

NAME: Vedashree Bhalerao

CLASS: AIA-1

ROLL NO:2213191

BATCH: B

Big Data Analytics Experiment no. 04

Aim: 1) PySpark - Read CSV file into Data Frame
2) Create and query a HIVE table in PySpark.

Theory:

PySpark Overview:

PySpark is the Python API for Apache Spark, an open-source, distributed computing system. It provides a programming interface for entire clusters with implicit data parallelism and fault tolerance. PySpark allows you to write Spark applications using Python.

PySpark DataFrame:

A DataFrame is a distributed collection of data organized into named columns, similar to a table in a relational database or a data frame in R or Python (with pandas). It is one of the most commonly used abstractions in Spark.

Hive and Hive Tables:

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. It provides an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

Why Use PySpark with Hive?

- **Scalability:** Handle large datasets efficiently.
- **SQL Compatibility:** Use familiar SQL queries for data processing.
- **Integration:** Easily integrate with existing Hadoop ecosystems.

Program Details:

Prerequisites:

1. **Apache Spark:** Installed and configured.
2. **Hadoop and Hive:** Installed and configured if using a Hadoop-based deployment.
3. **Python:** Ensure Python 2.7 or 3.4 and later versions are installed.

Steps:

Step 1: PySpark - Read CSV File into DataFrame

1. **Initialize SparkSession:**

```
from pyspark.sql import SparkSession
spark = SparkSession.builder \
```

- ```

 .appName("ReadCSV") \
 .getOrCreate()
2. Read CSV File:

df = spark.read.csv("path/to/your/csvfile.csv", header=True, inferSchema=True)
3. Display DataFrame:

df.show()
4. Print Schema:

df.printSchema()

```

### *Step 2: Create and Query a Hive Table in PySpark*

- Enable Hive Support in SparkSession:**

```

spark = SparkSession.builder \
 .appName("HiveExample") \
 .enableHiveSupport() \
 .getOrCreate()

```
- Create Hive Database (if not exists):**

```

spark.sql("CREATE DATABASE IF NOT EXISTS mydb")
spark.sql("USE mydb")

```
- Create Hive Table:**

```

spark.sql("""
CREATE TABLE IF NOT EXISTS my_table (
 id INT,
 name STRING,
 age INT
) ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
""")

```
- Load Data into Hive Table:**

```

spark.sql("LOAD DATA LOCAL INPATH 'path/to/your/csvfile.csv' INTO TABLE
my_table")

```
- Query Hive Table:**

```

result = spark.sql("SELECT * FROM my_table")
result.show()

```

### **Conclusion:**

In this lab assignment, we have learned how to perform two essential tasks using PySpark:

- Reading a CSV File into a DataFrame:**
  - You initialized a SparkSession and read a CSV file into a DataFrame.
  - You displayed the contents and schema of the DataFrame.
- Creating and Querying a Hive Table:**
  - You enabled Hive support in SparkSession and created a Hive database and table.
  - You loaded data from a CSV file into the Hive table and performed SQL queries on the table.

These skills are fundamental for working with large datasets in a distributed computing environment, allowing you to efficiently process and analyze data using the power of Apache Spark and Hive. By integrating these technologies, you can leverage the scalability of Spark and the SQL capabilities of Hive, making your data processing tasks more effective and streamlined.

## Code :-

```
import pandas as pd

def read_csv_file(file_path):
 return pd.read_csv(file_path)

df = read_csv_file('C:/Users/Hp/Documents/ds_salaries.csv')

print(df.head())

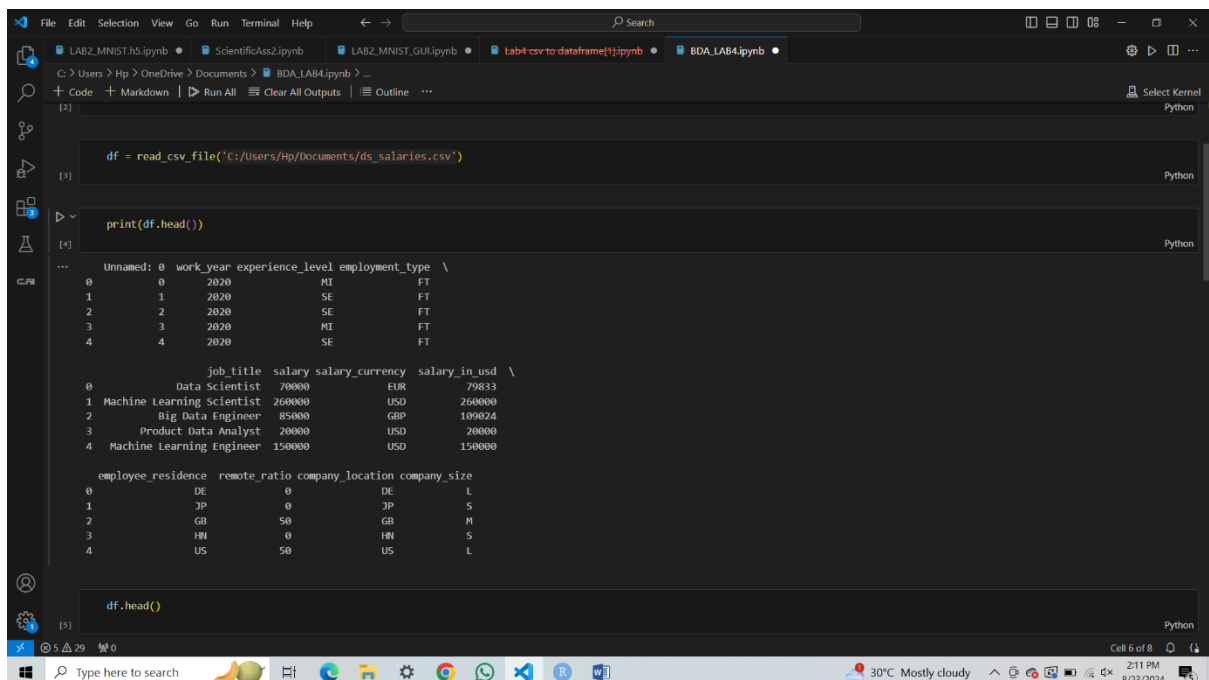
Example of reading a text file as input

df_txt = pd.read_csv('C:/Users/Admin/Desktop/crow.txt', delimiter=' ')

print(df_txt.head())

df_txt.head()
```

## OUTPUT:-



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
df = read_csv_file('C:/Users/Hp/Documents/ds_salaries.csv')
```

```
print(df.head())
```

The output of the second code cell is a table with 5 rows and 5 columns:

| Unnamed: 0 | work_year | experience_level | employment_type |    |
|------------|-----------|------------------|-----------------|----|
| 0          | 0         | 2020             | MI              | FT |
| 1          | 1         | 2020             | SE              | FT |
| 2          | 2         | 2020             | SE              | FT |
| 3          | 3         | 2020             | MI              | FT |
| 4          | 4         | 2020             | SE              | FT |

The output of the third code cell is a table with 5 rows and 5 columns:

|   | job_title                  | salary | salary_currency | salary_in_usd |
|---|----------------------------|--------|-----------------|---------------|
| 0 | Data Scientist             | 70000  | EUR             | 79833         |
| 1 | Machine Learning Scientist | 260000 | USD             | 260000        |
| 2 | Big Data Engineer          | 85000  | GBP             | 109024        |
| 3 | Product Data Analyst       | 20000  | USD             | 20000         |
| 4 | Machine Learning Engineer  | 150000 | USD             | 150000        |

The output of the fourth code cell is a table with 5 rows and 5 columns:

| employee_residence | remote_ratio | company_location | company_size |   |
|--------------------|--------------|------------------|--------------|---|
| 0                  | DE           | 0                | DE           | L |
| 1                  | JP           | 0                | JP           | S |
| 2                  | GB           | 50               | GB           | M |
| 3                  | HN           | 0                | HN           | S |
| 4                  | US           | 50               | US           | L |

The output of the fifth code cell is a table with 5 rows and 5 columns:

|   | job_title                  | salary | salary_currency | salary_in_usd |
|---|----------------------------|--------|-----------------|---------------|
| 0 | Data Scientist             | 70000  | EUR             | 79833         |
| 1 | Machine Learning Scientist | 260000 | USD             | 260000        |
| 2 | Big Data Engineer          | 85000  | GBP             | 109024        |
| 3 | Product Data Analyst       | 20000  | USD             | 20000         |
| 4 | Machine Learning Engineer  | 150000 | USD             | 150000        |

File Edit Selection View Go Run Terminal Help

LAB2\_MNIST.ipynb ScientificAss2.ipynb LAB2\_MNIST\_GUI.ipynb tab4.csv to dataframe.ipynb BDA\_LAB4.ipynb

C:\Users\Hp> OneDrive> Documents> BDA\_LAB4.ipynb > ...

+ Code + Markdown | Run All | Clear All Outputs | Outline ...

Select Kernel

|   | Unnamed: 0 | work_year | experience_level | employment_type | job_title                  | salary | salary_currency | salary_in_usd | employee_residence | remote_ratio | company_location | company_size |
|---|------------|-----------|------------------|-----------------|----------------------------|--------|-----------------|---------------|--------------------|--------------|------------------|--------------|
| 0 | 0          | 2020      | MI               | FT              | Data Scientist             | 70000  | EUR             | 79833         | DE                 | 0            | DE               | L            |
| 1 | 1          | 2020      | SE               | FT              | Machine Learning Scientist | 260000 | USD             | 260000        | JP                 | 0            | JP               | S            |
| 2 | 2          | 2020      | SE               | FT              | Big Data Engineer          | 85000  | GBP             | 109024        | GB                 | 50           | GB               | M            |
| 3 | 3          | 2020      | MI               | FT              | Product Data Analyst       | 20000  | USD             | 20000         | HN                 | 0            | HN               | S            |
| 4 | 4          | 2020      | SE               | FT              | Machine Learning Engineer  | 150000 | USD             | 150000        | US                 | 50           | US               | L            |

```
df_txt = pd.read_csv('C:/Users/Hp/Documents/crow.txt', delimiter=',')

print(df_txt.head())
```

Python

Python

On a very hot summer day in the fields \

0 Then an idea came to him. He started picking t...

1 Moral Of The Story

2 "Where there is a will

a thirsty crow flew everywhere in search of water. Suddenly \

0 the water rose. The crow drank the water and ...

1 NaN

2 there is a way!"

his gaze landed on a pitcher on the ground with little water in it. The water level was too low \

0 NaN

1 NaN

2 NaN

Spaces: 4 Cell 6 of 8 2:11 PM 8/23/2024

File Edit Selection View Go Run Terminal Help

LAB2\_MNIST.ipynb ScientificAss2.ipynb LAB2\_MNIST\_GUI.ipynb tab4.csv to dataframe.ipynb BDA\_LAB4.ipynb

C:\Users\Hp> OneDrive> Documents> BDA\_LAB4.ipynb > ...

+ Code + Markdown | Run All | Clear All Outputs | Outline ...

Select Kernel

his gaze landed on a pitcher on the ground with little water in it. The water level was too low \

0 NaN

1 NaN

2 NaN

and the pitcher had a narrow neck \

0 NaN

1 NaN

2 NaN

because of which the crow could not drink the water.

0 NaN

1 NaN

2 NaN

```
df_txt.head()
```

Python

|   | On a very hot summer day in the fields            | a thirsty crow flew everywhere in search of water. Suddenly | his gaze landed on a pitcher on the ground with little water in it. The water level was too low | and the pitcher had a narrow neck | because of which the crow could not drink the water. |
|---|---------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------|
| 0 | Then an idea came to him. He started picking t... | the water rose. The crow drank the water and ...            | NaN                                                                                             | NaN                               | NaN                                                  |
| 1 | Moral Of The Story                                | NaN                                                         | NaN                                                                                             | NaN                               | NaN                                                  |
| 2 | "Where there is a will                            | there is a way!"                                            | NaN                                                                                             | NaN                               | NaN                                                  |

Spaces: 4 Cell 6 of 8 2:11 PM 8/23/2024