|  |
| --- |
| **Project**  **Data Set : India Annual Health Survey (AHS) 2012-13** |

The dataset comprises a survey conducted in Empowered Action Group (EAG) states Uttarakhand, Rajasthan, Uttar Pradesh, Bihar, Jharkhand, Odisha, Chhattisgarh & Madhya Pradesh and Assam. These nine states, which account for about 48 percentage of the total population, 59 percentage of Births, 70 percentage of Infant Deaths, 75 percentage of Under 5 Deaths and 62 percentage of Maternal Deaths in the country, are the high focus States in view of their relatively higher fertility and mortality. A representative sample of about 21 million population and 4.32 million households were covered which is spread across the rural and urban area of these 9 states. The objective of the AHS is to yield a comprehensive, representative and reliable dataset on core vital indicators including composite ones like Infant Mortality Rate, Maternal Mortality Ratio and Total Fertility Rate along with their covariates (process and outcome indicators) at the district level and map the changes therein on an annual basis. These benchmarks would help in better and holistic understanding and timely monitoring of various determinants on well-being and health of population particularly Reproductive and Child Health.

[**Source**]🡪 <http://pib.nic.in/newsite/mbErel.aspx?relid=85350>

|  |
| --- |
| **Content** |

This dataset contains the data about the below 26 key indicators.

|  |  |
| --- | --- |
| **Key Indicatorsfile** |  |

**Acknowledgements:**

Department of Health and Family Welfare, Govt. of India has published this data in Open Govt Data Platform India portal under Govt. Open Data License - India.

References:

<https://nrhm-mis.nic.in/SitePages/Home.aspx>

<https://data.gov.in/catalog/indicators-annual-health-survey>

<https://data.gov.in/government-open-data-license-india>

**Problem Statement**

Ingest the India Annual Health Survey (AHS) 2012-13 data hosted on Amazon RDS into Hadoop correctly and process it to generate the following analyses:

**Analyses**

1. The child mortality rate in Uttar Pradesh
2. The fertility rate in Bihar
3. State-wise child mortality rate and state-wise child fertility rate and does high fertility correlate with high child mortality?
4. Find top 2 districts per state with the highest population per household
5. Find top 2 districts per state with the lowest sex ratios

Such analyses would help in vivid understanding and timely monitoring of different determinants on well-being and health of population particularly Child and Reproductive Health. Based on the analyses, one can also compare India’s position in Global HDI and can suggest ways that can improve it.

<https://en.wikipedia.org/wiki/List_of_countries_by_Human_Development_Index>

**Guidelines**

* Ingest data from Amazon RDS to HDFS using Sqoop.
* Create an external table in Hive for the ingested data containing all the columns as given in this document. Ingest the data from HDFS to the Hive table. Verify that the ingestion is successfully accomplished.

|  |  |
| --- | --- |
| Columnsfile |  |
| Sample Data Corresponding To The Columnsfile |  |

* Create a subset schema in Hive to store the data for the analyses to be done. The schema should be optimized to support ONLY the analyses to be done. You will be graded on your choice of the chosen columns, storage format (Parquet, RC, ORC, CSV), etc. Benchmark the performance of the storage formats before finalizing the one to be used.
* Write queries against each category of analyses. You will be graded on the relevance of your query to the analytical use case and the optimizations used. Generate the corresponding analyses’ charts on Hue.

**Note**: To access Amazon RDS, refer to the resources section for more details.

**Note**: The size of the dataset is around 2.5 MB. This is a representative sample and the actual dataset will be of a bigger size. This sample is specifically taken keeping in mind that the engineering process for the data of any size remains the same. Some optimizations might vary as the dataset grows larger. However, while designing the solution, keep optimization in mind and submit a solution that would work even if we increase the size of this dataset.

**Tasks**

**Data Ingestion from the RDS to HDFS using Sqoop**

Sqoop import command

Command to see the list of imported data

**External table creation in Hive and loading the ingested data into it. Data ingestion verification.**

Command to create the external table

Command to load the ingested data into the external table

Queries to verify that the ingestion is correctly accomplished

Query to count the total number of rows along with the screenshots of the data fetched by the query on MySQL Workbench and Hue

Query to select the top 10 rows and first 8 columns along with the screenshots of the data fetched by the query on MySQL Workbench and Hue

**Subset schema creation in Hive to support the analyses**

Columns used in the subset schema

Storage format used

[Benchmark the performance before finalizing the storage format to be used. Create one schema using default format and one in any other format such as ORC for the columns to be used. Insert data into both the tables created. Compare the runtimes of the following queries and decide which format to be used.

select count(\*) from <Table Name>;

select State\_Name, count(\*) from <Table Name> group by State\_Name;

select \* from <Table Name> where State\_Name = ‘Uttar Pradesh’;]

Create and insert command for the default format

Create and insert command for the formats such as ORC

Screenshot of runtimes against each query given above for the default format as well as for the formats such as ORC

Create and insert command for the partition table for analyses 1 & 2. The partition table should be created using the table created above.

