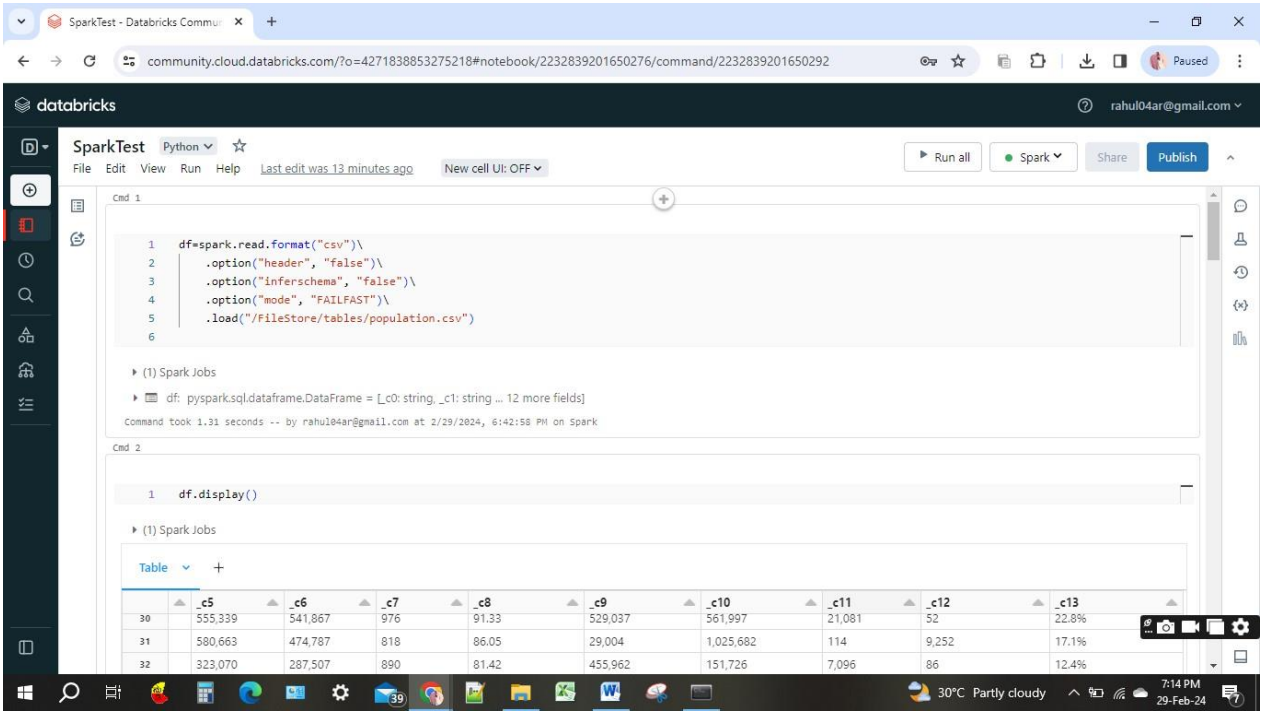
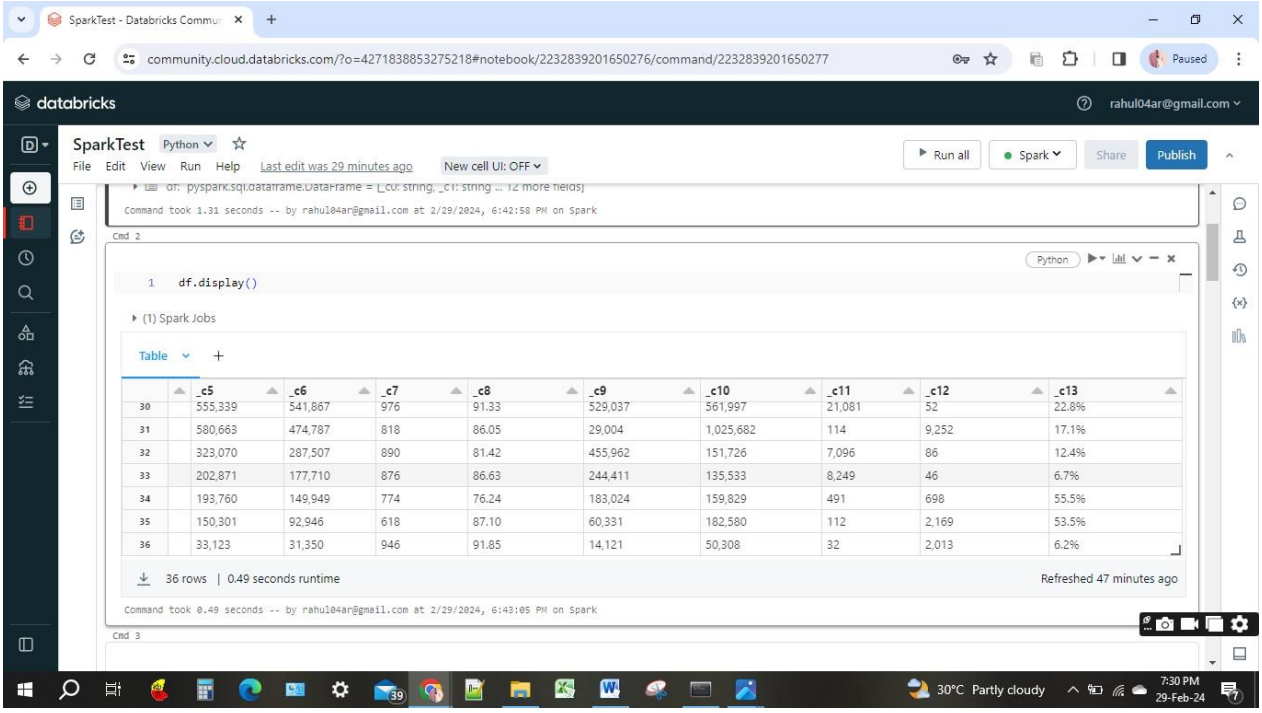
