CASE STUDY 4

**Hospital data anslysis**

This case study have a movie dataset, which contains the following data file:

**Dataset Description:**

**DRG Definition:**

The code and description identifying the MS-DRG. MS-DRGs are a classification system that groups similar clinical conditions (diagnoses) and procedures furnished by the hospital during their stay.

**Provider Id:** The CMS Certification Number (CCN) assigned to the Medicare-certified hospital facility.

**Provider Name:** The name of the provider.

**Provider Street Address:** The provider’s street address.

**Provider City:** The city where the provider is located.

**Provider State:** The state where the provider is located.

**Provider Zip Code:** The provider’s zip code.

**Provider HRR:** The Hospital Referral Region (HRR) where the provider is located.

**Total Discharges:** The number of discharges billed by the provider for inpatient hospital services.

**Average Covered Charges:** The provider’s average charge for services covered by Medicare for all discharges in the MS-DRG. These will vary from hospital to hospital because of the differences in hospital charge structures.

**Average Total Payments:** The average total payments to all providers for the MS-DRG including the MSDRG amount, teaching, disproportionate share, capital, and outlier payments for all cases. Also included in the average total payments are co-payment and deductible amounts that the patient is responsible for and any additional payments by

third parties for coordination of benefits.

**Average Medicare Payments:** The average amount that Medicare pays to the provider for Medicare’s share of the MS-DRG. Average Medicare payment amounts include the MS-DRG amount, teaching, disproportionate share, capital, and outlier payments for all cases. Medicare payments DO NOT include beneficiary co-payments and deductible amounts nor any additional payments from third parties for coordination of benefits.

**First we download the dataset and store it /user/acadgild/Hadoop/hospital**

**OBJECTIVE 1:-**

**Load file into spark**

For loading the file in spark first we open the spark shell by writing **spark-shell** on terminal after running all the services related to Hadoop

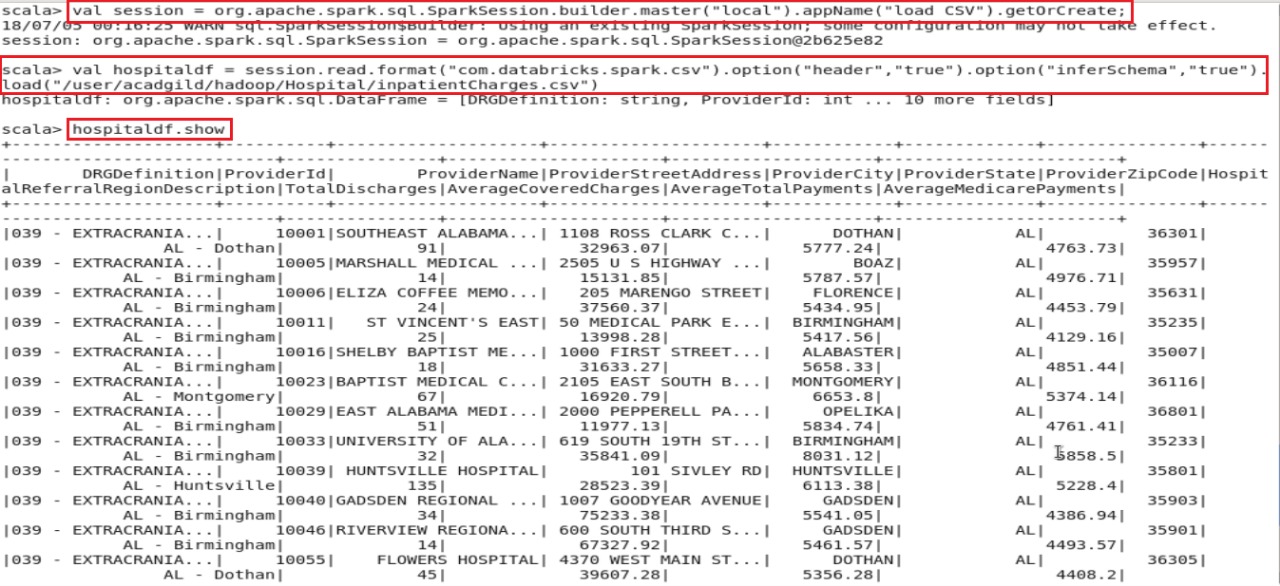
After that we will use different commands to load the file into spark

**1:- val session = org.apache.spark.sql.SparkSession.builder.master(“local”).appName(load csv).getOrCreate;**

This will create a session

**2:-val hospitaldf = session.read.format(“com.databricks.spark.csv”).option(“header”,”true”).option(“inferSchema”,”true”).load(“/user/acadgild/Hadoop/hospital/inpatientCharges.csv”)**

This will load the inpatientCharges.csv file which is present under /user/acadgild/Hadoop/hospital



