UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMÁTICAS

PRÁCTICA OBLIGATORIA PROGRAMACIÓN PARALELA: BiciMAD

Autores:

José Ignacio Alba Rodríguez
Álvaro Ezquerro Pérez
Alejandro Millán Arribas

Programación Paralela

2022-2023

Índice

1. Introducción

Esta práctica consiste en realizar un estudio sobre la base de datos de la información de uso del servicio *BiciMAD* porporcionada por el Ayuntamiento de Madrid. El objetivo principal es aplicar los métodos de Spark aprendidos en la asignatura para estudiar diferentes cuestiones sobre este conjunto de datos.

1.1. Problemática planteada

Lo que nos hemos propuesto es llevar a cabo un estudio sobre diferentes preguntas para, a partir de las conclusiones obtenidas, poder ofrecer distintos planes de mejora para el funcionamiente del servicio *BiciMAD*, como por ejemplo en qué momentos se requiere una mayor número de personal, qué estaciones sufren un mayor uso y por tanto un mayor desgaste a tener en cuenta o también a que público es más necesario hacer llegar la disponibilad de este servicio, entre otros.

Para llegar a todas estas medidas hemos deciddo intentar dar respuesta a las siguientes cuestiones:

1. Determinar los trayectos más realizados y los menos.
2. Cuáles son las estaciones más utilizadas y las que menos.
3. Determinar la hora punta de uso.
4. Calcular los porcentajes de uso dependiendo del rango de edad y el tipo de usuario.
5. Calcular la cantidad de bicicletas rotas, es decir, cuyo viaje consta de un tiempo bajo, digamos $\leq$60 segundos.
6. Intentar ver el número de clientes habituales que tiene este servicio.

1.2. Base de datos

Nostros hemos enfocado nuestro estudio sobre los meses de enero a junio de 2021, obteniendo los datos de el Ayuntamiento de Madrid. Los distintos datos de los que disponemos son:

- **Tipo de usuario:**
  - 0: No se ha podido determinar el tipo de usuario.
  - 1: Usuario anual (poseedor de un pase anual).
  - 2: Usuario ocasional.
  - 3: Trabajador de la empresa.
- **Código de usuario.**
- **Número de la estación donde se desengacha la bicicleta.**

- **Número de la estación donde se enchancha la bicicleta.**
- **Número de la base de la que se desengancha la bicicleta.**
- **Número de la base en la que se engancha la bicicleta.**
- **Tiempo transcurrido entre el enganche y el desenganche de la bicicleta.**
- **Hora a la que se realiza el desenganche de la bicicleta:**
  El fórmato es: "2019-06-01T00:00:00Z".
- **Rango de edad del usuario:**
  - Sin datos.
  - <17.
  - 17-18.
  - 19-26.
  - 27-40.
  - 41-65.
  - >65.

1.3. Tratamiento de los datos

Como ya hemos mencionado para el tratamiento de los datos lo realizaremos con la herramienta *Spark*. Para leer los archivos en formato *.json* utilizamos el paquete json de Python. Estos datos se guardan en una estructura RDD que es la que manejaremos a partir de ahora. El primer paso, para que sea más cómodo el manejo de los datos utilizamos la siguiente función, que nos permite manejar por separado cada uno de los distintos datos que hemos mencionado anteriormente.

```
    def mapper(line):
        data = json.loads(line)
        user_type = data['user_type']
        user_age = data['ageRange']
        user_day_code = data['user_day_code']
        start_station = data['idunplug_station']
        end_station = data['idplug_station']
        duration = data['travel_time']
        date = datetime.strptime(data['unplug_hourTime']['\$date'
           ][:-4],)
        return user_type, user_day_code, start_station, end_station,
            duration, date, user age
```

Una vez que hemos aplicado esta función ya podemos enfocarnos a responder cada una de las cuestiones que nos hemos planteado utilizando distintas metodologías.

2. Cuestiones planteadas

A continuación veremos como hemos abordado cada una de las cuestiones y que conclusiones hemos obtenido de cada una de ellas.

