en: *Bicimad_frecuencia.py*

- **datos**. Dicha función recibe como parámetros de entrada cada una de las líneas del fichero *.json* de entrada. Devuelve la tupla *(dia, estaciones)*, siendo estaciones a su vez una tupla formada por la estacion de origen y destino (las cuales están ordenadas por orden creciente de número de estación).

- **fin_de_semana**. Ésta nos devuelve el boolenao *False* si el día en cuestión es entre semana o bien *True* si se trata de sábado o domingo.

Funciones usadas en: *Bicimad_por_meses.py*

- **datos**. Recibe como parámetros de entrada cada una de las líneas del fichero *.json* de entrada. Devuelve la misma tupla que en el archivo anterior a excepción de que le añadimos en la primera componente de la tupla la hora y el mes *((dia, hora, mes), estaciones)* con el fin de realizar los análisis expuestos en el apartado 1.
- **repeticiones**. Esta función para cada semana y hora, ordena por los recorridos más utilizados y devuelve el número de veces que se ha realizado cada trayecto, recibiendo como entrada la tupla anteriormente descrita.
- **ordenar**. Recibe como parámetro de entrada las tuplas y llevamos a cabo una ordenación por hora de realización (comenzando a las 00.00h).
- **pasar_a_string**. Dicha función transforma las tuplas en cadenas de caracteres con el fin de poder imprimirlo en el fichero de salida.
- **meses**. Su implementación se realiza con el fin de que reciba como parámetro de entrada el número del mes pertinente y nos devuelva el nombre de dicho mes.

Funciones usadas en: *Bicimad_sin_separar_por_meses.py*

- **datos**. Recibe como parámetros de entrada cada una de las líneas del fichero *.json* de entrada. Devuelve la misma tupla que en el archivo anterior a excepción de que en este archivo ya no necesitamos el mes, luego la tupla resultante es: *((dia, hora), estaciones)* .
- La función **repeticiones**, así como **ordenar** y **pasar_a_string**; se implementan de la misma forma y con la misma finalidad que en el archivo anterior.

*Nota: Todas las ejecuciones que se llevan a cabo en el **main** correspondiente de cada archivo .py, se especifican en el apartado 1 de esta memoria.*

Instrucciones para ejecutar los programas. En lo que a ejecución respecta, en una primera instancia, el archivo *Bicimad_frecuencia.py* accede a un único fichero ***.json*** para realizar el análisis mencionado anteriormente. En cambio, tanto los ficheros *Bicimad_por_meses.py* como *Bicimad_sin_separar_por_meses.py* pueden acceder a varios ficheros de la misma índole que el anterior, pudiéndose limitar el análisis únicamente a un fichero si así el usuario lo desea.

3 Datos que hemos usado

1. *Bicimad_frecuencia.py*. Para este primer archivo hemos utilizado los datos del mes de enero de 2019.
2. *Bicimad_por_meses.py*. En este segundo archivo hemos utilizado los datos de enero y febrero de 2019.
3. *Bicimad_sin_separar_por_meses.py*. En este último archivo, hemos utilizado los datos de enero, febrero y marzo del mismo año.

4 Resultados y conclusiones

Resultados y conclusiones en: *Bicimad_frecuencia.py*

Podemos observar que el trayecto más realizado de lunes a viernes en enero de 2019 es de la estación 9 a la estación 149 un total de 321 veces. Sin embargo, de sábado a domingo es el bucle de la estación 58 esta vez un total de 100 veces.
También observamos que hay rutas que no se realizan ninguna vez a lo largo de la semana y por ello en este top 5 menos frecuente encontramos trayectos no realizados ninguna vez.

Funciones usadas en: *Bicimad_por_meses.py*

En este segundo archivo, hemos concluido que de lunes a viernes la franja horaria en la que más trayectos se realizan es de 8 am a 10 am. Sin embargo, esto cambia en el mes de febrero en el que observamos que la franja más concurrida estos días es de 5 pm a 8 pm. Tambien observamos que dichos trayectos más frecuentes tienen la misma estación de origen y destino.

Funciones usadas en: *Bicimad_sin_separar_por_meses.py*

En el último archivo vemos que la franja horaria más frecuentada de estos tres meses de 2019 es los lunes de 8 am a 9 am, martes encontramos dos horas puntas las 5 pm y las 10 am, los miércoles tenemos concurrida la franja de 5 pm a 7 pm. Por otro lado, la hora punta de los jueves es las 8 am y la de los viernes a las 4 pm.
Analizando ahora el fin de semana vemos que en ambos casos la hora punta es las 8 am.

Nota: Todas estas conclusiones generalizadas se pueden observar en los txt adjuntos.