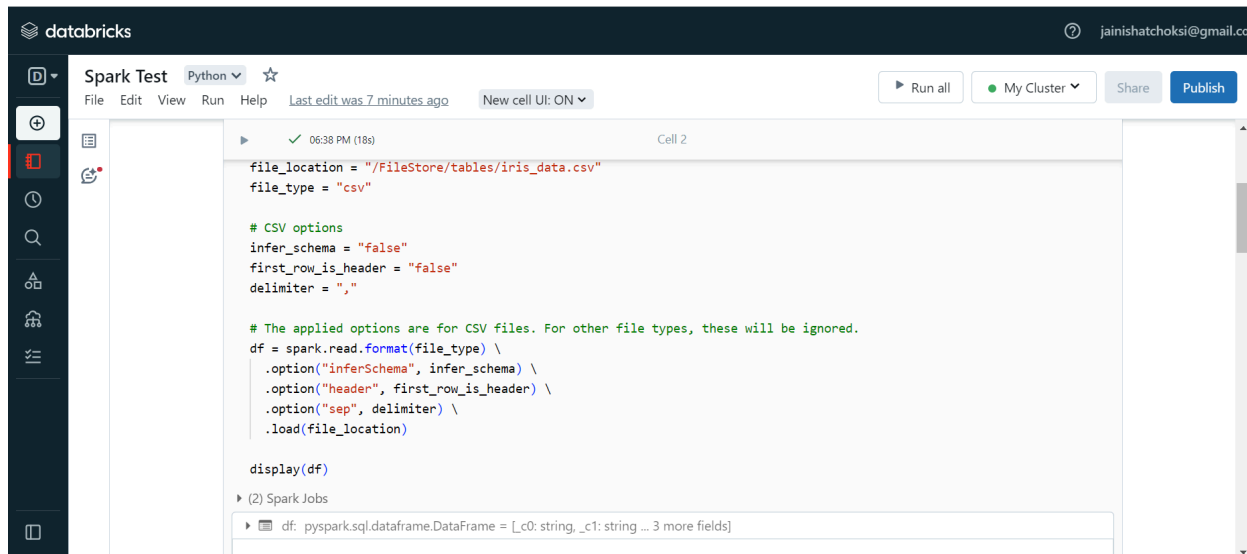


## 1. Read the dataset in databricks community.



The screenshot shows the Databricks Spark Test interface. The code in the cell reads a CSV file from the FileStore, sets options for schema inference and headers, and displays the resulting DataFrame.

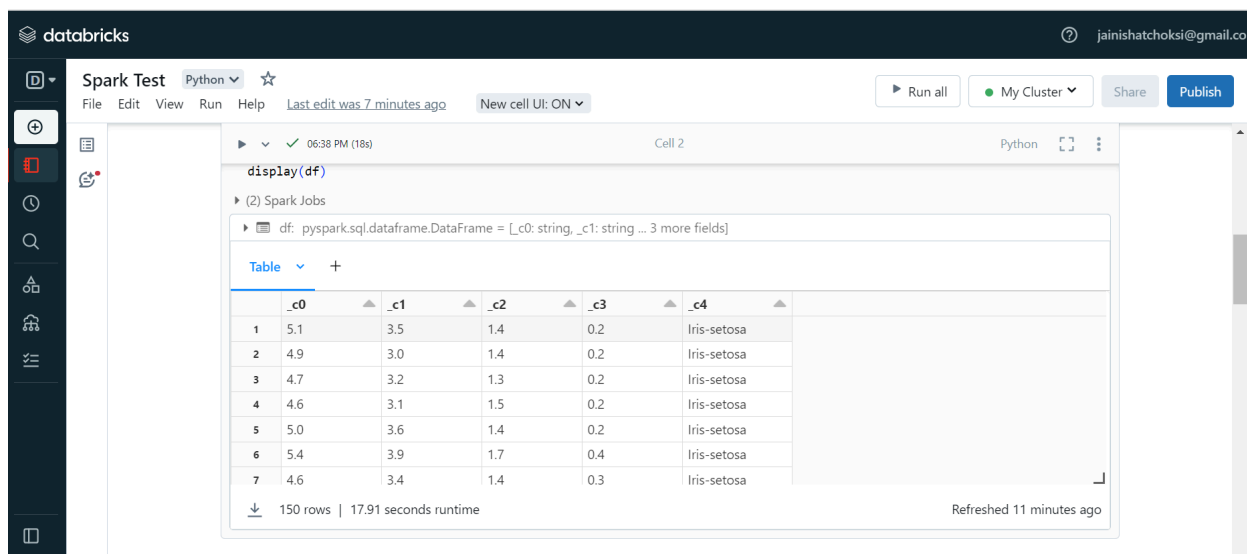
```
file_location = "/FileStore/tables/iris_data.csv"
file_type = "csv"

# CSV options
infer_schema = "false"
first_row_is_header = "false"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
df = spark.read.format(file_type) \
    .option("inferSchema", infer_schema) \
    .option("header", first_row_is_header) \
    .option("sep", delimiter) \
    .load(file_location)

display(df)
```