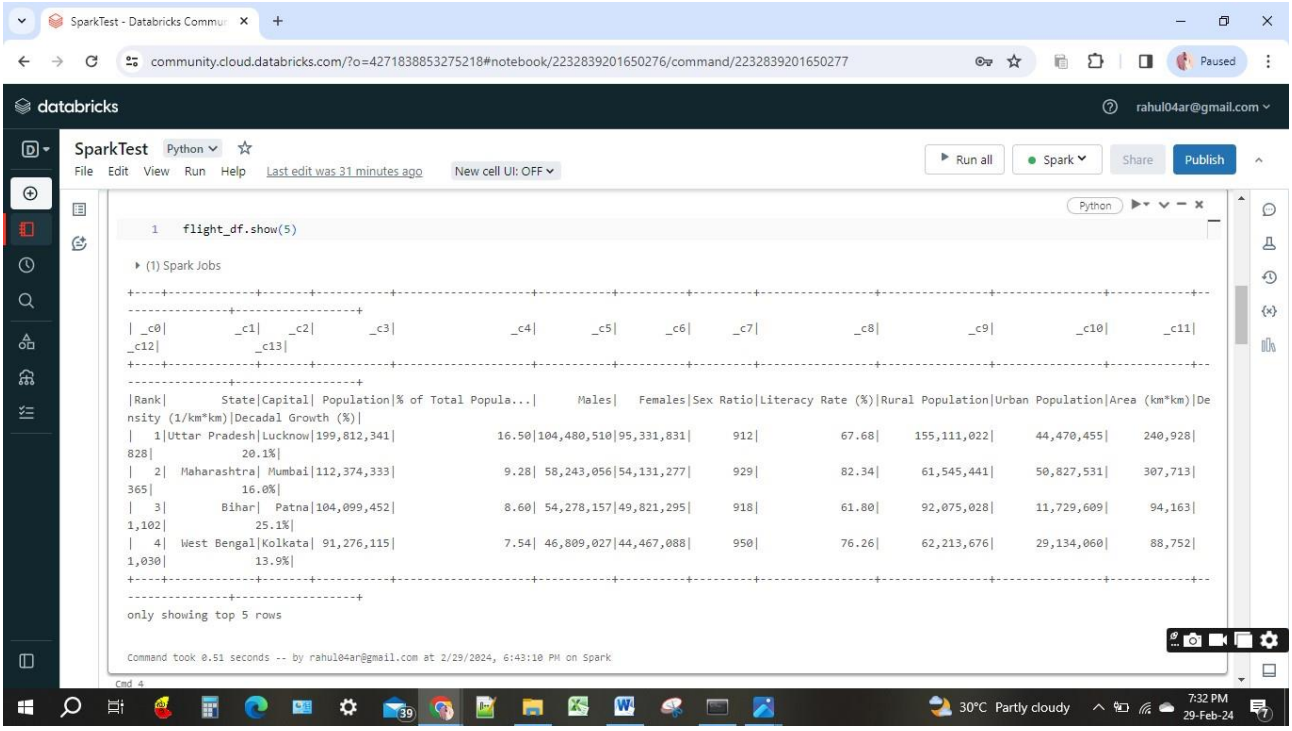
1. READ THE DATASET IN DATABRICKS COMMUNITY