**Note:** While doing the benchmarking, you might get the run times of formats such as ORC different than expected. These anomalies exist if the dataset is small. The main idea is to let you know that benchmarking is an important step in the end to end engineering process. Generally, you would benchmark the performance against different formats and choose the best. However, here go ahead with the formats such as ORC only.

**The result of each analysis along with the query and the corresponding chart generated in Hue. Keep optimizations in mind**

1. The child mortality rate of Uttar Pradesh

* Query
* Screenshot of the result
* Chart

1. The fertility rate of Bihar

* Query
* Screenshot of the result
* Chart

1. State wise child mortality rate and state wise child mortality rate and does high fertility correlate with high child mortality?

* Query
* Screenshot of the result
* Chart

1. Find top 2 districts per state with the highest population per household

* Query
* Screenshot of the result
* Chart

1. Find top 2 districts per state with the lowest sex ratios

* Query
* Screenshot of the result
* Chart

**Note**: For doing the analyses, prefer directly using the relevant columns rather than the computations. Your grading will not be affected if you directly use the relevant columns. Moreover, you will not get extra marks for doing unnecessary computations.

The analyses asked can be done directly using the relevant columns only. Approximations such as using the average function can be used.

The primary purpose is to get you familiar with the end to end engineering process. So the focus is not on the computations.

Refer to the sample solution given in Submission segment of the next Submission Guidelines session for getting more clarity on the things to be submitted.

**Resources**

Once you are connected to the RDS, you can execute the basic SQL queries to analyze the data.

Once you have got an idea about the data, you can import the same to your Hadoop using Sqoop. You can do using the same Sqoop import command taught earlier. Just remember to provide the correct --connect, --username and --password in the command as told in the note above.

**Note:** Close the connection once you are done to avoid issues such as connection limit exceeded.

|  |  |
| --- | --- |
| **Sqoop Import** | <https://sqoop.apache.org/docs/1.4.2/SqoopUserGuide.html#_controlling_the_import_process> |
| **Create Table in Hive** | <https://www.cloudera.com/documentation/enterprise/5-8-x/topics/impala_create_table.html> |
| **Load Data in Hive** | <https://www.cloudera.com/documentation/enterprise/5-2-x/topics/impala_load_data.html> |
| **Hue** | <https://www.cloudera.com/documentation/enterprise/5-9-x/topics/hue.html> |

**Additional Reading**

**Sqoop**: Import Data From MySQL to Hive :

<https://dzone.com/articles/sqoop-import-data-from-mysql-to-hive>

**Submission Guidelines**

**Grading Rubrics**

| **Criteria** | **Meet expectations** | **Does not meet expectations** |
| --- | --- | --- |
| Data Ingestion(10%) | * Has imported data from Amazon RDS to HDFS using Sqoop * Has used proper parameters in the import command of Sqoop * Has been able to list the imported data inside HDFS | * Could not import data to HDFS using Sqoop * Could not use proper parameters in the import command of Sqoop * Could not list the imported data inside HDFS |
| External table creation in Hive and loading the ingested data into it. Data ingestion verification (25%) | * Creates external table successfully * Load data into the external table successfully * Ingested data is successfully verified * The code is appropriately commented | * Table creation fails or contains unnecessary columns or misses the required ones * Data loading into the table from HDFS fails * Data not ingested correctly * The code is not commented |
| Subset schema creation in Hive to support the analyses (25%) | * Correctly identifies the columns to be used for the analyses * Correctly identifies the storage format to be used * Correctly creates the table with proper data types for default format and formats such as ORC * Correctly inserts data into them * Run Queries to do performance benchmarking * The partition table is correctly created using the table created above for analyses 1 and 2 | * Misses or use unnecessary columns * Don't use the correct storage format * Tables not created properly * Data types not properly defined * Data not inserted properly * Don't do performance benchmarking * The partition table is not created correctly |
| Analyses (40%) | * The queries are syntactically correct * The results do not contain unnecessary columns and have all the required columns * The group by, order by clauses and subqueries are correctly used * The analyses 1 & 2 are solved using the partition table * UDFs and UDAFs are used wherever appropriate for concise querying * Proper optimizations are used * The code is appropriately commented | * The queries contain syntax errors, the results contain either unnecessary columns or miss the required ones * Has used complicated methods/functions to arrive at the result * The analyses 1 & 2 are not solved using the partition table * UDFs and UDAFs are not used even if required * Optimizations not done * The code is not commented properly |

**Submission**

The deadline for the submission is December 9, 2018, 11:59 PM.

**Submissions required**

Upload a PDF document containing the tasks listed in Problem Statement segment of the previous Project session.

**Sample File:** Sample+Solution.pdf

