musixmatch

via S. Vitale, 5 - 40125 Bologna, Italy
+39 051.18899579
+39 051.18899165
musixmatch.com
@musixmatch

Data Engineering Exercise

Introduction

The goal of the exercise is to test the following skills of the candidate:

- optimizing the underlying data
- creating one or more datasets to support visualizations tools.

Even if the data is anonymized, we nonetheless require that the candidate does not show any “Musixmatch” name / brand / reference.

E.g. Just call the dataset “Test”.

Database connection

The sample data is stored on a publicly exposed MySQL database. The database machine is intentionally underpowered, in order to stimulate data optimization and preparation.

The connection parameters are in the accompanying email.

Database Tables:

The sample data consists of two tables: *users* and *views*.
Each row on the table *users* represents a user profile; while each row on the table *views* represents a single lyrics visualization.

The *users* table consists of the following fields:

- **active**: Indicates if the user is still active
- **age_range**: indicates the approximate age of the user
- **application_id**: indicates the id of the application used to create the user profile the first time
- **country**: country where the user was located during creation
- **gender**: gender of the user, if available
- **language** : language of the user, if available
- **user_email**: email of the user if available
- **user_id**: unique user identifier

The approximate number of rows is 41 million.

The *views* table represents lyrics visualizations during a 1 year period. Consists of the following fields:

- **abstract_id**: Is the ID of the lyrics. Abstrack stands for “abstract track” and indicates all the songs which share the same lyrics.

- **api_application_id**: indicates the id of the application used to create the user profile the first time
- **artist**: name of the artist
- **artist_id**: artist identifier
- **country_code**: country where the user was located to see the lyrics
- **timestamp**: unix timestamp of the lyrics visualization in milliseconds
- **geoip_city**: city where the user was located to see the lyrics
- **guid**: if present, it identifies the device identifier of the user seeing the lyrics
- **latitude**: gps position of the user
- **longitude**: gps position of the user
- **product_type**: type of product
- **track**: title of the track which has been seen
- **transaction_id**: transaction identifier
- **user_id**: unique user identifier

The approximate number of rows is 31 million, across a 1 year timespan.

Exercise description

Step 1:
Prepare the data in order to lower the cardinality of the tables, if needed for speeding up the next step.
You can either create new tables with just the field needed, correctly indexed; or modify the existing ones.

Step 2:
Using PySpark create one or more datasets to support a visualization tool that have to answer to the following questions:

- How many devices are active daily? What product? In which countries? With what applications?
-
- How many users are active daily? What product? In which countries? With what applications?

- What are the most viewed daily lyrics by country and / or application

- How many distinct lyrics account daily, weekly and monthly for 50% of the views. Drill down by country and application.
  Is this constant over the full period of time

Do not invest more than 8h for the task.

To create the spark code the candidate can use any tool (jupiter notebooks or other editor) that wants. Otherwise, a simple solution could be to use “Databricks community

via S. Vitale, 5 - 40125 Bologna, Italy
+39 051.18899579
+39 051.18899165
musixmatch.com
@musixmatch

Edition” (Login - Databricks Community Edition) that is a free version of a cloud-based big data platform; a free account permits the use of a small Spark cluster.