Analyzing US birth data from 2016 to 2021

Bellevue University

DSC 650 – Big Data

Professor Nasheb Ismaily

May 30, 2024

Eyram Kueviakoe

Outline

|  |  |
| --- | --- |
| Introduction ------------------------------------- | - 2- |
| Data Source ------------------------------------- | - 2- |
| Presentation of the dataset ------------------ | - 2- |
| Data Ingestion ---------------------------------- | - 3- |
| Loading data into HDFS ----------------------- | - 3- |
| Using Hive --------------------------------------- | - 5- |
| Using PySpark ---------------------------------- | - 9- |
| Conclusion ------------------------------------- | - 12- |

**Introduction**

Birth rates are a key indicator of population growth and can impact various sectors like healthcare and education. Understanding birth trends is important to improve the educational system and healthcare. It is also useful in making policies.

This project analyzes birth data from 2016 to 2021in the US to discover trends and patterns. We will be focusing on how the educational level of parents affects birth rates.

During this project, we will use big data tools to handle and analyze the dataset efficiently. We will use HDFS to store the data, spark to process the data, and Hive to query the data.

**Data source**

The dataset used for this project was obtained from Kaggle. It contains detailed information about the birth records from 2016 to 2021. It includes information on the number of births grouped by year, state and the educational level of the parents. The goal of the project is to use big data tools to analyze and query the data.

The link to the dataset is: <https://www.kaggle.com/datasets/danbraswell/temporary-us-births?select=us_births_2016_2021.csv>

**Presentation of the dataset**

**Columns description:**

**State**: Full name of the state

**State Abbreviation**: 2-character abbreviation of the state.

**Year**: The 4-digit year in which the births were recorded.

**Gender**: The gender of the baby (Male/Female).

**Education Level of Mother**: The education level of the mother.

**Education Level Code**: The corresponding code for the education level of the mother

**Number of Births**: The number of births for the specified category.

**Average Age of Mother (years):** The average age of the mother in the specified category.

**Average Birth Weight (g):** The average birth weight in grams for the specified category.

Showing a sample of the dataset (with relevant columns)

A close-up of a document

Description automatically generated

**Data Ingestion**

The first step of this project is to import the dataset into HDFS (Hadoop Distributed File System). To achieve this, we will connect to the HDFS instance in our google cloud virtual machine and upload the csv file containing the data.

Step 1: We download the csv file from Kaggle and upload it to our github repository.

Step 2: Uploading the csv file into our virtual machine using the command

*wget https://raw.githubusercontent.com/kueyram/dsc650/main/us\_births\_2016\_2021.csv*A screenshot of a computer screen

Description automatically generated

*Fig1: Downloading the csv file onto the virtual machine*

Step 3: Starting the Docker container and then accessing the master container

A computer screen with text on it

Description automatically generated

*Fig2: Starting docker container*

Step 4: Load the csv file into HDFS

A screenshot of a computer

Description automatically generated

*Fig3: Loading the csv file into hdfs*

Step 5: Checking if the file was successfully uploaded.

A screenshot of a computer

Description automatically generated

*Fig4: Showing the csv was uploaded into hdfs*

**Create table and load data in Hive**

Step6: Let’s start a hive session and create a table using this command

*CREATE TABLE birth\_data(*

*`State` STRING,*

*`State Abbreviation` STRING,*

*`Year` INT,*

*`Gender` STRING,*

*`Education Level of Mother` STRING,*

*`Education Level Code` INT,*

*`Number of Births` INT,*

*`Average Age` FLOAT,*

*`Average Birth Weight (g)` FLOAT)*

*ROW FORMAT DELIMITED*

*FIELDS TERMINATED BY ','*

*STORED AS TEXTFILE*

*tblproperties("skip.header.line.count"="1");*

A screenshot of a computer

Description automatically generated

*Fig5: Accessing Hive and creating a table to store the data*

Step 7: Loading the data into the Hive table:

A screenshot of a computer

Description automatically generated

*Fig 6: Loading the data into the Hive table*

Step 8: Running queries on the data

* Query 1: Let’s count the number of rows in the dataset

*SELECT COUNT(\*) AS NumberRows FROM birth\_data;*

*A screen shot of a computer

Description automatically generated*

*Fig 7: Number of rows in the dataset*

*We have 5496 rows in the dataset*

* Query 2: Let’s count the number of births in the dataset

*SELECT SUM(`Number of Births`) AS NumberBirths FROM birth\_data;*

A computer screen shot of a black screen

Description automatically generated

*Fig8: Number of births in the dataset*

There are 8889084 births in the dataset

* Query 3: Number of births in each state in 2020

*SELECT State, SUM(`Number of Births`) AS total\_births\_2020*

*FROM birth\_data*

*WHERE Year = 2020*

*GROUP BY State;*

A screenshot of a computer

Description automatically generated

*Fig8: Number of births in each state in 2020*

Query 4: Top 5 education level with the highest births

*SELECT `Education Level of Mother`, SUM(`Number of Births`) AS Number\_Births*

*FROM birth\_data*

*GROUP BY `Education Level of Mother`*

*ORDER BY Number\_Births DESC*

*LIMIT 5;*

A screenshot of a computer program

Description automatically generated

*Fig 9: Top 5 education level with the highest births*

**Using Spark**

Step 8: Loading data into spark and creating a dataframe

*birth\_data\_df = spark.read.format('csv').option('header','true').load('us\_births\_2016\_2021.csv')*

Let’s very the dataframe was created.

*Birth\_date\_df.show()*

A screen shot of a computer

Description automatically generated

*Fig 9: Checking that the dataframe was created and populated*

Step 9: Let’s remove the redundant columns from the dataframe

The dataframe has a column called State and another column which has the state names abbreviated. It also has the educational level and the code that corresponds to to the educational level.

We will drop State Abbreviation, and Education Level Code

*birth\_data\_cleaned = birth\_data\_df.drop("State Abbreviation", "Education Level Code")*

Checking the new dataframe

Birth\_data\_cleaned.show()

A screenshot of a computer

Description automatically generated

*Fig 10: New dataframe after redundant columns were removed*

With the cleaned dataframe, we can create visualizations and find possible correlations between the number of births and the education level of the mother.

Step 10: Number of births grouped by education level

*births\_by\_education\_level = birth\_data\_cleaned.groupBy('Education Level of Mother').sum('Number of Births')*

*births\_by\_education\_level = births\_by\_education\_level.withColumnRenamed('sum(Number of Births)', 'Number of Births')*

*births\_by\_education\_level.show()*

A black background with white text

Description automatically generated

*Fig 11: Births grouped by educational level*

**Conclusion**

After ingesting the data into HDFS, we were able to define our table schema in Hive and import the data into the table. This gave us the opportunity to use HiveQL to run different queries on the table.

For advanced data manipulation and transformation, we use PySpark to create a dataframe that can be used to create visualization and graphs. PySpark can be used for data analysis, complex transformation, and machine learning.

The analysis of the data shows that as the mother’s education level rises, there is a decrease in the number of births. This means that there is a possible correlation between higher education among mothers and reduced fertility rates. This can be used by healthcare professionals and educators to promote family planning.

In the future, we could include real-time data processing to ensure the analysis is up-to-date and responsive to changing trends. We could also use machine learning to make predictions. Finally, we could implement advanced visualization and interactive dashboards to present and communicate our findings.