2.1. Determinar los trayectos más realizados y los que menos

La función que hemos diseñado para resolver esta cuestión es:

```
    def trayectos_habituales(rdd):
        trayectos_ordenados = rdd.map(lambda x: (x[2],x[3]) ).\
                              countByValue().\
                              sortBy(lamda x: x[1] , ascending =
                                 False)
    return trayectos_ordenados
```

Los resultados que hemos obtenido en cada uno de los ficheros son:

- **Enero 2021**:

```
    Las 5 rutas mas repetidas son:
    [(149, 149), (57, 57), (187, 187), (132, 132), (175, 175)]
    realizadas cada una este numero de veces:
    [154, 144, 111, 108, 95]
```

- **Febrero 2021:**

```
    Las 5 rutas mas repetidas son:
    [(132, 132), (57, 57), (220, 220), (135, 135), (212, 212)]
    realizadas cada una este numero de veces:
    [251, 233, 173, 169, 158]
```

- **Marzo 2021:**

```
    Las 5 rutas mas repetidas son:
    [(132, 132), (57, 57), (90, 90), (135, 135), (56, 56)]
    realizadas cada una este numero de veces:
    [374, 272, 262, 248, 214]
```

- **Abril 2021:**

```
    Las 5 rutas mas repetidas son:
    [(132, 132), (57, 57), (135, 135), (56, 56), (175, 175)]
    realizadas cada una este numero de veces:
    [341, 199, 196, 186, 185]
```

- **Mayo 2021:**

```
    Las 5 rutas mas repetidas son:
    [(132, 132), (220, 220), (135, 135), (175, 175), (57, 57)]
    realizadas cada una este numero de veces:
    [313, 296, 265, 255, 250]
```

- **Junio 2021:**

```
    Las 5 rutas mas repetidas son:
    [(135, 135), (132, 132), (57, 57), (64, 64), (175, 175)]
    realizadas cada una este numero de veces:
    [258, 249, 212, 193, 188]
```

Llama la atención que los trayectos más realizado son aquellos donde la estación de partida y la de llegada son la misma, por lo que podemos intuir que la mayoría de trayectos que se realizan con el servicio BiciMad no son como un servicio de transporte para ir de un lugar a otro, si no que el uso principal es más bien para dar paseos por la ciudad. La principal conclusión que podemos obtener a partir de ello es que lo usuarios no dan a BiciMAD el uso para el que en principio está diseñado, que es un medio de transporte como pudiera ser el autobús o el coche particular. Una medida a considerar a partir de este resultado es intentar potenciar el uso de BiciMAD como medio de transporte, ya sea aumentando el número de estaciones o aumentando la zona de servicio en todo el municipio de Madrid.

2.2. Cuáles son las estaciones más utilizadas y las que menos

La función que hemos diseñado para resolver esta cuestión es:

```
    def rutas_ordenadas(rdd):
        rutas_ordenadas = rdd.map(lambda x: ((x[2], x[3]), 1).\
            reduceByKey(lambda x, y: x+y).\
            sortBy(lambda x: x[1], ascending = False)
        return rutas_ordenadas
```

El resultado obtenido en cada uno de los ficheros es:

- **Enero 2021:**

```
    Estas son las 5 estaciones mas transitadas:
    [57, 43, 175, 208, 149]
    con este numero de usos cada una:
    [3276, 3180, 2352, 2337, 2176]
    Ademas, estas son las 5 estaciones menos transitadas:
    [257, 265, 266, 261, 267]
    con este numero de usos cada una:
    [84, 104, 112, 125, 125]
```

- **Febrero 2021:**

```
    Estas son las 5 estaciones mas transitadas:
    [43, 57, 175, 208, 132]
    con este numero de usos cada una:
    [6419, 6284, 4852, 4838, 4624]
    Ademas, estas son las 5 estaciones menos transitadas:
    [209, 266, 257, 258, 260]
    con este numero de usos cada una:
    [21, 190, 263, 322, 346]
```

- **Marzo 2021:**

```
    Estas son las 5 estaciones mas transitadas:
    [57, 43, 208, 132, 90]
    con este numero de usos cada una:
    [8544, 8239, 6622, 6494, 6449]
    Ademas, estas son las 5  estaciones menos transitadas:
    [2009, 124, 257, 266, 28]
    con este numero de usos cada una:
    [79, 85, 413, 459, 474]
```

- **Abril 2021:**

```
    Estas son las 5 estaciones mas transitadas:
    [57, 43, 132, 90, 175]
    con este numero de usos cada una:
    [7747, 7722, 6179, 6085, 6013]
    Ademas, estas son las 5 estaciones menos transitadas:
    [209, 266, 257, 258, 225]
    con este numero de usos cada una:
    [48, 389, 409, 537, 624]
```

- **Mayo 2021:**

```
    Estas son las 5 estaciones mas transitadas:
    [43, 57, 175, 132, 208]
    con este numero de usos cada una:
    [9676, 9303, 7872, 7504, 7324]
    Ademas, estas son las 5  estaciones menos transitadas:
    [209, 266, 157, 257, 28]
    con este numero de usos cada una:
    [258, 264, 397, 508, 617]
```

- **Junio 2021:**

```
    Estas son las 5 estaciones mas transitadas:
    [43, 57, 175, 208, 132]
    con este numero de usos cada una:
    [9684, 9019, 7477, 7314, 7102]
```

```
    Ademas, estas son las 5  estaciones menos transitadas:
    [209, 266, 257, 28, 225]
    con este numero de usos cada una:
    [40, 447, 599, 612, 767]
```

A partir de estos resultados podemos concluir que la mayoría de estaciones más utilizadas coinciden con las que se realizan más trayectos sobre ellas. Sin embargo, hay algunas que no, por lo que podemos concluir que estas sirven como punto de partida o de llegada para multitud de viajes entre distintos puntos de la ciudad. Conocer cuales son estas estaciones más utilizadas creemos que es relevante pues por ese mismo motivo también serán las que sufran un mayor desgaste y deterioro, por lo que sería necesario aplicar sobre ellas un mayor mantenimiento que sobre las demás estaciones, además de aumentar la capacidad de bicicletas en estas estaciones, si son las que sufren mayor demanda debemos evitar que estas no puedan dar el servicio que los ciudadanos requieren de ellas. En cambio, en cuanta a las estaciones menos utilizadas podemos oberservar que son prácticamente las mismas en ambos caso, por lo que sería recomendable llevar a cabo campañas para incentivar el uso en las zonas donde se encuentran estas estaciones.

2.3. Determinar la hora punta de uso

La función que hemos diseñado para resolver esta cuestión es:

```
    def horas_ordenadas(rdd):
        horas = rdd.map(lambda x: (x[5].hour, 1)).\
            reduceByKey(lambda x, y: x+y).\
            sortBy(lambda x: x[1], ascending = False)
        total = horas.map(lambda x: x[1]).sum()
        horas = horas.map(lambda x: (x[0], x[1]*100/total))
        return horas
```

El resultado obtenido en cada uno de los ficheros es:

- **Enero 2021:**

```
    Las horas ordenadas en cuanto a mayor uso son:
    [13, 17, 18, 16, 14]
    con porcentajes
    [8.1334, 8.0881, 7.5118, 7.2651, 6.8202]
```

- **Febrero 2021:**

```
    Las horas ordenadas en cuanto a mayor uso son:
    [17, 13, 18, 16, 14]
    con porcentajes
    [8.8801, 8.3952, 7.9301, 7.4238, 7.3002]
```

- **Marzo 2021:**

```
Las horas ordenadas en cuanto a mayor uso son:
[17, 18, 13, 16, 14]
con porcentajes:
[8.5659, 8.0358, 7.6347, 7.3155, 6.8783]
```

- **Abril 2021:**

```
Las horas ordenadas en cuanto a mayor uso son:
[17, 16, 12, 18, 15]
con porcentajes
[8.9743, 8.8333, 7.652, 7.3359, 7.1615]
```

- **Mayo 2021:**

```
Las horas ordenadas en cuanto a mayor uso son:
[17, 16, 12, 18, 15]
con porcentajes:
[8.3155, 8.161, 7.2876, 7.0969, 6.646]
```

- **Junio 2021:**

```
Las horas ordenadas en cuanto a mayor uso son:
[17, 16, 18, 12, 15]
con porcentajes:
[8.055, 7.7642, 7.2579, 6.91, 6.2039]
```

Se puede observar claramente que las horas del día en las que se concentra el mayor uso del servicio son entre las 12:00h y las 18:00h. Esto nos indica que a estas horas también debe ser cuando se concentre la mayor actividad de los servicios de mantenimiento, reparación, atención al cliente y demás, pues será cuando más averías, dudas e incidencias haya. Luego a estas horas se debe garantizar el funcionamiento al cien por ciento del sistema, mientras que quizás a otras horas del día puede ser mucho menor ya que el uso es mínimo y por tanto sería innecesario malgastar recursos en esos momentos.

