==========
Advanced Certificate Program in Machine Learning and Cloud - upGrad Capstone Project
User Demographics Prediction using Telecom dataset
Data Ingestion Commands

Authors :
Sanjana Mantri
Meghna Shekhar
==========

Getting the source files from S3
==========
wget https://capstone-project-mlc-metadata.s3.amazonaws.com/app_labels_new.txt
wget https://capstone-project-mlc-metadata.s3.amazonaws.com/label_categories.csv

SQL Analysis
==========

Connecting to RDS instance
==========
mysql -h mlc-testcapstone.cyaielc9bmnf.us-east-1.rds.amazonaws.com -u student -p

password to be used
==========
STUDENT123

Showing the available databases
==========
show databases;

Connecting to mlctest database
==========
use mlctest;

Sanity Analysis
==========
desc app_events;
desc train;
desc brand_device;
desc events;

Count Analysis
==========
select count(*) from app_events;
select count(*) from brand_device;
select count(*) from events;
select count(*) from train;

SQL Tasks
==========
1. select count(distinct(device_id)) from train;
2. select device_id, count(device_id) as number_of_duplicate_devices from brand_device group by device_id having count(device_id) > 1);
3. select count(distinct(phone_brand)) from brand_device;
4. select count(device_id) from events where longitude = 0 and latitude = 0;

List the scoop command to load tables from mysql to HDFS
==========
sqoop import --connect jdbc:mysql://mlc-testcapstone.cyaielc9bmnf.us-east-1.rds.amazonaws.com:3306/mlctest --table app_events --target-dir /user/hadoop/mlctest/app_events --username student -P -m 1
sqoop import --connect jdbc:mysql://mlc-testcapstone.cyaielc9bmnf.us-east-1.rds.amazonaws.com:3306/mlctest --table brand_device --target-dir /user/hadoop/mlctest/brand_device --username student -P -m 1
sqoop import --connect jdbc:mysql://mlc-testcapstone.cyaielc9bmnf.us-east-1.rds.amazonaws.com:3306/mlctest --table events --target-dir /user/hadoop/mlctest/events --username student -P -m 1
sqoop import --connect jdbc:mysql://mlc-testcapstone.cyaielc9bmnf.us-east-1.rds.amazonaws.com:3306/mlctest --table train --target-dir /user/hadoop/mlctest/train --username student -P -m 1

Listing hadoop filesystem
==========
hadoop fs -ls mlctest/app_events
hadoop fs -ls mlctest/brand_device
hadoop fs -ls mlctest/events
hadoop fs -ls mlctest/train
hadoop fs -ls mlctest

Connecting to HIVE, creating database and using the same to create HIVE tables
==========
beeline -u jdbc:hive2://localhost:10000/default -n hadoop
create database mlctest;
use mlctest;

Creation of HIVE external Tables
==========
create external table if not exists app_events_external (event_id int, app_id string, is_installed int, is_active int) row format delimited fields terminated by "," lines terminated by "\n" stored as textfile;
create external table if not exists train_external (device_id string, gender string, age int, group_train string) row format delimited fields terminated by "," lines terminated by "\n" stored as textfile;
create external table if not exists brand_device_external (device_id string, phone_brand string, device_model string) row format delimited fields terminated by "," lines terminated by "\n" stored as textfile;
create external table if not exists events_external (event_id int, device_id string, event_time timestamp, latitude float, longitude float) row format delimited fields terminated by "," lines terminated by "\n" stored as textfile;
create external table if not exists app_labels_external (app_id string, label_id int) row format delimited fields terminated by "," lines terminated by "\n" stored as textfile TBLPROPERTIES("skip.header.line.count"="1");
create external table if not exists label_categories_external (label_id int, category string) row format delimited fields terminated by "," lines terminated by "\n" stored as textfile TBLPROPERTIES("skip.header.line.count"="1");