**3 :- hospitalldf.show :-** this will show the data which is present in the file which is loaded in spark

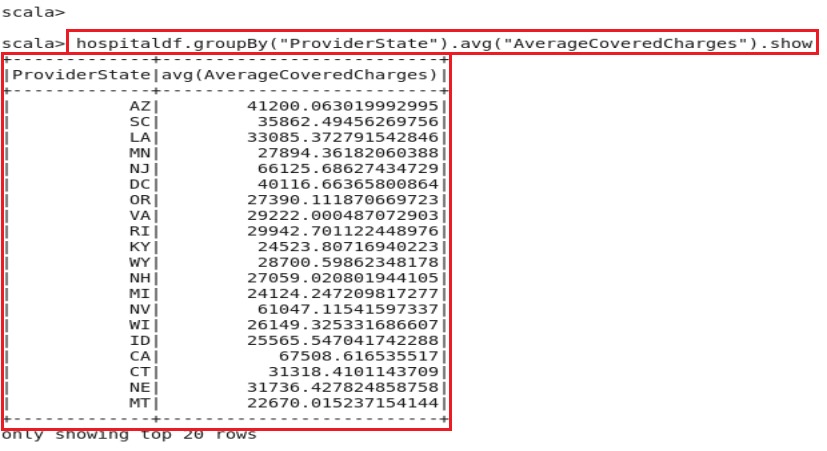
**OBJECTIVE 2:-**

**Task 1:- What is the average amount of AverageCoveredCharges per state.**

**Solution:-**

For finding the average amount of AverageCoveredCharges per state we will first use the group by function which will group thee data based on ProviderState and on that grouped data we will apply average function on AverageCoveredCharges to calculate the average covered charges and finally use show function to show the result.

**Command:- hospitaldf.groupBy(“ProviderState”).avg(“AverageCoveredCharges”).show**



This will give the desired result………

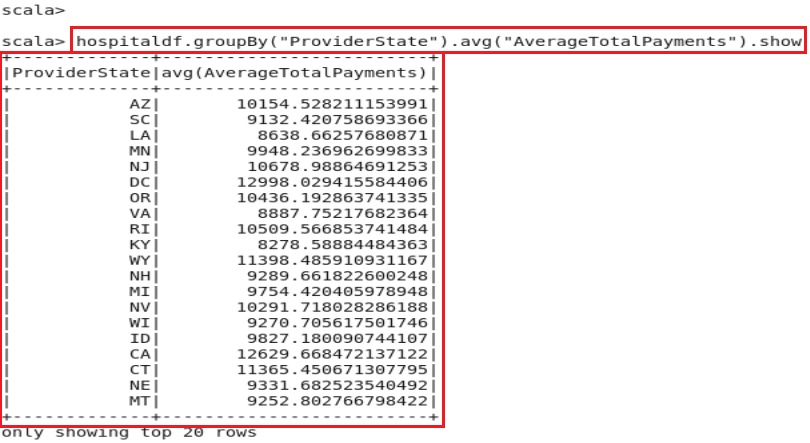
**OBJECTIVE 2:-**

**Task 2:- find out the AverageTotalPayments charges per state.**

**Solution:**

For finding the average of AverageTotalPayments per state we will first use the group by function which will group thee data based on ProviderState and on that grouped data we will apply average function on AverageTotalPayments to calculate the average

Total charges and finally use show function to show the result.

****

**Command:- hospitaldf.groupBy(“ProviderState”).avg(“AverageTotalPayments”).show**

This will give the desired result………

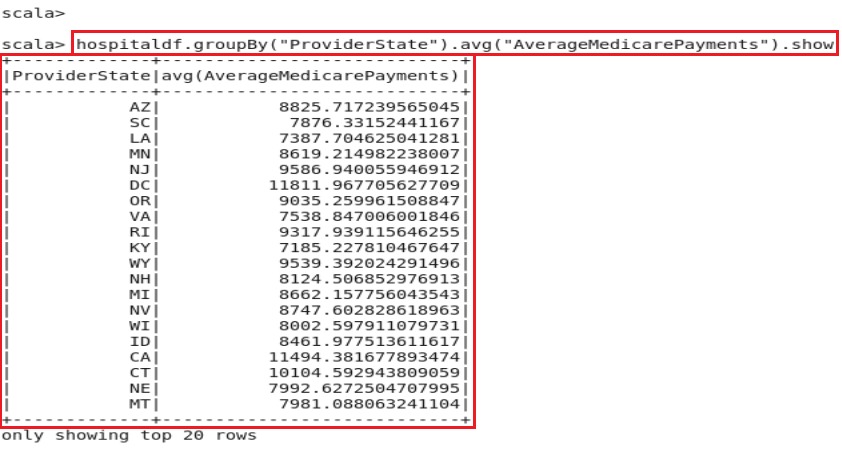
**OBJECTIVE 2:-**

**Task3:- find out the AverageMedicarePayments charges per state.**

**Solution:**

For finding the average of AverageMedicarePayments per state we will first use the group by function which will group thee data based on ProviderState and on that grouped data we will apply average function on AverageMedicarePayments to calculate the average

medicare payment and finally use show function to show the result.



