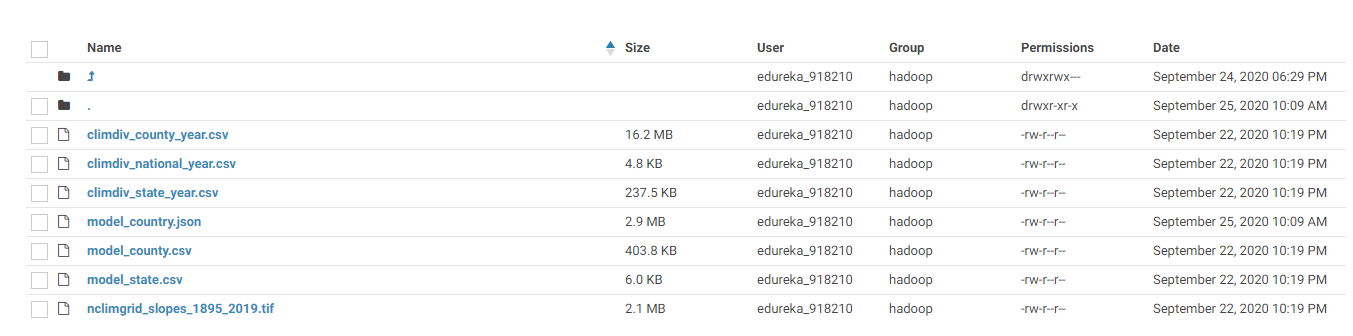
**Weather Data Analysis – in Apache Spark**

I tried to bring analysis on the Data collected on the Weather Details of various countries. These data’s are of utmost importance and help us to analyse and understand the global impact across various countries.

The sources from which the Data are downloaded are :

<https://data.world/>

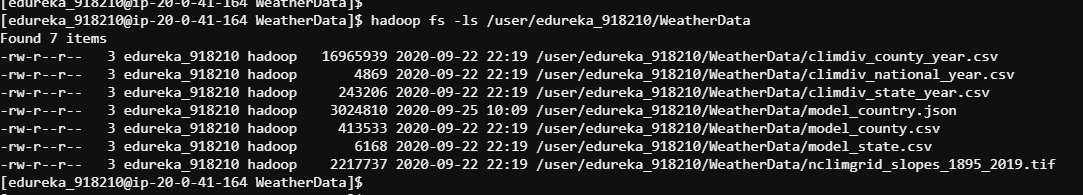
The raw files stored in the HDFS filesystem are :



The commands that is used to transfer the files from the local system to the HDFS are :

hadoop fs -ls /user/edureka\_918210/WeatherData

hadoop fs -put <filename1> <filename2> <filename3> <filename4> <filename5> <filename6> /user/edureka\_918210/WeatherData



**The commands that are used in spark are as follows :**

var myDF\_climdiv\_county\_year=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/user/edureka\_918210/WeatherData/climdiv\_county\_year.csv").toDF("fips","year","temp","tempc")

var myDF\_climdiv\_national\_year=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/user/edureka\_918210/WeatherData/climdiv\_national\_year.csv").toDF("year","temp","tempc")

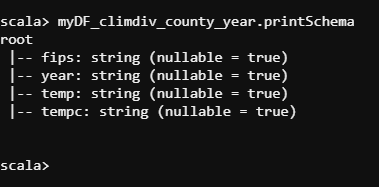
var myDF\_climdiv\_state\_year=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/user/edureka\_918210/WeatherData/climdiv\_state\_year.csv").toDF("fips","year","temp","tempc")

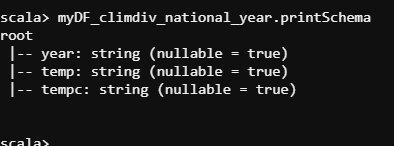
var myDF\_model\_country=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/user/edureka\_918210/WeatherData/model\_county.csv").toDF("fips","Fall","Spring","Summer","Winter","max\_warming\_season","Annual","CTYNAME","STNAME")

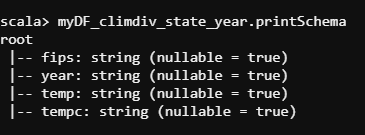
var myDF\_model\_country\_json=spark.read.format("json").option("delimiter",",").option("header","false").option("inferSchema","true").load("/user/edureka\_918210/WeatherData/model\_country.json").toDF

