In [1]:

```
#student_name:YonghengHou
#student_number:5556661
#login:yh790
from pyspark.sql import SparkSession
import numpy as np
```

In [2]:

```
spark = SparkSession.builder.appName("CSCI316-project") \
.config("spark-master", "local") \
.getOrCreate()
spark
```

Out[2]:

**SparkSession - in-memory**

**SparkContext**

Spark UI (http://windows10.microdone.cn:4041)

**Version**

`v2.4.4`

**Master**

`local[*]`

**AppName**

`CSCI316-project`

In [3]:

```
from pyspark.sql.types import *
myManualSchema = StructType([
StructField("Session_ID",IntegerType(), True),
StructField("Timestamp", TimestampType(), True),
StructField("Item_ID", IntegerType(), True),
StructField("Category", StringType(), True)])

df_fd = spark \
.read \
.format("csv") \
.schema(myManualSchema).load("yoochoose-clicks.dat")
```

In [4]:

```
df_fd.printSchema()
```

```
root
 |-- Session_ID: integer (nullable = true)
 |-- Timestamp: timestamp (nullable = true)
 |-- Item_ID: integer (nullable = true)
 |-- Category: string (nullable = true)
```

In [5]:

```
#transfer the 8-10 digits number to "B"
from pyspark.sql.functions import regexp_replace,col
cateFilter="[0-9]{7,10}"
df_FD=df_fd.withColumn("Category",regexp_replace(col("Category"),cateFilter,"B"))
df_FD.show()
```

```
+----------+--------------------+---------+--------+
|Session_ID|           Timestamp|  Item_ID|Category|
+----------+--------------------+---------+--------+
|         1|2014-04-07 20:51:...|214536502|       0|
|         1|2014-04-07 20:54:...|214536500|       0|
|         1|2014-04-07 20:54:...|214536506|       0|
|         1|2014-04-07 20:57:...|214577561|       0|
|         2|2014-04-07 23:56:...|214662742|       0|
|         2|2014-04-07 23:57:...|214662742|       0|
|         2|2014-04-07 23:58:...|214825110|       0|
|         2|2014-04-07 23:59:...|214757390|       0|
|         2|2014-04-08 00:00:...|214757407|       0|
|         2|2014-04-08 00:02:...|214551617|       0|
|         3|2014-04-03 00:17:...|214716935|       0|
|         3|2014-04-03 00:26:...|214774687|       0|
|         3|2014-04-03 00:30:...|214832672|       0|
|         4|2014-04-07 22:09:...|214836765|       0|
|         4|2014-04-07 22:26:...|214706482|       0|
|         6|2014-04-07 02:58:...|214701242|       0|
|         6|2014-04-07 03:02:...|214826623|       0|
|         7|2014-04-02 17:38:...|214826835|       0|
|         7|2014-04-02 17:39:...|214826715|       0|
|         8|2014-04-06 18:49:...|214838855|       0|
+----------+--------------------+---------+--------+
only showing top 20 rows
```