Load into Hive tables from HDFS and validate Data in the tables
==========
load data inpath '/user/hadoop/mlctest/app_events' into table app_events_external;
select * from app_events_external limit 5;
load data inpath '/user/hadoop/mlctest/brand_device' into table brand_device_external;
select * from brand_device_external limit 5;

```
load data inpath '/user/hadoop/mlctest/events' into table events_external;
select * from events_external limit 5;
load data inpath '/user/hadoop/mlctest/train' into table train_external;
select * from train_external limit 5;
```

Load Data into Hive tables from local files

==========

```
load data local inpath '/home/hadoop/app_labels_new.txt' into table app_labels_external;
select * from app_lables_external limit 5;
load data local inpath '/home/hadoop/label_categories.csv' into table label_categories_external;
select * from label_categories_external limit 5;
```

HQL Tasks

==========

1. Which are the top 10 most popular brands and respective % for Male and Female in it ? [Do handle the device_id duplicates from brand_device table]

==========

==========

```
SELECT b.phone_brand AS Phone_Brand,
       Count(*)           AS Total,
       Sum(CASE t.gender
            WHEN 'M' THEN 1
            ELSE 0
          end) * 100 / Count(*) AS male_pct,
       Sum(CASE t.gender
            WHEN 'F' THEN 1
            ELSE 0
          end) * 100 / Count(*) AS female_pct
FROM   (SELECT *
        FROM   train_external) t
       JOIN (SELECT DISTINCT( device_id ),
                    phone_brand
            FROM   brand_device_external) b
         ON t.device_id = b.device_id
GROUP  BY b.phone_brand
ORDER  BY total DESC
LIMIT  10;
```

2. Which are the top 10 most popular brands for Male and Female ? [Do handle the device_id duplicates from brand_device dataset]

==========

==========

```
SELECT b.phone_brand as Phone_Brand,
       Count(*) AS Total,
       t.gender as Gender
FROM   (SELECT *
        FROM   train_external
        WHERE  gender = 'M') t
       JOIN (SELECT DISTINCT( device_id ),
                    phone_brand
            FROM   brand_device_external) b
         ON t.device_id = b.device_id
GROUP  BY b.phone_brand, t.gender
ORDER  BY total DESC
LIMIT  10;

SELECT b.phone_brand AS Phone_Brand,
       Count(*)      AS Total,
       t.gender      AS Gender
```

```
FROM   (SELECT *
        FROM   train_external
        WHERE  gender = 'F') t
       JOIN (SELECT DISTINCT( device_id ),
                    phone_brand
             FROM   brand_device_external) b
         ON t.device_id = b.device_id
GROUP  BY b.phone_brand,
          t.gender
ORDER  BY total DESC
LIMIT  10;
```

3. Count and percentage Analysis of the Gender in the train Dataset

```
==========================================================
SELECT SUM(IF(gender = 'M', 1, 0)) AS male_count,
       Round(( SUM(IF(gender = 'M', 1, 0)) / Count(1) ) * 100, 2)
       ||'%'                       AS male_ratio,
       SUM(IF(gender = 'F', 1, 0)) AS female_count,
       Round(( SUM(IF(gender = 'F', 1, 0)) / Count(1) ) * 100, 2)
       ||'%'                       AS female_ratio
FROM   train_external;
```

4. Top mobile phone brands offering the highest number of models [Give top three brands]

```
=================================================================
===============
select phone_brand, count(device_model) as model_count from brand_device_external group by
phone_brand order by model_count desc limit 3;
```

5. Average number of events per device id [Applicable to device_id from train table which have atleast one associated event in the event table]

```
=================================================================
=========================================================
```

5.5.1 Overall Average events across devices

```
==========================================
SELECT Round(Count(DISTINCT( event_id )) / Count(DISTINCT( device_id ))) AS
       avg_event_per_device
FROM   events_external
WHERE  device_id IN (SELECT DISTINCT( train.device_id ) AS device_id
                     FROM   train_external AS train
                            INNER JOIN events_external AS events
                                    ON train.device_id = events.device_id);
```