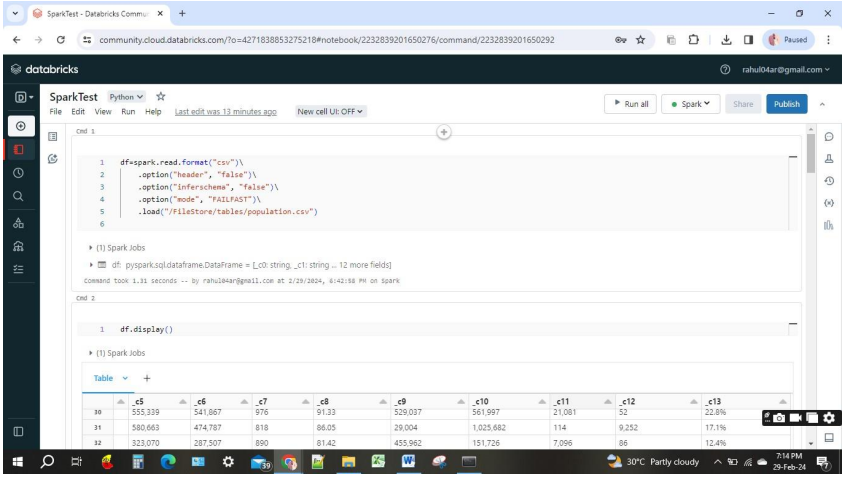






2. HOW MANY TYPES OF MODES WE HAVE IN SPARK?

1. **Failfast Mode**: This mode is used to instruct Spark to fail the entire operation (e.g., reading data into a DataFrame) if it encounters any malformed records or parsing errors. It stops processing the data immediately upon encountering the first error and throws an exception. This mode ensures data integrity but may result in the loss of valid records if the errors are not handled appropriately.
2. **Dropmalformed Mode**: In this mode, Spark skips and drops any rows that contain malformed records or parsing errors while reading data into a DataFrame. It allows you to continue processing the remaining valid records while ignoring the problematic ones. This mode can be useful when you want to prioritize processing valid data and handle errors separately.
3. **Permissive Mode**: Permissive mode is a combination of both failfast and dropmalformed modes. It attempts to parse all records in the input data, even if some records are malformed or contain parsing errors. Spark generates a DataFrame with a column containing the malformed records along with a new column indicating the parsing status (valid or invalid) of each record. This mode allows you to capture and handle parsing errors while still processing the valid data.



3. WHAT IS CLUSTER IN SPARK?

A Spark cluster is a group of interconnected computers (nodes) working together to process data. It consists of a master node, which coordinates tasks, and multiple worker nodes, where data processing occurs. Executors, running on worker nodes, execute tasks assigned by the master node. The driver program, running on the master node, coordinates the overall execution of a Spark application. Cluster managers allocate resources and manage the cluster infrastructure. Spark clusters enable distributed computing, allowing large datasets to be processed efficiently across multiple nodes.

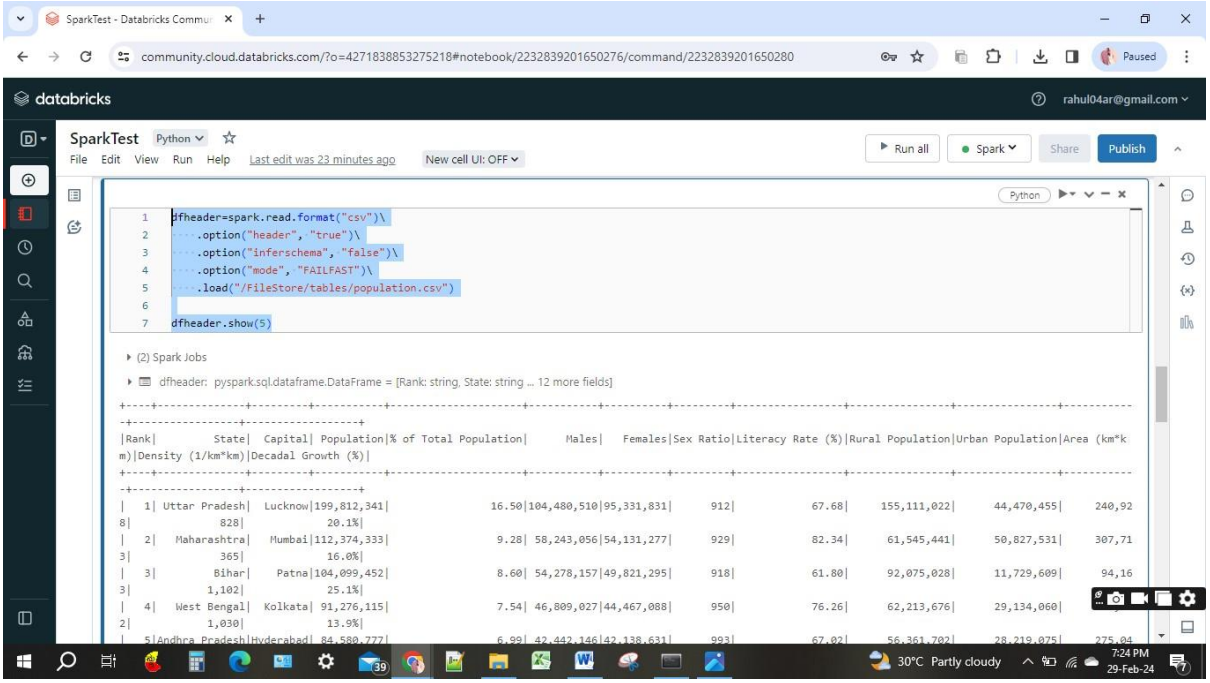
4. WHAT IS TABLE IN SPARK ?