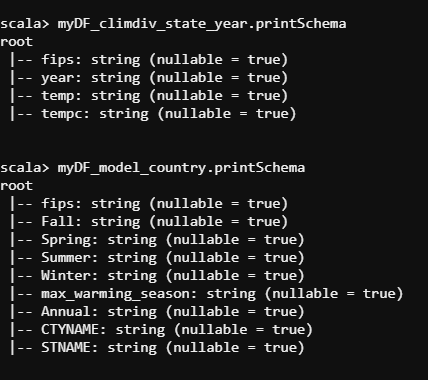
var myDF\_model\_state=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/user/edureka\_918210/WeatherData/model\_state.csv").toDF("fips","Fall","Spring","Summer","Winter","max\_warming\_season","Annual","STUSAB","STATE\_NAME","STATENS")

**Here's the schema details :**

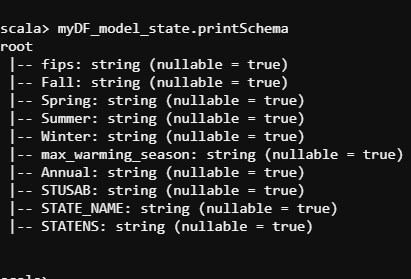












**The Result Details are shown below :**

myDF\_climdiv\_county\_year.show(false)

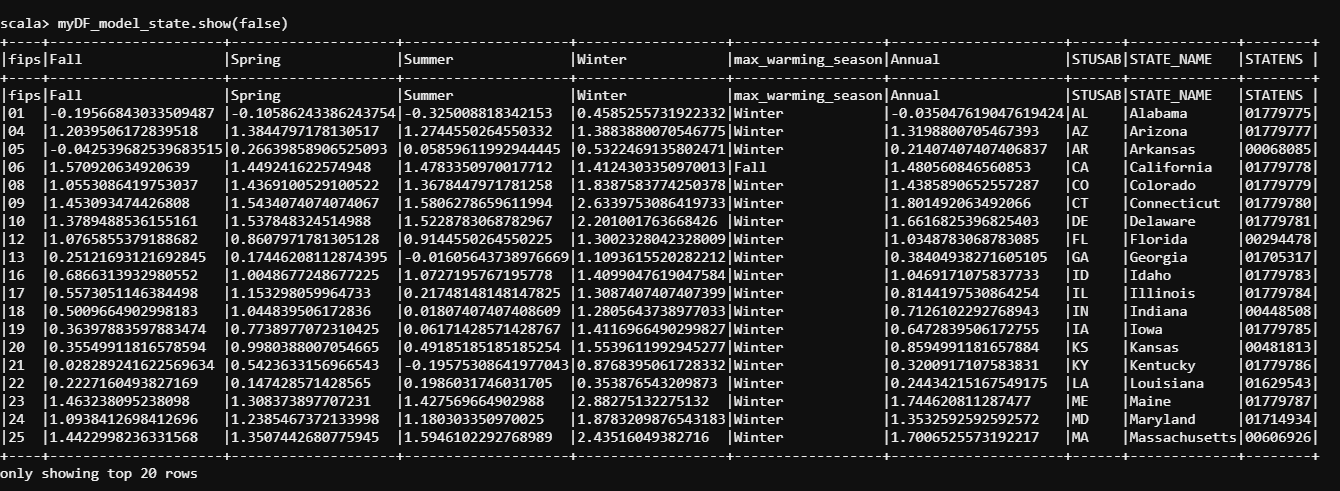
myDF\_climdiv\_national\_year.show(false)