In [6]:

```
#create second same dataframe and  change column name in order to join this two tables in the ne
xt steps
second_df=df_FD.selectExpr("Session_ID as newSession_ID","Timestamp as newTimestamp","Item_ID as
newItem_ID","Category as newCategory")

#for join,I index two dataframes,and one dataframe index from 0 ,anther one is index from 1,
#it can has dislocation by subtracting to calcaulate interval time
from pyspark.sql.functions import monotonically_increasing_id
df1 = df_FD.withColumn("index", monotonically_increasing_id()+1)
df2 =second_df.withColumn("index", (monotonically_increasing_id()))
df1.show(10)
df2.show(10)
```

```
+----------+--------------------+---------+--------+-----+
|Session_ID|           Timestamp|  Item_ID|Category|index|
+----------+--------------------+---------+--------+-----+
|         1|2014-04-07 20:51:...|214536502|       0|    1|
|         1|2014-04-07 20:54:...|214536500|       0|    2|
|         1|2014-04-07 20:54:...|214536506|       0|    3|
|         1|2014-04-07 20:57:...|214577561|       0|    4|
|         2|2014-04-07 23:56:...|214662742|       0|    5|
|         2|2014-04-07 23:57:...|214662742|       0|    6|
|         2|2014-04-07 23:58:...|214825110|       0|    7|
|         2|2014-04-07 23:59:...|214757390|       0|    8|
|         2|2014-04-08 00:00:...|214757407|       0|    9|
|         2|2014-04-08 00:02:...|214551617|       0|   10|
+----------+--------------------+---------+--------+-----+
only showing top 10 rows

+-------------+--------------------+----------+-----------+-----+
|newSession_ID|        newTimestamp|newItem_ID|newCategory|index|
+-------------+--------------------+----------+-----------+-----+
|            1|2014-04-07 20:51:...| 214536502|          0|    0|
|            1|2014-04-07 20:54:...| 214536500|          0|    1|
|            1|2014-04-07 20:54:...| 214536506|          0|    2|
|            1|2014-04-07 20:57:...| 214577561|          0|    3|
|            2|2014-04-07 23:56:...| 214662742|          0|    4|
|            2|2014-04-07 23:57:...| 214662742|          0|    5|
|            2|2014-04-07 23:58:...| 214825110|          0|    6|
|            2|2014-04-07 23:59:...| 214757390|          0|    7|
|            2|2014-04-08 00:00:...| 214757407|          0|    8|
|            2|2014-04-08 00:02:...| 214551617|          0|    9|
+-------------+--------------------+----------+-----------+-----+
only showing top 10 rows
```

In [7]:

```
#lefter join two tables,df1 and df2
df3 = df1.join(df2, "index", "left_outer")
df3.show(10)
```

```
+-----+----------+--------------------+---------+--------+-------------+----------
----------+---------+-----------+
|index|Session_ID|           Timestamp|  Item_ID|Category|newSession_ID|        ne
wTimestamp|newItem_ID|newCategory|
+-----+----------+--------------------+---------+--------+-------------+----------
----------+---------+-----------+
|   26|        11|2014-04-03 21:45:...|214821275|       0|           11|2014-04-03
21:45:...| 214821371|          0|
|   29|        11|2014-04-03 21:46:...|214821371|       0|           11|2014-04-03
21:53:...| 214717089|          0|
|  474|       154|2014-04-03 20:04:...|214560187|       0|          154|2014-04-03
20:05:...| 214716984|          0|
|  964|       337|2014-04-04 05:52:...|214820842|       0|          337|2014-04-04
05:56:...| 214826897|          0|
| 1677|       564|2014-04-02 21:49:...|214629060|       0|          564|2014-04-02
22:09:...| 214840899|          0|
| 1697|       564|2014-04-02 23:26:...|214596647|       0|          564|2014-04-02
23:27:...| 214837558|          0|
| 1806|       531|2014-04-02 01:58:...|214748336|       0|          531|2014-04-02
01:59:...| 214717247|          0|
| 1950|       638|2014-04-02 23:37:...|214579730|       0|          637|2014-04-02
05:54:...| 214537867|          0|
| 2040|       603|2014-04-07 18:20:...|214684513|       0|          602|2014-04-02
23:21:...| 214819562|          0|
| 2214|       661|2014-04-02 04:40:...|214832559|       0|          661|2014-04-02
04:41:...| 214819550|          0|
+-----+----------+--------------------+---------+--------+-------------+----------
----------+---------+-----------+
only showing top 10 rows
```

