Advertisement_Analysis

November 8, 2023

[10]:

```
from pyspark.sql import SparkSession
from pyspark.sql.types import *
from datetime import datetime

# Create Spark session
spark = SparkSession.builder \
    .appName("Spark with Hive") \
    .enableHiveSupport() \
    .getOrCreate()

hdfs_path = '/tmp/input_data/'

## Loading of JSON Ad_campaigns.json data in dataframe

ad_campaigns_df=spark.read.format("json")\
                .option("multiline", "true")\
                .load(hdfs_path+"ad_campaigns_data.json")
```

[11]:

```
## Loading user profile data
user_profile_df=spark.read.format("json")\
                .option("multiline", "true")\
                .load(hdfs_path+"user_profile_data.json")
user_profile_df.show()
```

```
+---------+------------------+-------+------+-------------------+
|age_group|          category|country|gender|            user_id|
+---------+------------------+-------+------+-------------------+
|    18-25|[shopper, student]|    USA|  male|1264374214654454321|
+---------+------------------+-------+------+-------------------+
```

[12]:

```
##loading profile_data
store_df=spark.read.format("json")\
                .option("multiline", "true")\
                .load(hdfs_path+"store_data.json")
```

[13]:

```
from pyspark.sql.functions import *
```

```
[ ]: #Analyse data for each campaign_id, date, hour, os_type & value to get all the␣
       ↪events with counts
```

```
[14]: ad_campaigns_df.show()
```

```
+----------------+-----------+--------------------+-----------+-----------------
---+----------+-------+---------+-------------------+
|campaign_country|campaign_id|       campaign_name|device_type|
event_time|event_type|os_type| place_id|            user_id|
+----------------+-----------+--------------------+-----------+-----------------
---+----------+-------+---------+-------------------+
|             USA|    ABCDFAE|Food category tar…|
apple|2018-10-12T13:10:…|impression|    ios|CASSBB-11|1264374214654454321|
+----------------+-----------+--------------------+-----------+-----------------
---+----------+-------+---------+-------------------+
```

```
[15]: ad_campaigns=ad_campaigns_df.groupBy("campaign_id",
                              substring(col("event_time"), 0, 10).alias("date"),
                              substring(col("event_time"),12, 2).alias("hour"),
                              col("os_type"),
                              col("event_type")
                             ).agg(count("event_type").alias("events"))\
                              .selectExpr(
                                "campaign_id",
                                "date",
                                "hour",
                                "'os_type' as type",
                                "os_type as value",
                                "struct(event_type, events) as event"
                                ) \
                                .groupBy("campaign_id", "date", "hour", "type",␣
        ↪"value") \
                                .agg(collect_list("event").alias("events")) \
                                .selectExpr(
                                    "campaign_id",
                                    "date",
                                    "hour",
                                    "type",
                                    "value",
                                    "map_from_entries(events) as event"
                                )

      ad_campaigns.show()

      ad_campaigns.coalesce(1).write.format('json').save('/tmp/output_data/
        ↪ad_campaigns/')
```

```
print("Write Successfull")
```

```
+-----------+----------+----+-------+-----+-----------------+
|campaign_id|      date|hour|   type|value|            event|
+-----------+----------+----+-------+-----+-----------------+
|    ABCDFAE|2018-10-12|  13|os_type|  ios|{impression -> 1}|
+-----------+----------+----+-------+-----+-----------------+
```

```
Write Successfull
```

[17]:

```
ad_campaigns.show(6)
```

```
+-----------+----------+----+-------+-----+-----------------+
|campaign_id|      date|hour|   type|value|            event|
+-----------+----------+----+-------+-----+-----------------+
|    ABCDFAE|2018-10-12|  13|os_type|  ios|{impression -> 1}|
+-----------+----------+----+-------+-----+-----------------+
```

[18]:

```
store_df.show(6)
```

```
+--------------------+----------+
|           place_ids|store_name|
+--------------------+----------+
|[CASSBB-11, CADGB...|  McDonald|
+--------------------+----------+
```

[19]:

```
#Analyse data for each campaign_id, date, hour, store_name & value to get all␣
 ↪the events with counts

stores=ad_campaigns_df.join(store_df, array_contains(store_df.place_ids,␣
 ↪ad_campaigns_df.place_id),"left")\
                    .groupBy("campaign_id",
                             substring("event_time", 0, 10).alias('date'),
                             substring("event_time", 12, 2).alias('hour'),
                             "store_name",
                             "event_type"
                             ).agg(count("event_type").alias('events'))\
                    .selectExpr("campaign_id",
                            "date",
                            "hour",
                            "'store_name' as type",
                            "store_name as value",
                            "struct(event_type, events) as event_dict")\
                    .groupBy("campaign_id",
                            "date",
                            "hour",
                            "type",
                            "value"
                            ).agg(collect_list("event_dict").alias('event'))\
                    .select("campaign_id",
                            "date",
                            "hour",
                            "type",
                            "value",
                            map_from_entries("event").alias('event'))
stores.show()
```

```
+-----------+----------+----+----------+--------+-----------------+
|campaign_id|      date|hour|      type|   value|            event|
+-----------+----------+----+----------+--------+-----------------+
|    ABCDFAE|2018-10-12|  13|store_name|McDonald|{impression -> 1}|
+-----------+----------+----+----------+--------+-----------------+
```

[20]:

```
## write data
stores.coalesce(1).write.format('json').save('/tmp/output_data/stores/')
print("Write successful")
```

```
Write successful
```

[21]:

```
user_profile_df.show()
```

```
+---------+------------------+-------+------+-------------------+
|age_group|          category|country|gender|            user_id|
+---------+------------------+-------+------+-------------------+
|    18-25|[shopper, student]|    USA|  male|1264374214654454321|
+---------+------------------+-------+------+-------------------+
```

[22]:

```
#Analyse data for each campaign_id, date, hour, gender_type & value to get all␣
 ↪the events with counts
user_profile=ad_campaigns_df.join(user_profile_df, ad_campaigns_df.user_id ==␣
 ↪user_profile_df.user_id, "left")\
                            .select("campaign_id",
                                    substring("event_time", 0, 10).
 ↪alias("date"),
                                    substring("event_time", 12, 2).
 ↪alias("hour"),
                                    lit('gender').alias("type"),
                                    col("gender").alias("value"),
                                    "event_type")\
                            .groupBy("campaign_id", "date", "hour", "type",␣
 ↪"value", "event_type")\
                            .agg(count("event_type").alias("event_count"))\
                            .select("campaign_id", "date", "hour", "type",␣
 ↪"value", struct("event_type", "event_count").alias("events_map"))\
                            .groupBy("campaign_id", "date", "hour", "type",␣
 ↪"value")\
                            .agg(collect_list("events_map").alias("map_list"))\
                            .select("campaign_id", "date", "hour", "type",␣
 ↪"value", map_from_entries("map_list").alias("event"))

user_profile.show()
```

```
+-----------+----------+----+------+-----+-----------------+
|campaign_id|      date|hour|  type|value|            event|
+-----------+----------+----+------+-----+-----------------+
|    ABCDFAE|2018-10-12|  13|gender| male|{impression -> 1}|
+-----------+----------+----+------+-----+-----------------+
```

[23]:

```
user_profile.coalesce(1).write.format('json').save('/tmp/output_data/
 ↪user_profile')
print("Write successfull")
```

```
Write successfull
```

[ ]:

```
```