A table in Spark typically refers to structured data organized into rows and columns, represented using Data Frame or Dataset APIs. Tables can be created from various data sources and manipulated using Spark's rich functionality for querying and transforming structured data. They provide a convenient way to work with structured data in Spark, enabling tasks like data analysis, processing, and querying.

5. WHAT WOULD YOU DO IF YOU WANT TO SHOW THE HEADERWHILE SHOWING UP 5 RECORDS OF TABLE? WRITE THE CODE

```
dfheader=spark.read.format("csv")\
    .option("header", "true")\
    .option("inferschema", "false")\
    .option("mode", "FAILFAST")\
    .load("/FileStore/tables/population.csv")

dfheader.show(5)
```



5. WHAT IS COUNT ? PERFORM IN SPARK

In the context of Apache Spark, "count" refers to an operation that calculates the number of elements in a dataset. It is a basic aggregation function used to determine the size of the dataset, whether it's an RDD (Resilient Distributed Dataset), DataFrame, or Dataset.

In Spark, the count() function is available as part of the DataFrame API and RDD API. When called on a DataFrame, it returns the number of rows in the DataFrame. Similarly, when called on an RDD, it returns the total number of elements in the RDD.

SparkTest - Databricks Commu

community.cloud.databricks.com/?o=4271838853275218#notebook/2232839201650276/command/2232839201650280

databricks

rahul04ar@gmail.com

SparkTest Python

File Edit View Run Help Last edit was 25 minutes ago New cell UI: OFF

Run all Spark Share Publish

Command took 0.11 seconds -- by rahul04ar@gmail.com at 2/29/2024, 6:43:36 PM on Spark

Cmd 7

Display number of rows using count( )

```
df.count()
```

(2) Spark Jobs

Out[23]: 36

Command took 0.49 seconds -- by rahul04ar@gmail.com at 2/29/2024, 6:45:30 PM on Spark

Cmd 8

Finding Null Occurrences of each column in data frame using Count( )

```
from pyspark.sql.functions import col,count,when

result = df.select ([count(when(col(c).isNull(),c)).alias(c) for c in df.columns])

display(result)
```

(2) Spark Jobs

result: pyspark.sql.dataframe.DataFrame = [_c0: long, _c1: long ... 12 more fields]

30°C Partly cloudy 7:26 PM 29-Feb-24

SparkTest - Databricks Commu

community.cloud.databricks.com/?o=4271838853275218#notebook/2232839201650276/command/2232839201650280

databricks

rahul04ar@gmail.com

SparkTest Python

File Edit View Run Help Last edit was 26 minutes ago New cell UI: OFF

Run all Spark Share Publish

Cmd 8

Finding Null Occurrences of each column in data frame using Count( )

```
from pyspark.sql.functions import col,count,when

result = df.select ([count(when(col(c).isNull(),c)).alias(c) for c in df.columns])

display(result)
```

(2) Spark Jobs

result: pyspark.sql.dataframe.DataFrame = [_c0: long, _c1: long ... 12 more fields]

Table

	_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	_c8	_c9	_c10
1	0	0	1	0	0	0	0	0	0	0	0

1 row | 1.74 seconds runtime Refreshed 27 minutes ago

Command took 1.74 seconds -- by rahul04ar@gmail.com at 2/29/2024, 7:00:29 PM on Spark

Cmd 9

```
df.groupBy('State')
```

30°C Partly cloudy 7:27 PM 29-Feb-24

6. WHAT IS GROUP BY ? PERFORM IN SPARK

In Apache Spark, "group by" is a data manipulation operation that groups the rows of a DataFrame or Dataset based on one or more columns. It is commonly used for aggregation tasks where you want to group data by certain attributes and perform aggregate functions (such as sum, count, average, etc.) on each group.

The groupBy() function in Spark is used to specify the column(s) by which you want to group the data. After grouping, you typically apply aggregation functions to compute summary statistics or derive new insights from the grouped data.