In [8]:

```
#this function is to calculate interval time and put the result into new column
import pyspark.sql.functions as F
df4=df3.withColumn(
    "interval_time",
    (F.col("newTimestamp").cast("long") - F.col("Timestamp").cast("long")))
df4.show(10)
```

```
+-----+----------+--------------------+---------+--------+-------------+----------
----------+----------+-----------+-------------+
|index|Session_ID|           Timestamp|  Item_ID|Category|newSession_ID|        ne
wTimestamp|newItem_ID|newCategory|interval_time|
+-----+----------+--------------------+---------+--------+-------------+----------
----------+----------+-----------+-------------+
|   26|        11|2014-04-03 21:45:...|214821275|       0|           11|2014-04-03
21:45:...| 214821371|          0|           28|
|   29|        11|2014-04-03 21:46:...|214821371|       0|           11|2014-04-03
21:53:...| 214717089|          0|          385|
|  474|       154|2014-04-03 20:04:...|214560187|       0|          154|2014-04-03
20:05:...| 214716984|          0|           42|
|  964|       337|2014-04-04 05:52:...|214820842|       0|          337|2014-04-04
05:56:...| 214826897|          0|          222|
| 1677|       564|2014-04-02 21:49:...|214629060|       0|          564|2014-04-02
22:09:...| 214840899|          0|         1226|
| 1697|       564|2014-04-02 23:26:...|214596647|       0|          564|2014-04-02
23:27:...| 214837558|          0|           49|
| 1806|       531|2014-04-02 01:58:...|214748336|       0|          531|2014-04-02
01:59:...| 214717247|          0|           23|
| 1950|       638|2014-04-02 23:37:...|214579730|       0|          637|2014-04-02
05:54:...| 214537867|          0|       -63803|
| 2040|       603|2014-04-07 18:20:...|214684513|       0|          602|2014-04-02
23:21:...| 214819562|          0|      -417493|
| 2214|       661|2014-04-02 04:40:...|214832559|       0|          661|2014-04-02
04:41:...| 214819550|          0|           11|
+-----+----------+--------------------+---------+--------+-------------+----------
----------+----------+-----------+-------------+
only showing top 10 rows
```

In [9]:

```
#because all seesion_id is continous,so it will come out the condition that last record of one s
eession_id
# subtract the first record of another one session_id,so it is wrong condition,so following step
is to filter this
"""
example:
last column negative number

| 1950|       638|2014-04-02 23:37:...|214579730|       0|          637|2014-04-02 05:54:...| 21
4537867|          0|        -63803|
| 2040|       603|2014-04-07 18:20:...|214684513|       0|          602|2014-04-02 23:21:...| 21
4819562|          0|       -417493|
| 2214|       661|2014-04-02 04:40:...|214832559|       0|          661|2014-04-02 04:41:...| 21
4819550|          0|            11|

"""
df5=df4.filter((col("Session_ID") == col("newSession_ID")))
df5.show(10)
```

```
+-----+----------+--------------------+---------+--------+-------------+----------
----------+----------+-----------+-------------+
|index|Session_ID|           Timestamp|  Item_ID|Category|newSession_ID|        ne
wTimestamp|newItem_ID|newCategory|interval_time|
+-----+----------+--------------------+---------+--------+-------------+----------
----------+----------+-----------+-------------+
|   35|        11|2014-04-03 21:57:...|214826837|       0|           11|2014-04-03
21:57:...| 214819762|          0|           20|
|  201|        62|2014-04-07 01:44:...|214826619|       0|           62|2014-04-07
01:45:...| 214746427|          0|           53|
|  288|        86|2014-04-02 05:21:...|214648340|       0|           86|2014-04-02
05:21:...| 214648438|          0|           31|
|  643|       219|2014-04-01 18:07:...|214725500|       0|          219|2014-04-01
18:10:...| 214839660|          0|          161|
| 1219|       389|2014-04-07 03:20:...|214691396|       0|          389|2014-04-07
03:21:...| 214691321|          0|           42|
| 1184|       397|2014-04-05 18:19:...|214553540|       0|          397|2014-04-05
18:19:...| 214572538|          0|           26|
| 1310|       479|2014-04-03 20:09:...|214820814|       0|          479|2014-04-03
20:11:...| 214746339|          0|          144|
| 1521|       496|2014-04-08 03:39:...|214638977|       0|          496|2014-04-08
03:39:...| 214638977|          0|            9|
| 1647|       554|2014-04-08 11:36:...|214829312|       0|          554|2014-04-08
11:36:...| 214774685|          0|           33|
| 2154|       651|2014-04-02 04:17:...|214718117|       0|          651|2014-04-02
04:17:...| 214690845|          0|           37|
+-----+----------+--------------------+---------+--------+-------------+----------
----------+----------+-----------+-------------+
only showing top 10 rows
```

