Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

Artificial Intelligence and Data Science Department
Big Data Analytics/Odd Sem 2023-23/Experiment 5

Name: Shreya Singh	Class/Roll No.: D16AD/55	Grade:

Title of Experiment: Create HIVE Database and Descriptive analytics-based statistics, visualization using Hive/PIG.

Objective of Experiment:

This project aims to create an HIVE database and perform descriptive analytics-based statistics and visualization using Hive and PIG. This involves setting up a data storage and processing environment using Hadoop and Hive, analyzing the data to extract meaningful insights, and creating visualizations to present these insights effectively.

Outcome of Experiment: Thus we created a Hive Database and performed descriptive, Analytics-based statistics and visualization on the forestfire dataset using HIVE.

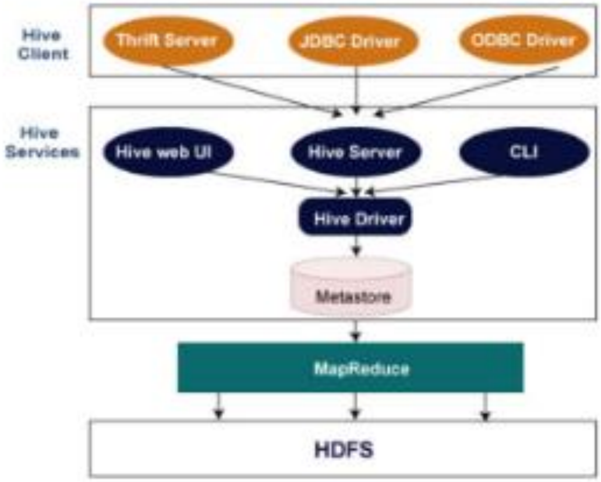
Problem Statement:
Establish a robust data storage and processing environment utilizing Hadoop and Hive, apply statistical analysis techniques to gain valuable insights from a forest fire dataset, and effectively visualize these insights for enhanced decision-making and understanding of forest fire patterns.

Description / Theory:

Hive is a data warehousing and query tool that simplifies the process of working with large datasets stored in a Hadoop cluster, especially for people who may not be skilled programmers or database experts. It's a part of the Hadoop ecosystem and was originally developed by Facebook.

Hive is like a bridge between big data stored in Hadoop and the world of SQL and data analysis. It allows you to work with massive datasets using a familiar SQL-like language without having to write complex code for data processing. It's a valuable tool in the big data ecosystem, especially for those who want to analyze and extract insights from vast amounts of information.

Hive Architecture:



The main components of Hive architecture:

- **User Interface (UI) / Hive CLI:** The Hive Command-Line Interface provides a way for users to interact with Hive by submitting SQL-like queries and managing Hive operations.

- **Hive Metastore:** The Metastore stores metadata about tables, partitions, schemas, and other information related to data stored in Hive. It serves as a centralized repository for managing metadata.

- **Execution Engine: (Hive Driver)**

  MapReduce: Hive can use the Hadoop MapReduce framework as an execution engine to process queries and transform them into MapReduce jobs.

  Tez: Alternatively, Hive can utilize Apache Tez as an optimized execution engine for faster query processing.

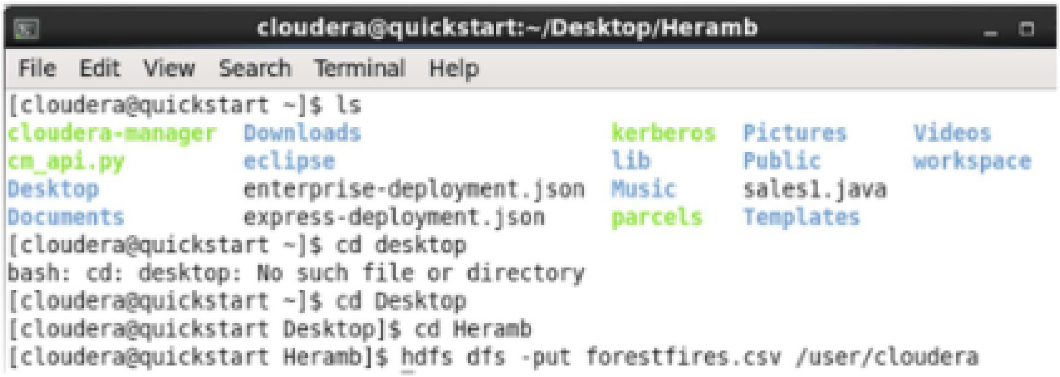
- **Storage Handler:**
  Storage Handler: Storage handlers define how data is stored, retrieved, and processed from various storage formats and systems, enabling Hive to integrate with different storage systems like HBase, ORC, Parquet, etc.