5.5.2 Average events per device

```
==================================
SELECT device_id,
       Count(DISTINCT( event_id )) avg_event_per_device
FROM   events_external
WHERE  device_id IN (SELECT DISTINCT( train.device_id ) AS device_id
                     FROM   train_external AS train
                            INNER JOIN events_external AS events
                                    ON train.device_id = events.device_id)
GROUP  BY device_id
ORDER  BY avg_event_per_device DESC
LIMIT  10;
```

6. Count and percentage of device_id in train table have corresponding events data available?

```
==========================================================================
===============
SELECT Max(IF(device_type = 'event_device_id', event_device_count, 0)) AS
       event_device,
       Round(( ( Max(IF(device_type = 'event_device_id', event_device_count, 0))
                 / Max(
                 IF(
                       device_type = 'all', event_device_count, 0)) ) * 100 ),
       2)
       ||'%'                                          AS
       event_device_pct,
       Max(IF(device_type = 'all', event_device_count, 0))           AS
       total_device
FROM   (SELECT 'event_device_id'                 AS device_type,
               Count(DISTINCT( train.device_id )) AS event_device_count
        FROM   train_external AS train
               inner join events_external AS EVENTS
                       ON train.device_id = EVENTS.device_id
        UNION
        SELECT 'all'                       AS device_type,
               Count(DISTINCT( device_id )) AS total_device_count
        FROM   train_external) sub;
```

```
=============================================
TASK 2 - DATA PREPARATION FOR MODELLING
=============================================
```

Creation of external tables and loading data

```
=======================================
```

```
create external table if not exists non_event_data_external (device_id string, phone_brand string,
device_model string, gender string, age int, group_train string) row format delimited fields
terminated by "," lines terminated by "\n" stored as textfile;

insert overwrite table non_event_data_external
select tr.device_id, br.phone_brand, br.device_model, tr.gender, tr.age, tr.group_train
from brand_device_external br
inner join train_external tr
on tr.device_id = br.device_id;


create external table if not exists events_train_external (device_id string, event_id int, event_time
timestamp, latitude float, longitude float, gender string, age int, group_train string) row format
delimited fields terminated by "," lines terminated by "\n" stored as textfile;


insert overwrite table events_train_external
select ev.device_id, ev.event_id, ev.event_time, ev.latitude, ev.longitude,
tr.gender,tr.age, tr.group_train
from  events_external ev
inner join train_external tr
on ev.device_id = tr.device_id;

create external table if not exists app_data_external (event_id int, app_id string, is_installed int,
is_active int, label_id int, category string) row format delimited fields terminated by "," lines
terminated by "\n" stored as textfile;

insert overwrite table app_data_external
```

```
select app_eve.event_id, app_eve.app_id, app_eve.is_installed, app_eve.is_active, lbl.label_id,
lbl.category
from app_events_external app_eve
join app_labels_external app_lbl
on app_eve.app_id = app_lbl.app_id
join label_categories_external lbl
on lbl.label_id = app_lbl.label_id;

select count(*) from app_data_external
```

CSV File Creation from external HIVE Tables

```
==========================================
hive -e 'set hive.cli.print.header=true; select * from mlctest.non_event_data_external' | sed 's/[\t]/,/
g'  > /home/hadoop/non_events.csv;
hive -e 'set hive.cli.print.header=true; select * from mlctest.events_train_external' | sed 's/[\t]/,/g'
> /home/hadoop/events.csv;
hive -e 'set hive.cli.print.header=true; select * from mlctest.app_data_external' | sed 's/[\t]/,/g'  > /
home/hadoop/appdata.csv;
```

Copying the CSV file to S3 Bucket

```
==============================

aws s3 cp non_events.csv s3://capstone-sanjanameghna/non_events.csv;
aws s3 cp events.csv s3://capstone-sanjanameghna/events.csv;
aws s3 cp appdata.csv s3://capstone-sanjanameghna/appdata.csv;
```