Below the code, the Spark Jobs section shows the DataFrame schema: `df: pyspark.sql.dataframe.DataFrame = [_c0: string, _c1: string ... 3 more fields]`.



The screenshot shows the Databricks Spark Test interface with the output of the DataFrame displayed as a table. The table has 7 rows and 6 columns. The first column is an index, and the other columns are labeled \_c0 through \_c4. The data represents the Iris dataset.

	_c0	_c1	_c2	_c3	_c4
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa

Below the table, it indicates 150 rows and 17.91 seconds runtime. The table was refreshed 11 minutes ago.

## 2. How many types of modes we have in spark?

In Spark, there are 3 modes:

**Local Mode:** Spark runs on a single machine with a single JVM. Local Mode is suitable for development, testing and small scale data.

**Standalone Mode:** Spark operates in a standalone cluster manager that manages the allocation of resources. It is relatively easy to setup, configure. Suitable for small to medium sized clusters.

**Cluster Mode:** Spark application runs on a cluster managed by external cluster managers like Apache Hadoop, Kubernetes. Suitable for Resource allocation, Scheduling and monitoring of Spark application.

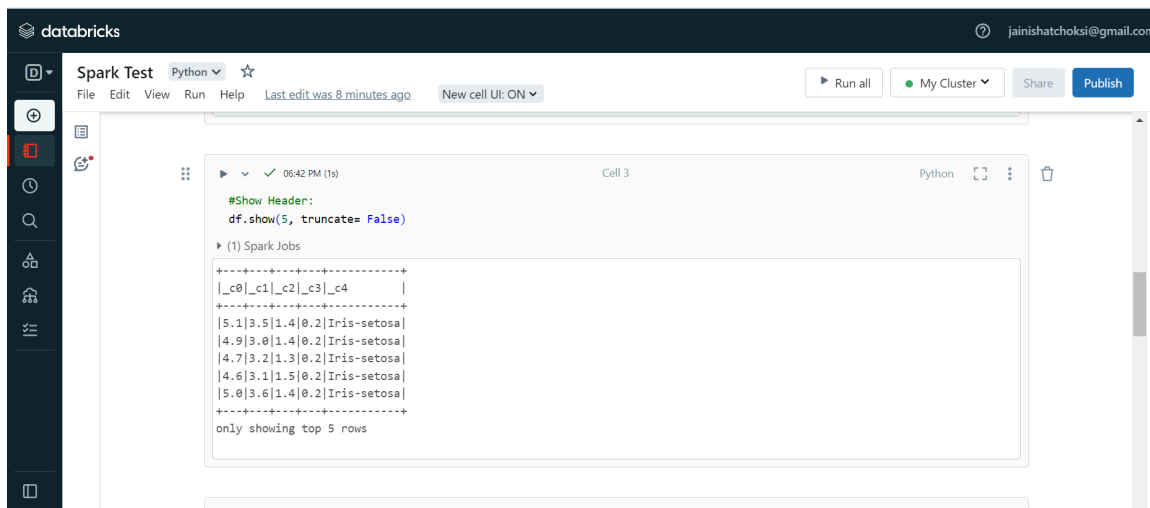
## 3. What is Cluster in Spark?

A Cluster in Spark refers to a group of computing resources that work together to process and analyze large datasets.

4. What is Table in Spark?

In Spark, a table is a structured collection of data organized into rows and columns. Tables can be created from various ways:  
Dataframes, RDD, external databases.