2.4. Calcular los porcentajes de uso dependiendo del rango de edad y tipo de usuario

La función que hemos diseñado para resolver esta cuestión es:

```
def edades_ordenadas(rdd):
    edades = rdd.map(lambda x: (user_ages[x[0]], 1)).\
        reduceByKey(lambda x, y: x+y).\
        sortBy(lambda x: x[1], ascending = False)
    total = edades.map(lambda x: x[1]).sum()
    edades = edades.map(lambda x: (x[0], x[1]*100/total))
    return edades
```

El resultado obtenido en cada uno de los ficheros es:

- **Enero 2021:**

```
    Las edades de los usuarios ordenadas en cuanto a mas uso son:
    [NaN, 27-40, 41-65, 19-26, <17, >65, 17-18]
    con porcentajes:
    [53.9963, 21.0145, 19.0422, 3.9079, 1.0928, 0.4974, 0.4489]
    Los tipos de usuario ordenadas en cuanto a mas uso son:
    [Usuario anual, Trabajador de empresa, NaN, Usuario ocasional]
    con porcentajes:
    [95.4394, 3.7901, 0.5444, 0.226]
```

- **Febrero 2021:**

```
    Las edades de los usuarios ordenadas en cuanto a mas uso son:
    [NaN, 27-40, 41-65, 19-26, <17, >65, 17-18]
    con porcentajes:
    [54.4523, 21.1341, 18.3466, 3.8328, 1.412, 0.5231, 0.2991]
    Los tipos de usuario ordenadas en cuanto a mas uso son:
    [Usuario anual, Trabajador de empresa, NaN, Usuario ocasional]
    con porcentajes:
    [94.7822, 4.2335, 0.7035, 0.2808]
```

- **Marzo 2021:**

```
    Las edades de los usuarios ordenadas en cuanto a mas uso son:
    [NaN, 27-40, 41-65, 19-26, <17, >65, 17-18]
    con porcentajes:
    [54.6143, 20.9724, 18.7327, 3.4235, 1.5218, 0.4808, 0.2545]
    Los tipos de usuario ordenadas en cuanto a mas uso son:
    [Usuario anual, Trabajador de empresa, NaN, Usuario ocasional]
    con porcentajes:
    [93.26, 5.4785, 0.7658, 0.4957]
```

- **Abril 2021:**

```
    Las edades de los usuarios ordenadas en cuanto a mas uso son:
    [NaN, 27-40, 41-65, 19-26, <17, >65, 17-18]
    con porcentajes:
    [55.0183, 20.9787, 18.8116, 3.3707, 1.0083, 0.5152, 0.2973]
    Los tipos de usuario ordenadas en cuanto a mas uso son:
    [Usuario anual, Trabajador de empresa, NaN, Usuario ocasional]
    con porcentajes:
    [94.1547, 4.6018, 0.7447, 0.4987]
```

- **Mayo 2021:**

```
Las edades de los usuarios ordenadas en cuanto a mas uso son:
[NaN, 27-40, 41-65, 19-26, <17, >65, 17-18]
con porcentajes:
[55.8058, 20.6822, 18.2624, 3.3106, 1.0991, 0.5115, 0.3283]
Los tipos de usuario ordenadas en cuanto a mas uso son:
[Usuario anual, Trabajador de empresa, NaN, Usuario ocasional]
con porcentajes:
[93.818, 4.6305, 1.0148, 0.5366]
```

- **Junio 2021:**

```
Las edades de los usuarios ordenadas en cuanto a mas uso son:
[NaN, 27-40, 41-65, 19-26, <17, >65, 17-18]
con porcentajes:
[55.5935, 20.6716, 18.6095, 3.1952, 1.1585, 0.4964, 0.2753]
Los tipos de usuario ordenadas en cuanto a mas uso son:
[Usuario anual, Trabajador de empresa, NaN, Usuario ocasional]
con porcentajes:
[94.6854, 3.6694, 1.1733, 0.472]
```

El porcentaje de uso según edades es casi idéntico en los tres casos. Por un lado tenemos que la inmensa mayoría de usuarios son de una edad menor de 26 años, es decir, jóvenes. Esto es algo muy positivo pues significa que el uso de la bici está muy extendido entre los más jóvenes y optan por ello como medio de transporte. Sin embargo, en cuanto a la población adulta y más mayor pues prácticamente no usan este servicio. Creemos que es importante intentar fomentar en esta parte de la población el uso de este servicio.