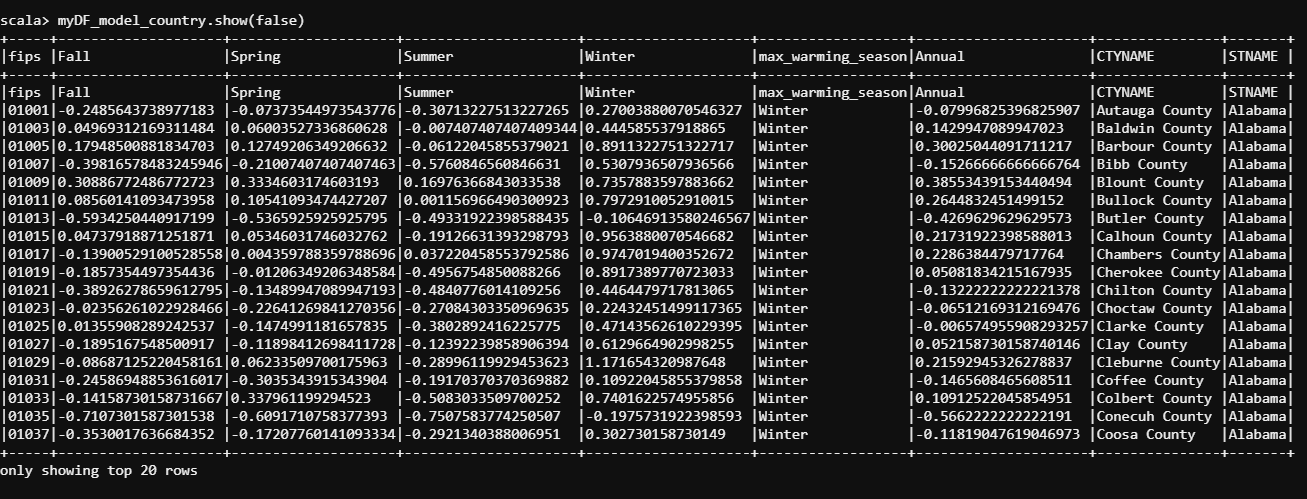
myDF\_climdiv\_state\_year.show(false)

myDF\_model\_country.show(false)

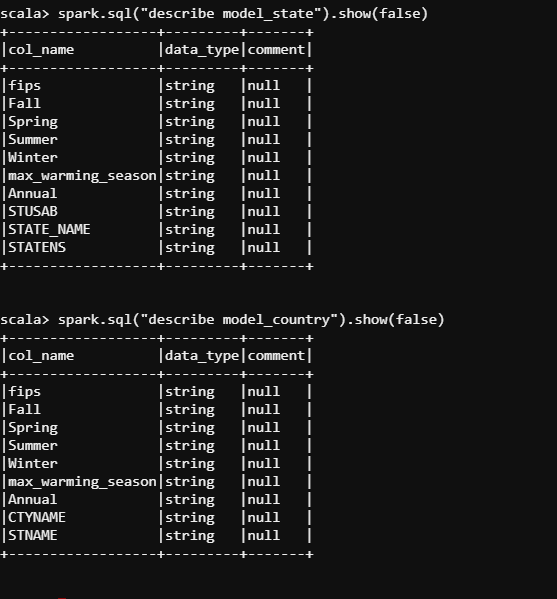
myDF\_model\_country\_json.show(false)

myDF\_model\_state.show(false)

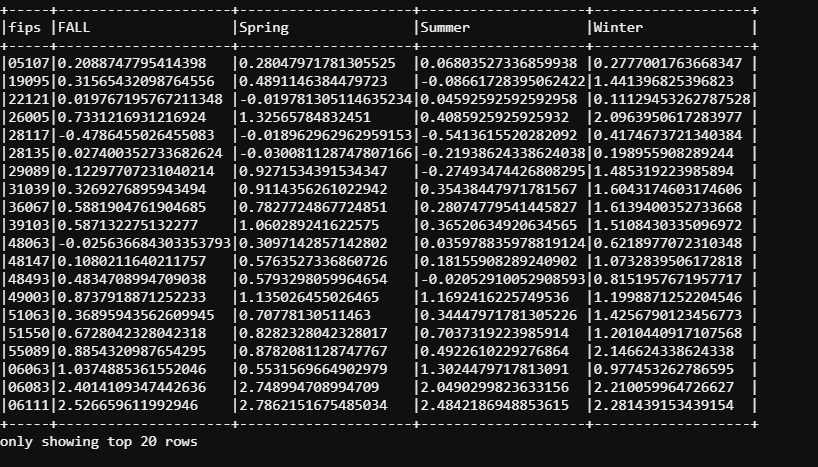




**SPARK SQL SUCCESSFUL CONVERSION**



spark.sql("SELECT model\_country.fips,model\_country.FALL,model\_country.Spring,model\_country.Summer,model\_country.Winter FROM model\_country FULL OUTER JOIN model\_state ON model\_country.fips = model\_state.fips").show(false)



**SAVE AS CSV FILES**

spark.sql("SELECT model\_country.FALL,model\_country.Spring,model\_country.Summer,model\_country.Winter FROM model\_country FULL OUTER JOIN model\_state ON model\_country.fips = model\_state.fips").write.save("/user/edureka\_918210/WeatherData/join\_by\_fips.csv")