5. What would you do if you want to show the header while showing up 5 records of table?  
Write the code.



The screenshot shows the Databricks Spark Test interface. A Python cell (Cell 3) contains the following code:

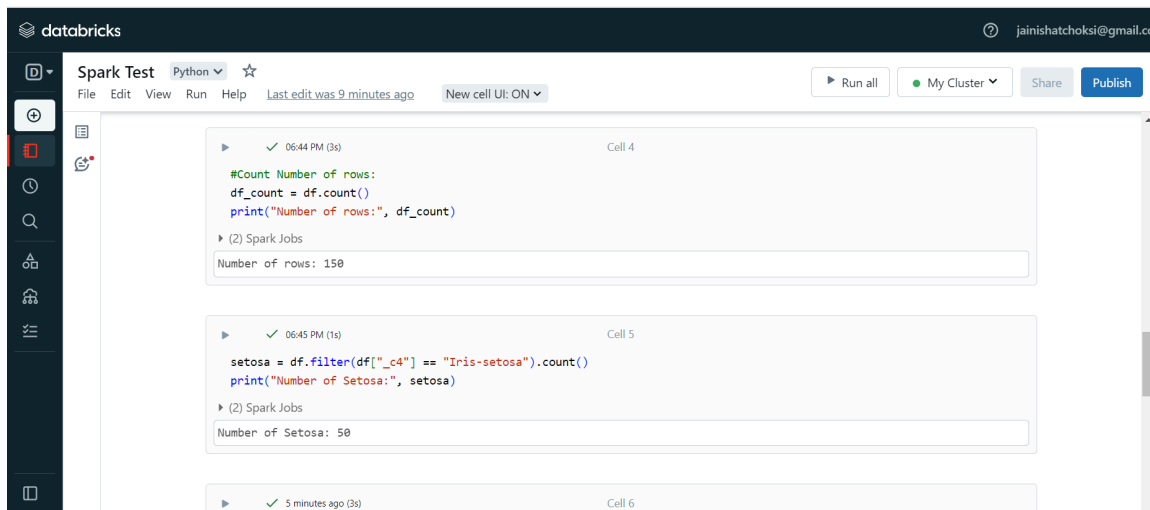
```
#Show Header:  
df.show(5, truncate= False)
```

The output of the code is displayed below the cell:

```
┌───┬───┬───┬───┬───┬───┐  
|_c0|_c1|_c2|_c3|_c4|  |  
├───┴───┴───┴───┴───┴───┤  
|5.1|3.5|1.4|0.2|Iris-setosa|  
|4.9|3.0|1.4|0.2|Iris-setosa|  
|4.7|3.2|1.3|0.2|Iris-setosa|  
|4.6|3.1|1.5|0.2|Iris-setosa|  
|5.0|3.6|1.4|0.2|Iris-setosa|  
├───┴───┴───┴───┴───┴───┤  
only showing top 5 rows
```

6. What is Count? Perform in Spark.

In Spark, count() is an action that returns the number of rows in a Dataframe.



The screenshot shows the Databricks Spark Test interface with two Python cells. Cell 4 contains the following code:

```
#Count Number of rows:  
df_count = df.count()  
print("Number of rows:", df_count)
```

The output of Cell 4 is:

```
Number of rows: 150
```

Cell 5 contains the following code:

```
setosa = df.filter(df["_c4"] == "Iris-setosa").count()  
print("Number of Setosa:", setosa)
```

The output of Cell 5 is:

```
Number of Setosa: 50
```

7. What is GroupBy? Perform in Spark.

In Spark, groupBy() is a transformation that groups the Dataframe rows based on the specified columns.

Databricks

jainishatchoksi@gmail.com

Spark Test

Python

☆

File Edit View Run Help

Last edit was 11 minutes ago

New cell UI: ON

Run all

My Cluster

Share

Publish

Cell 6

✓ 06:47 PM (3s)

#Group by Columns:  
group\_df = df.groupby("\_c4").count()  
group\_df.show()

(2) Spark Jobs

group\_df: pyspark.sql.dataframe.DataFrame = [\_c4: string, count: long]

+-----+  
| \_c4 | count |  
+-----+  
Iris-virginica	50
Iris-setosa	50
Iris-versicolor	50
+-----+

Cell 7

Python