Checkpoints

Checkpoint 1

Load the data into HDFS, Hive Managed table, Hive External table and Spark DataFrame.

1. Commit the screenshot of the view/result of the top 25 rows from each individual store (HDFS, Hive – Managed/External and Spark DataFrame).

[cloudera@quickstart ~]$ hdfs dfs –ls

[cloudera@quickstart ~]$ hdfs dfs -put aadhar.csv /user/cloudera

hive> drop database if exists aadhar_db;

OK

Time taken: 0.364 seconds

hive> create database if not exists aadhar_db;

OK

Time taken: 3.404 seconds

hive> create table if not exists aadhar(registrar varchar(100), private_agency varchar(100), state varchar(50), district varchar(50), sub_district varchar(50), pin_code int, gender char(2), age int, aadhar_generated int, enrollment_rejected int, email_id int, mobile_no int) row format delimited fields terminated by ',' stored as textfile;

OK

Time taken: 0.5 seconds

hive> describe formatted aadhar;

OK

 # col_name data_type comment

registrar varchar(100)

private_agency varchar(100)

state varchar(50)

district varchar(50)
sub_district varchar(50)
pin_code int
gender char(2)
age int
aadhar_generated int
enrollment_rejected int
email_id int
mobile_no int

 # Detailed Table Information
Database: default
Owner: cloudera
CreateTime: Thu Aug 08 21:28:52 PDT 2019
LastAccessTime: UNKNOWN
Protect Mode: None
Retention: 0
Location: hdfs://quickstart.cloudera:8020/user/hive/warehouse/aadhar
Table Type: MANAGED_TABLE
Table Parameters:
transient_lastDdlTime 1565324932

 # Storage Information
SerDe Library: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat: org.apache.hadoop.mapred.TextInputFormat
OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed: No
Num Buckets: -1
Bucket Columns: []
Sort Columns: []
Storage Desc Params:
field.delim ,
serialization.format ,
Time taken: 0.235 seconds, Fetched: 38 row(s)

hive> load data inpath '/user/cloudera/aadhar.csv' into table aadhar;

Loading data to table default.aadhar

Table default.aadhar stats: [numFiles=1, totalSize=46483335]

OK

Time taken: 0.53 seconds

hive> create external table if not exists aadhar1(registrar varchar(100), private_agency varchar(100), state varchar(50), district varchar(50), sub_district varchar(50), pin_code int, gender char(2), age int, aadhar_generated int, enrollment_rejected int, email_id int, mobile_no int) row format delimited fields terminated by ',' stored as textfile location '/user/cloudera/aadhar.csv';

OK

Time taken: 0.076 seconds

hive> describe formatted aadhar1;

OK

 # col_name data_type comment

registrar varchar(100)

private_agency varchar(100)

state varchar(50)

district varchar(50)

sub_district varchar(50)

pin_code int

gender char(2)

age int

aadhar_generated int

enrollment_rejected int

email_id int

mobile_no int

 # Detailed Table Information

Database: default

Owner: *cloudera*

CreateTime: *Thu Aug 08 21:35:35 PDT 2019*

LastAccessTime: *UNKNOWN*

Protect Mode: *None*

Retention: *0*

Location: *hdfs://quickstart.cloudera:8020/user/cloudera/aadhar.csv*

Table Type: *EXTERNAL_TABLE*

Table Parameters:

EXTERNAL *TRUE*

transient_lastDdlTime *1565325335*

 # Storage Information

SerDe Library: *org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe*

InputFormat: *org.apache.hadoop.mapred.TextInputFormat*

OutputFormat: *org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat*

Compressed: *No*

Num Buckets: *-1*

Bucket Columns: *[]*

Sort Columns: *[]*

Storage Desc Params:

field.delim *,*

serialization.format *,*

Time taken: 0.089 seconds, Fetched: 39 row(s)

scala> import org.apache.spark.sql.hive.HiveContext

scala> val hc = new org.apache.spark.sql.hive.HiveContext(sc)

*scala> val aadhardf = hc.sql("select * from aadhar")*

aadhardf: org.apache.spark.sql.DataFrame = [registrar: string, private_agency: string, state: string, district: string, sub_district: string, pin_code: int, gender: string, age: int, aadhar_generated: int, enrollment_rejected: int, email_id: int, mobile_no: int]

scala> val RDD=sc.textFile("/user/cloudera/aadhar.csv")

scala> val first=RDD.first()

scala> val filterRDD = RDD.filter(w=>w!=first)

scala> val Aadhar=filterRDD.map(x=>(x.split(",")(0),x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4),x.split(",")(5),x.split(",")(6),x.split(",")(7).toInt,x.split(",")(8).toInt,x.split(",")(9).toInt,x.split(",")(10).toInt,x.split(",")(11).toInt))

scala> val aadhardf = Aadhar.toDF("registrar","private_agency","state","district","sub_district","pin_code","gender","age","aadhar_generated","rejected","mobile_no","email_id");

Checkpoint 2

2. Describe the schema.

hive> describe aadhar;

OK

registrar varchar(100)

private_agency varchar(100)

state varchar(50)

district varchar(50)

sub_district varchar(50)

pin_code int

gender char(2)

age int

aadhar_generated int

enrollment_rejected int

email_id int

mobile_no int

3. Find the count and names of registrars in the table.

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select registrar, count(registrar) from aadhar group by registrar;

