Read Text file into PySpark Dataframe.

=> Some file formats other than csv and textfile are:- parquet, orc, avro etc.

=> spark.read.format("text").load("output.txt") OR spark.read.csv or spark.read.text

Process of execution:

-> It converts textfile into dataframe.
when running on the pyspark notebook, it stores as RDD.

eg:-

```
from pyspark.sql import SparkSession.
spark = SparkSession.builder.getOrCreate()
df = spark.read.format("text").load("output.txt")
```

=> Using selectExpr and split method.

```
df = selectExpr("split(Value, ',')
```

or.

```
df.selectExpr as Text_Data_In_Rows_Using_format_load").
                                        show(4, false)
```

=> For eg:- calculating avg of age:-

```
df.selectExpr("AVG(age) as avg_age").show()
```

Method to add new Column with constant Value.

→ Using LIT

dataframe.withColumn("column name", lit(value)).show

LIT → function to download column.

→ Using Concat

dataframe.withColumn("column name", concat_ws("-", "Name", "Company")).show

→ Add Column when not exists on DF.

Syntax:-

```
if 'column name' not in dataframe.columns:
    dataframe.withColumn("column-name", lit(value))
```

Manipulating Dataframes:

→ GroupBy and Aggregate Functions:

Group by is used to collect identical data

① Create a dataframe

② Syntax for groupBy():-

df_pyspark.groupBy("column name").sum("value").show()

df_pyspark.groupBy("Department").sum("Salary").show()

Using Pivot / Unpivot

Used to rotate data from one column into multiple columns. (Transpose row to column).

Syntax :- df_pyspark.groupBy("Departments").pivot("Name").sum("Salary").show().

Handling missing values

dropping rows based on null values

df_pyspark.na.drop().show()

Parameters of drop :-

df_pyspark.na.drop(how="all").show().

df_pyspark.na.drop(how="any", thresh=2).show()

df_pyspark.na.drop(how="any", subset=["salary"]).show()

Subset = optional list of col. names to consider. Columns specified in subset that do not have matching data type are ignored.

for eg :- if value is a string, non-string value will be ignored.

Order By and Sort.

(1) sort() → default is asc order

Syntax = df_pyspark.sort("salary").show()
= df_pyspark.sort(df_pyspark["salary"].desc()).show()
= df_pyspark.sort("salary", "name").show()

(2) OrderBy()
Syntax → df_pyspark.orderBy("salary").show()

Joins

1. Inner Join
2. Outer Join
3. Left Join / Left Outer Join
4. Right Join / Right Outer Join
5. Left Semi Join
6. Left Anti Join.

- Step 1:- Create two diff. dataframes.