- **SerDe (Serializer/Deserializer):**
  SerDe: Serializer/Deserializer libraries define how data is serialized (stored) and deserialized (retrieved) in Hive, allowing it to work with various data formats, including JSON, CSV, and custom binary formats.

Program & Output:

Download Dataset from : **https://archive.ics.uci.edu/ml/datasets/forest+fires**

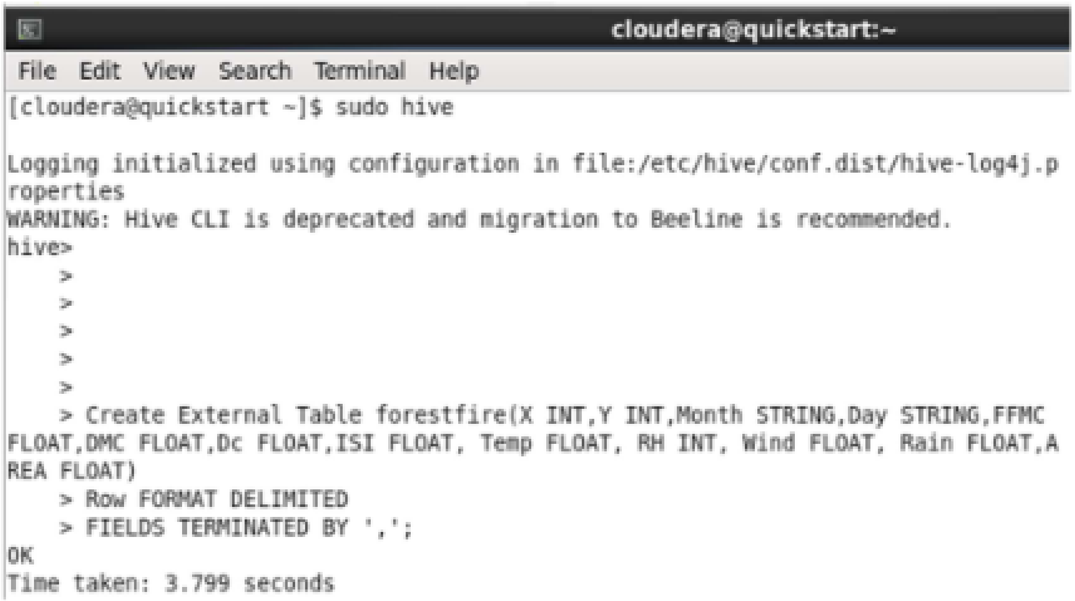
Upload Dataset into Cloudera.



cloudera@quickstart:~/Desktop/Heramb
File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ ls
cloudera-manager  Downloads                   kerberos  Pictures    Videos
cm_api.py         eclipse                     lib       Public      workspace
Desktop           enterprise-deployment.json  Music     sales1.java
Documents         express-deployment.json     parcels   Templates
[cloudera@quickstart ~]$ cd desktop
bash: cd: desktop: No such file or directory
[cloudera@quickstart ~]$ cd Desktop
[cloudera@quickstart Desktop]$ cd Heramb
[cloudera@quickstart Heramb]$ hdfs dfs -put forestfires.csv /user/cloudera
```

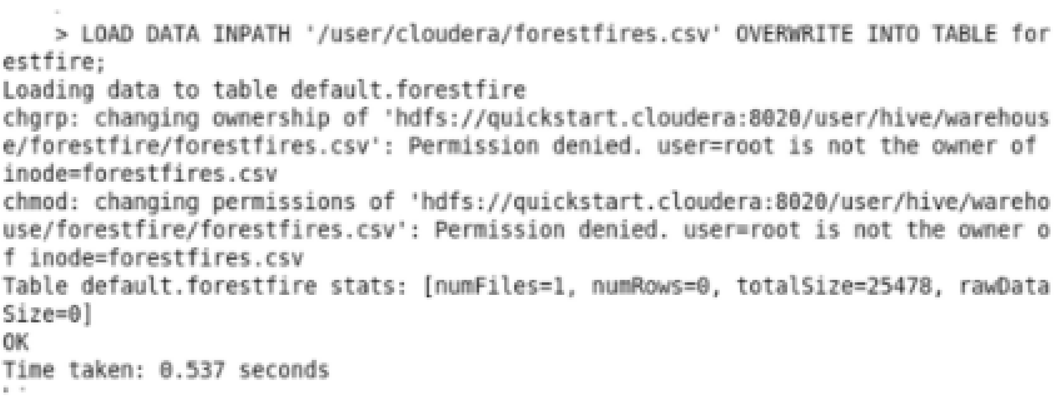
Opening Hive Shell & Creating ForestFire Table:



```
[cloudera@quickstart ~]$ sudo hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
    >
    >
    >
    >
    >
    > Create External Table forestfire(X INT,Y INT,Month STRING,Day STRING,FFMC
FLOAT,DMC FLOAT,Dc FLOAT,ISI FLOAT, Temp FLOAT, RH INT, Wind FLOAT, Rain FLOAT,A
REA FLOAT)
    > Row FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 3.799 seconds
```

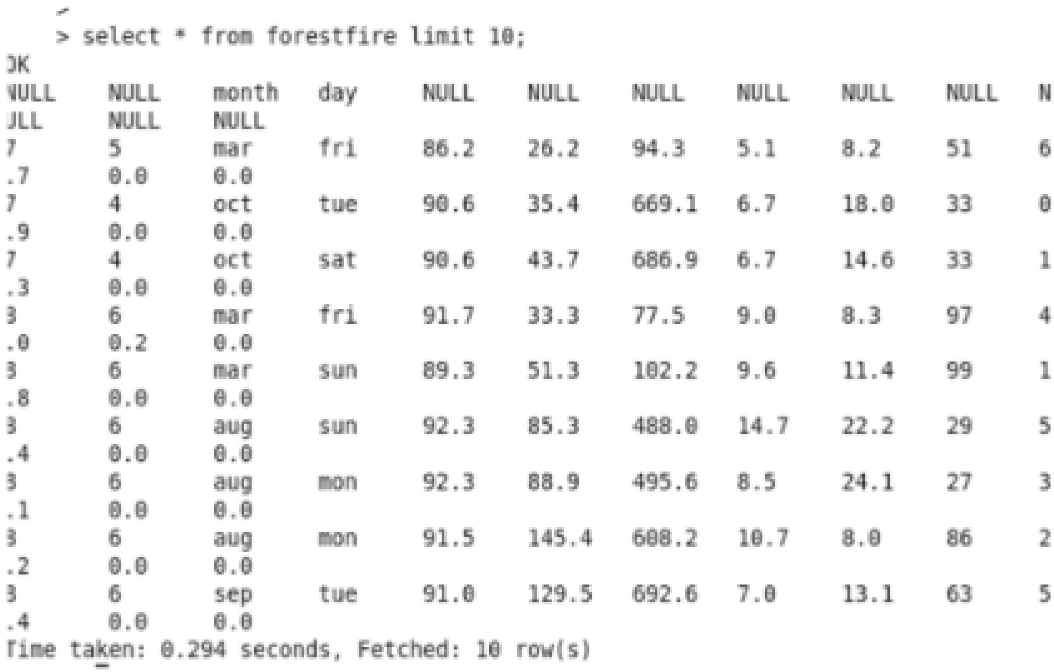
Loading Data From Dataset Into ForestFire Table:



```
    > LOAD DATA INPATH '/user/cloudera/forestfires.csv' OVERWRITE INTO TABLE for
estfire;
Loading data to table default.forestfire
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehous
e/forestfire/forestfires.csv': Permission denied. user=root is not the owner of
inode=forestfires.csv
chmod: changing permissions of 'hdfs://quickstart.cloudera:8020/user/hive/wareho
use/forestfire/forestfires.csv': Permission denied. user=root is not the owner o
f inode=forestfires.csv
Table default.forestfire stats: [numFiles=1, numRows=0, totalSize=25478, rawData
Size=0]
OK
Time taken: 0.537 seconds
```

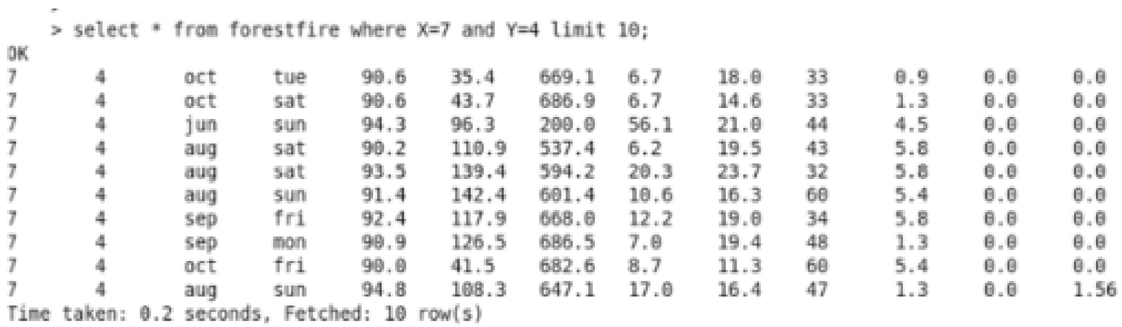
Executing Queries:

Query 1 : select * from forestfire limit 10;



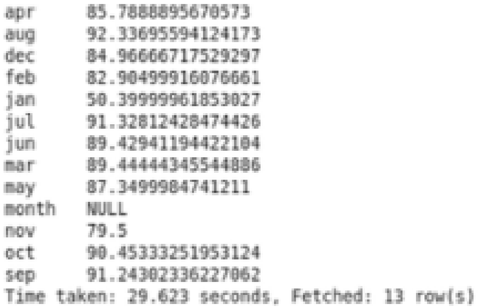
```
    > select * from forestfire limit 10;
)K
NULL    NULL    month   day     NULL    NULL    NULL    NULL    NULL    NULL    N
JLL     NULL    NULL
7       5       mar     fri     86.2    26.2    94.3    5.1     8.2     51      6
.7      0.0     0.0
7       4       oct     tue     90.6    35.4    669.1   6.7     18.0    33      0
.9      0.0     0.0
7       4       oct     sat     90.6    43.7    686.9   6.7     14.6    33      1
.3      0.0     0.0
3       6       mar     fri     91.7    33.3    77.5    9.0     8.3     97      4
.0      0.2     0.0
3       6       mar     sun     89.3    51.3    102.2   9.6     11.4    99      1
.8      0.0     0.0
3       6       aug     sun     92.3    85.3    488.0   14.7    22.2    29      5
.4      0.0     0.0
3       6       aug     mon     92.3    88.9    495.6   8.5     24.1    27      3
.1      0.0     0.0
3       6       aug     mon     91.5    145.4   608.2   10.7    8.0     86      2
.2      0.0     0.0
3       6       sep     tue     91.0    129.5   692.6   7.0     13.1    63      5
.4      0.0     0.0
Time taken: 0.294 seconds, Fetched: 10 row(s)
```

Query 2: : select * from forestfire where x=7 and y=4 limit 10;

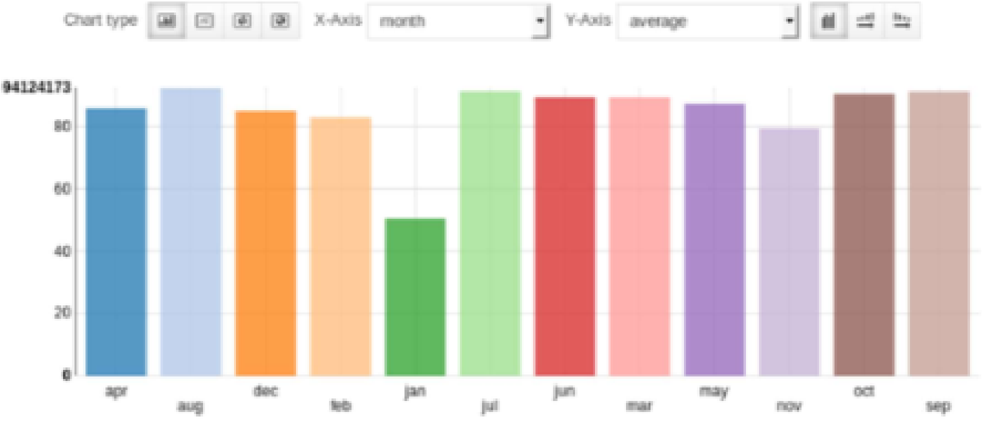


```
    -
    > select * from forestfire where X=7 and Y=4 limit 10;
OK
7       4       oct     tue     90.6    35.4    669.1   6.7     18.0    33      0.9     0.0     0.0
7       4       oct     sat     90.6    43.7    686.9   6.7     14.6    33      1.3     0.0     0.0
7       4       jun     sun     94.3    96.3    200.0   56.1    21.0    44      4.5     0.0     0.0
7       4       aug     sat     90.2    110.9   537.4   6.2     19.5    43      5.8     0.0     0.0
7       4       aug     sat     93.5    139.4   594.2   20.3    23.7    32      5.8     0.0     0.0
7       4       aug     sun     91.4    142.4   601.4   10.6    16.3    60      5.4     0.0     0.0
7       4       sep     fri     92.4    117.9   668.0   12.2    19.0    34      5.8     0.0     0.0
7       4       sep     mon     90.9    126.5   686.5   7.0     19.4    48      1.3     0.0     0.0
7       4       oct     fri     90.0    41.5    682.6   8.7     11.3    60      5.4     0.0     0.0
7       4       aug     sun     94.8    108.3   647.1   17.0    16.4    47      1.3     0.0     1.56
Time taken: 0.2 seconds, Fetched: 10 row(s)
```

Query 3: select MONTH, avg(FFMC) as Average from forestfire group by MONTH;

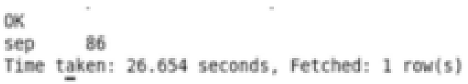


```
apr     85.7888895670573
aug     92.33695594124173
dec     84.96666717529297
feb     82.90499916076661
jan     50.39999961853027
jul     91.32812428474426
jun     89.42941194422104
mar     89.44444345544886
may     87.3499984741211
month   NULL
nov     79.5
oct     90.45333251953124
sep     91.24302336227062
Time taken: 29.623 seconds, Fetched: 13 row(s)
```



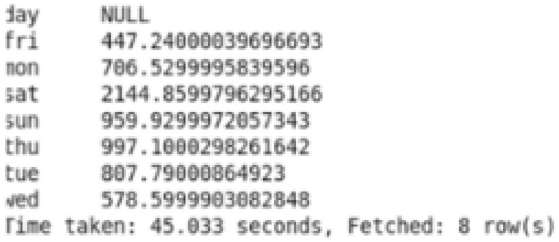
X-Axis month
Y-Axis average

Query 4: SELECT MONTH , MAX(RH) AS MAXIMUM FROM forestfire GROUP BY MONTH HAVING MONTH ='sep';

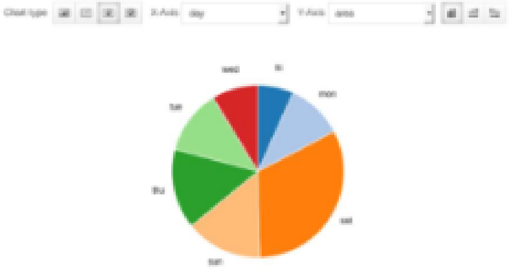


```
OK
sep     86
Time taken: 26.654 seconds, Fetched: 1 row(s)
```

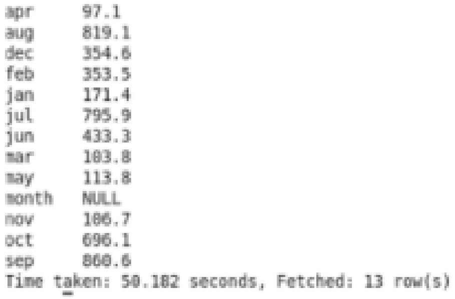
Query 5: select DAY, SUM(AREA) AS AREA from forestfire group by DAY ORDER BY DAY;



```
day      NULL
fri      447.24000039696693
mon      706.5299995839596
sat      2144.8599796295166
sun      959.9299972057343
thu      997.1000298261642
tue      807.79000864923
wed      578.5999903082848
Time taken: 45.033 seconds, Fetched: 8 row(s)
```



Query 6: SELECT MONTH, MAX(DC) AS MAXIMUM FROM forestfire GROUP BY MONTH ORDER BY MONTH;



```
apr     97.1
aug     819.1
dec     354.6
feb     353.5
jan     171.4
jul     795.9
jun     433.3
mar     183.8
may     113.8
month   NULL
nov     106.7
oct     696.1
sep     860.6
Time taken: 50.182 seconds, Fetched: 13 row(s)
```

Results and Discussions:

Resutls:

- We created a Hive database and loaded forest fire data.
- Explored the data with initial queries.
- Computed average FFMC by month.
- Identified maximum RH for September.
- Calculated total area burned by day.
- Determined maximum DC by month.

Discussion:

- Efficient data storage and initial data exploration are key.
- Average FFMC helps analyze moisture variations monthly.
- Maximum RH in September aids fire risk assessment.
- Total burned area by day reveals patterns.
- Maximum DC by month indicates drought risks.
- Location-based queries provide specific incident details.

This demonstrates Hive and Hadoop's utility for forest fire data analysis, aiding fire management decisions.