4. Find the number of states, districts in each state and sub-districts in each district.

hive> select state, count(state), district, count(district), sub_district count(sub_district) over (partition by state,district) from aadhar group by state, district,sub_district;

5. Find the number of males and females in each state from the table and display a suitable plot.

Hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select state,gender, count(gender) from aadhar group by state,gender;

6. Find out the names of private agencies for each state.

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select state,private_agency, count(private_agency) from aadhar group by state,private_agency;

7. Plot the number of private agencies for each state.

Checkpoint 3

8. Find top 3 states generating most number of Aadhaar cards?

hive> create table aadhar_generated as select state,sum(aadhar_generated) as sum_generation from aadhar group by state;

Hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select * from aadhar_generated order by sum_generation desc limit 3;

9. Find top 3 private agencies generating the most number of Aadhar cards?

hive> create table aadhar_generated_pr as select private_agency,sum(aadhar_generated) as sum_generation from aadhar group by private_agency;

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select * from aadhar_generated_pr order by sum_generation desc limit 3;

10. Find the number of residents providing email, mobile number? (Hint: consider non-zero values.)

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select count(email_id) from aadhar where email_id is NOT NULL and mobile_no is NOT NULL;

11. Find top 3 districts where enrolment numbers are maximum?

hive> create table aadhar_generated_district as select district,sum(aadhar_generated) as sum_generation from aadhar group by district;

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select * from aadhar_generated_district order by sum_generation desc limit 3;

12. Find the no. of Aadhaar cards generated in each state?

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select state, sum(aadhar_generated) as aadhar_generated_by_state from aadhar group by state;

Checkpoint 4

13. Create a data frame using the file and provide its summary.

scala> aadhardf.describe().show()

```
+-------+------------------+-----------------+------------------+------------------+-------------+
|summary|               age| aadhar_generated|          rejected|         mobile_no|     email_id|
+-------+------------------+-----------------+------------------+------------------+-------------+
|  count|            440818|           440818|            440818|            440818|       440818|
|   mean|19.704367788974135|1.6014296149431284|0.08751003815633662|0.04415427682172688|1.0544964134858377|
| stddev|18.686811059770278| 3.391819119747009|0.40708726865347666|0.23721206910475311|1.5477642589293523|
|    min|                 0|                0|                 0|                 0|            0|
|    max|               118|              391|                40|                15|           93|
+-------+------------------+-----------------+------------------+------------------+-------------+
```

14. Write a command to see the correlation between “age” and “mobile_number”? (Hint: Consider the percentage of people who have provided the mobile number out of the total applicants)

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select corr(age, mobile_no) from aadhar;

15. Find the number of unique pincodes in the data?

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select distinct(pin_code) from aadhar;

16. Find the number of Aadhaar registrations rejected in Uttar Pradesh and Maharashtra?

Hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select state, sum(enrollment_rejected) from aadhar where state like "%Uttar P%" or state like "%Mahar%" group by state;

Checkpoint 5

On the given dataset, perform EDA and find:

17. The top 3 states where the percentage of Aadhaar cards being generated for males is the highest.

hive> create table male_percent as select state, round((sum(aadhar_generated)/sum(aadhar_generated+enrollment_rejected))*100,2) as percentage from aadhar where gender = 'M' group by state;

hive> select * from male_percent order by percentage desc limit 3;

18. In each of these 3 states, identify the top 3 districts where the percentage of Aadhaar cards being rejected for females is the highest.

hive> create table female_top_district as select district, round((sum(enrollment_rejected)/sum(aadhar_generated+enrollment_rejected))*100,2) as percentage from aadhar where gender = 'F' and state in ("Andaman and Nicobar Islands", "Others", "Lakshadweep") group by district;

hive> insert overwrite local directory "/home/cloudera/hive/aadhar" row format delimited fields terminated by ',' stored as textfile select * from female_top_district order by percentage desc limit 3;

19. The top 3 states where the percentage of Aadhaar cards being generated for females is the highest.

hive> create table female_top_state as select state, round((sum(aadhar_generated)/sum(aadhar_generated+enrollment_rejected)

)*100,2) as percentage from aadhar where gender = 'F' group by state;

hive> insert overwrite local directory "/home/cloudera/hive/aadhar" row format delimited fields terminated by ',' stored as textfile select * from female_top_state order by percentage desc limit 3;

20. In each of these 3 states, identify the top 3 districts where the percentage of Aadhaar cards being rejected for males is the highest.

hive> create table male_top as select district, round((sum(enrollment_rejected)/sum(aadhar_generated+enrollment_rejected))*100,2) as percentage from aadhar where gender = 'M' and state in ("Dadra and Nagar Haveli", "Others", "Sikkim") group by district;

hive> insert overwrite local directory "/home/cloudera/hive/aadhar" row format delimited fields terminated by ',' stored as textfile select * from male_top order by percentage desc limit 3;

21. The summary of the acceptance percentage of all the Aadhaar cards applications by bucketing the age group into 10 buckets.

Hive>create table aadhar_bucket(registrar string,private_agency string,state string,district string,sub_district string,pincode string,gender string, age int,aadhar_generated int,rejected int,email_id int,moblie_number int) clustered by (age) into 10 buckets row format delimited fields terminated by ',' stored as textfile TBLPROPERTIES('serialization.null.format'='','skip.header.line.count'='1');

hive> insert overwrite local directory '/home/cloudera/hive/aadhar' row format delimited fields terminated by ',' stored as textfile select round((sum(rejected)/sum(aadhar_generated+rejected))*100,2) from aadhar_bucket;