**Command:- hospitaldf.groupBy(“ProviderState”).avg(“AverageMedicarePayments”).show**

This will give the desired result………

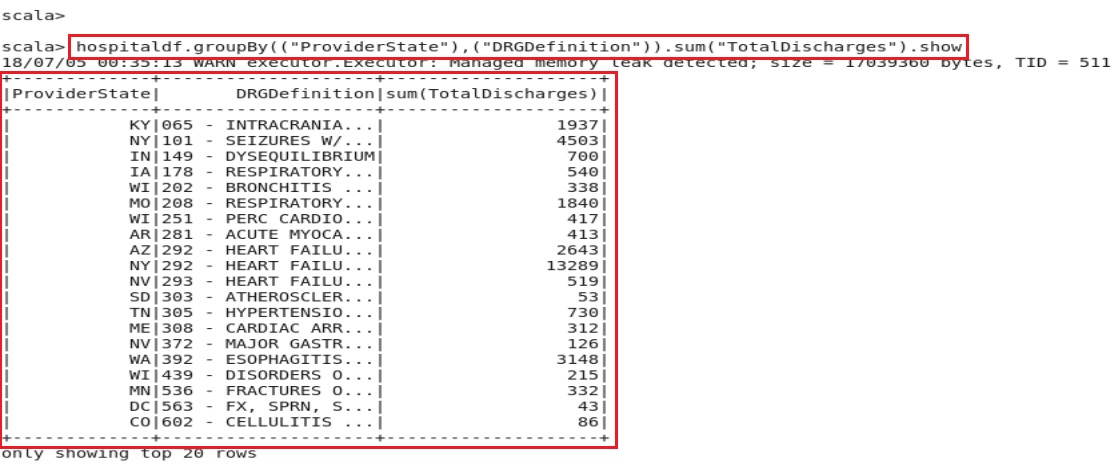
**OBJECTIVE 3:-**

**Task 1:- Find out the total number of Discharges per state and for each disease**

**Solution:**

For finding the total number of discharge per state and for each desease we will first use the group by function which will group thee data based on ProviderState and DRGDefination

on that grouped data we will apply sum function on Total Discharges to calculate the sum and finally use show function to show the result.



**Command:-hospitaldf.groupBy((“ProviderState”),(DRGDefination)).sum(“TotalDischarges”).show**

This command will give the respective result

**OBJECTIVE 3**

**Task 2:- Sort the output in descending order of totalDischarges**

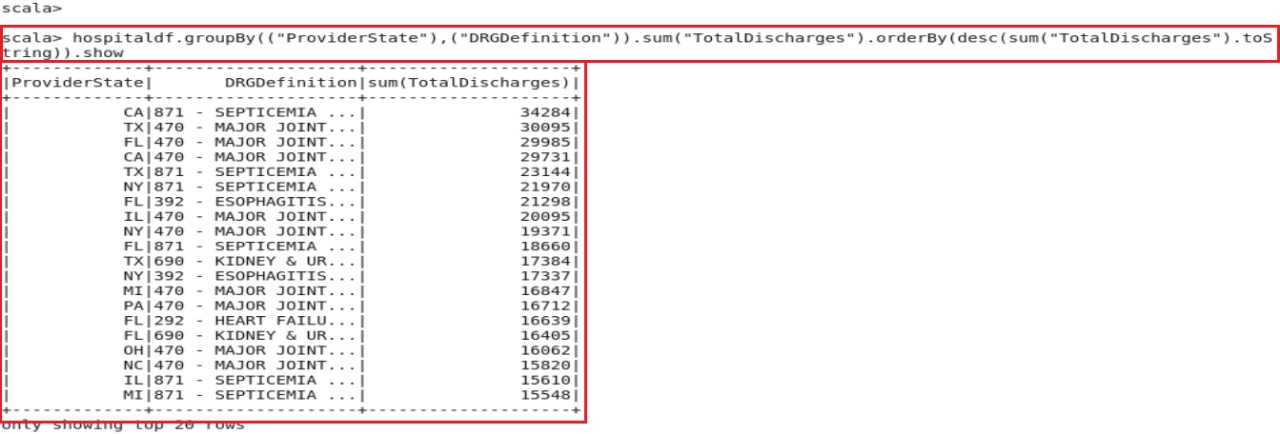
**Solution:**

**First we will repeat the step that we done before in task2 of objective 3**

For finding the total number of discharge per state and for each desease we will first use the group by function which will group thee data based on ProviderState and DRGDefination

on that grouped data we will apply sum function on Total Discharges to calculate the sum

and then we will use orderBy(desc(sum(“TotalDischarges”).toString)) this will order the sum of Total Discharges in descending order and finally use show function to show the result.

****

**Command:- hospitaldf.groupby((“ProviderState”),(“DRGDefination”)).sum(“TotalDischarges”).orderBy(desc(sum(“TotalDischarges”).toString)).show**

This will give the desired result………