2.5. Calcular la cantidad de bicicletas rotas

En esta cuestión buscamos calcular el número de veces que un usuario ha dispuesto a utilizar el servicio pero se ha encontrado con una bicicleta rota. Este tipo de incidencia no está especificada en la base de datos así que para considerarla lo que hemos hecho es considerar que la bicicleta estaba rato cuando el trayecto ha tenido una duración de menos de 1 minuto. La función que hemos diseñado para resolver esta cuestión es:

```
def bicis_rotas(rdd):
    total = rdd.count()
    rotas = rdd.filter(lambda x: x[4]<60).count()
    return rotas, round(rotas/total, 4))
```

Los resultados en cada uno de los ficheros ha sido:

- **Enero 2021:**

```
El numero de veces que alguien ha cogido una bici rota es:
2996
```

```
    de un total de viajes de:
    125642 (0.0238%)
```

- **Febrero 2021:**

```
    El numero de veces que alguien ha cogido una bici rota es:
    8634
    de un total de viajes de:
    262103 (0.0329%)
```

- **Marzo 2021:**

```
    El numero de veces que alguien ha cogido una bici rota es:
    14732
    de un total de viajes de:
    360684 (0.0408 %)
```

- **Abril 2021:**

```
    El numero de veces que alguien ha cogido una bici rota es:
    11195
    de un total de viajes de:
    341061 (0.0328 %)
```

- **Mayo 2021:**

```
    El numero de veces que alguien ha cogido una bici rota es:
    12688
    de un total de viajes de:
    414249 (0.0306 %)
```

- **Junio 2021:**

```
    El numero de veces que alguien ha cogido una bici rota es:
    11098
    de un total de viajes de:
    413370 (0.0268 %)
```

3. Conclusión del plan de mejora y eficiencia para el servicio BiciMAD

Una vez que hemos estudiado los distintos aspectos a tener en cuenta del servicio de transporte BiciMAD vamos a proponer un plan con medidas a tener en cuenta para mejorar y aumentar la eficiencia de dicho servicio. Nuestro plan consta esencialmente de los siguientes puntos:

1. Potenciar el uso de BiciMAD como medio de transporte para desplazarse en la ciudad, es decir, que su uso principal no sea únicamente el de pasear en bicicleta por la ciudad, si no que sirva para sustituir a autobuses, metro, vehículo particular, etc. Este factor es importante para reducir el uso de medios de transporte más contaminantes y apostar por medidas más ecológicas y saludables, donde el servicio BiciMAD debe jugar un importante papel.

2. Las estaciones más utilizadas y que por tanto sufrirán un mayor desgaste como las estaciones 57, 43 o 175 deben contar con un mantenimiento más intenso y constante que el resto de estaciones. Además, se debe intentar, en la medida de lo posible, aumentar el número de bicicletas disponibles en estas estaciones ya que no debería permitirse que se queden sin servicio en ningún momento.

3. A las horas puntas de uso, es decir, entre las 12:00h y las 18:00h apróximadamente se debe contar con el cien por cien de los servicios de BiciMAD (reparaciones, atención al ciente, etc.), ya que en estos momentos es cuando se darán un mayor número de incidencias a resolver de forma simultánea. En caso de que aún así fuese insuficiente para mantener el funcionamiento del servicio, lo que se debería hacer es aumentar la capacidad de BiciMAD en todos los aspectos.

4. En cuanto a las horas puntas de uso, también se debe intentar garantizar, en la medida de lo posible, que sean accesibles en los meses de más calor, es decir, en verano. Ya que mucha gente que requiera de este servicio en los meses que nosotros hemos estudiado lo seguirá necesitando en los meses más calurosos. Como medidas para ayudar en estos casos proponemos cubrir las estaciones para que estas se encuentren a la sombra y que no sea un problema la alta temperatura de las bicicletas para su uso. Además, esta medida también sería útil en verano pues permitiría que no se mojasen las bicicletas en caso de lluvias. Otra medida sería proporcionar fuentes de agua potable en las estaciones para que los usuarios puedan evitar sufrir por la deshidratación en caso de altas temperaturas.

5. Por último, creemos importante llevar a cabo planes de atracción de usuarios para fomentar el uso de BiciMAD. Sería importante enfocar estos planes tanto en las zonas de las estaciones que sufren un menor, como en el sector de la población adulta que hemos visto que apenas dan uso al servicio BiciMAD.