In [10]:

```
#1.this step is for the condition that in one seesion_id,it has more than or equal two same item
_id operations,
#because this operations belong to one item,so group it,and sum the interval_time of same item_i
d records.
#2.why I max("Category") here,because I just want to keep Category column after groupby,But I di
dnt find good
#solution to generate it,so I have to do it like this.
"""
example:interval time of index 4 and index 5 sum together
                                     item_id           cate  index
             2|2014-04-07 23:56:...| 214662742|          0|    4|
|            2|2014-04-07 23:57:...| 214662742|          0|    5|
|            2|2014-04-07 23:58:...| 214825110|          0|    6|
|            2|2014-04-07 23:59:...| 214757390|          0|    7|
|            2|2014-04-08 00:00:...| 214757407|          0|    8|
|            2|2014-04-08 00:02:...| 214551617|          0|    9|
"""
df6=df5.groupBy([col("Session_ID"),col("Item_ID")])\
.agg(F.sum(col("interval_time")).alias("complete_interval_time"),F.max(col("Category")).alias("C
ategory"))
df6.show(10)
```

```
+----------+---------+----------------------+--------+
|Session_ID|  Item_ID|complete_interval_time|Category|
+----------+---------+----------------------+--------+
|        87|214554637|                    13|       0|
|       119|214716954|                   794|       0|
|       397|214843492|                   127|       0|
|       491|214832559|                    42|       0|
|       492|214718220|                    93|       0|
|       516|214676364|                    59|       0|
|       577|214708372|                    48|       0|
|       626|214827005|                    54|       0|
|       651|214838833|                   252|       0|
|       726|214589632|                    23|       0|
+----------+---------+----------------------+--------+
only showing top 10 rows
```

In [11]:

```
#1. so far, we have the complete_interval_time for each user on each same item
# 2, group by category, then sum the complete_interval_time, then divide by total number of each us
er each item
df7=df6.groupBy([col("Category")]).agg(F.count(col('Item_ID')),F.sum(col('complete_interval_tim
e')))
df7.show(10)
```

```
+--------+--------------+---------------------------+
|Category|count(Item_ID)|sum(complete_interval_time)|
+--------+--------------+---------------------------+
|       7|        220258|                   47393374|
|      11|         44447|                    6591571|
|       3|        534948|                   70182484|
|       8|         27314|                    5099778|
|       0|       9850312|                 1725863516|
|       5|        257109|                   62751734|
|       B|         46800|                    9880439|
|       6|        223263|                   52894611|
|       S|       6534387|                 1111613782|
|       9|         61836|                   12428756|
+--------+--------------+---------------------------+
only showing top 10 rows
```

In [12]:

```
#lastly, calculate the average time
final_result=df7.withColumn("Average_time", (col("sum(complete_interval_time)")/col("count(Item_I
D)")))
final_result.show(15)
```

```
+--------+--------------+---------------------------+------------------+
|Category|count(Item_ID)|sum(complete_interval_time)|      Average_time|
+--------+--------------+---------------------------+------------------+
|       7|        220258|                   47393374|215.17208909551525|
|      11|         44447|                    6591571|148.30182014534165|
|       3|        534948|                   70182484| 131.1949647442368|
|       8|         27314|                    5099778|186.70930658270484|
|       0|       9850312|                 1725863516|175.20902038432894|
|       5|        257109|                   62751734| 244.0666565542241|
|       B|         46800|                    9880439|211.12049145299144|
|       6|        223263|                   52894611|236.91615269883502|
|       S|       6534387|                 1111613782|   170.11753084107|
|       9|         61836|                   12428756|200.99547189339543|
|       1|       1013746|                  201798049| 199.0617462362367|
|      10|         41407|                    8073943| 194.9898084864878|
|       4|        281139|                   55556414|197.61190727718318|
|      12|          9672|                    2787961|288.25072373862696|
|       2|        742758|                  157088599|211.49364799840595|
+--------+--------------+---------------------------+------------------+
```

In [ ]: