Contents:-

1) PySpark Dataframe
2) Reading the Dataset.
3) Checking the datatypes of the Column: Schema
4) Selecting columns and Indexing
5) Check Describe option similar to Pandas
6) Adding & Dropping columns.

We learnt functions to read a file, and describe the datatypes of column.

=> While reading data from file (using read() method), all the columns in the Pyspark dataframe have string datatype by default, irrespective of being integer of boolean or date, in the actual data.

Taking this table as example here, for reading

test1.csv

Name	age	Experience
Krish	31	10
Sudhansu	30	8
Sunny	29	4

To make PySpark interpret datatype of each column correctly, according to values. we need to pass another parameter to the csv() [or related] method as follows:-

```
df_spark=spark.read.....csv('test1.csv', inferSchema=True)
```

(inferSchema: capital)

this is the parameter needed to interpret datatypes

- Then, if we display the schema, we can see that all columns have appropriate datatypes.

- We can collab both header being True, and inferSchema being True, as follows :-

```
df_spark=...csv('test1.csv', header=True, inferSchema=True)
```

header=True: by passing this, we say that 1st row will be header

inferSchema=True: Datatype interpretation acc. to values

22:05

Getting the Column names :-

```
df_spark.columns
```

- Contains (showed in notebook) a list of strings, where each string represent column names.

Selecting single/specific column(s):-

- select() method takes the column name(s) (in quotes) as parameter, and returns that entire column as a PySpark dataframe.

```
df_spark.select('Name')
```

⇒ By displaying the returned dataframe by above code, we can see like this :-

Name
Krish
Sudhanshu
Sunny

⇒ For selecting multiple columns, we pass the column names in a Python list :-

```
df_spark.select(['Name', 'Experience'])
```

It will return the dataframe with these two columns in it.

⇒ Thus, basically, by using select() command we get a sub-dataframe from the main dataframe

---

⇒ Checking Datatypes of Columns :-

⇒ Another way of checking datatypes of each column of the dataframe is by dtypes() :-

```
df_spark.dtypes
[('Name', 'string'), ('age', 'int'), ('Experience', 'int')]
```

⇒ Returns a list of tuples, in which each tuple contains two elements.

1st element is the column name as string.

2nd element is the column datatype as string.

Description of DataFrame :-

```
df_spark.describe()
```

Returns / displays the description of dataframe.

Also, we can check description of each column in terms of count, mean, min, max, standard deviation, by this :-

```
df_spark.describe().show()
```

Adding Columns :-

withColumn() method is used to return a spark dataframe, by adding a column, or replacing the column with the same name

Takes 2 parameters :-

1st parameter is the new column name to be added.

2nd param : The value in the new column.

May be, we can perform some operation to get set of values for the new column.

Ex:- the method → `df_pyspark.withColumn("Experience after 2 ... years", df_pyspark['Experience']+2).show()` ← new column name

adding 2 to each value of 'Experience' column.

Name	Age	Experience	Experience after 2 yrs
Krish	31	10 (+2)	12
Sudhansu	30	8 (+2)	10
Sunny	29	4 (+2)	6

⇒ We store this dataframe to new variable for further processing.

⇒ Dropping Columns :-

⇒ drop() method takes single column name as string, or list of column names, and drop the column(s), and returns the modified dataframe.

⇒ This method does not affect the original dataframe — , just returns modified dataframe.

⇒ If column name specified does not exist, then no operation takes place.

⇒ `df_spark.drop('Experience')`

⇒ (or) `df_spark.drop(['age', 'Experience'])`

30:40

Renaming Column :-

- withColumnRenamed() method is used to rename any column name, and returns the modified Dataframe.
- It takes 2 parameter :-
  i) 1st param: Existing column name
  ii) 2nd param: New column name

```
df_spark.withColumnRenamed('Name', 'Full Name')
```

'Name' → existing column name

'Full Name' → new name set for the column