Pyspark handson

Go to File ->Open and select challenge.csv and original.csv from your system in python notebook

**Step 1: Creating a SparkSession**

A SparkSession is an entry point into all functionality in Spark, and is required if you want to build a dataframe in PySpark. Run the following lines of code to initialize a SparkSession:

from pyspark.sql import SparkSession

spark=SparkSession.builder.appName("Datacamp").config("spark.memory.offHeap.enabled","true").config("spark.memory.offHeap.size","10g").getOrCreate()

Using the codes above, we built a spark session and set a name for the application. Then, the data was cached in off-heap memory to avoid storing it directly on disk, and the amount of memory was manually specified.

**Step 2: Creating the DataFrame**

We can now read the dataset we just downloaded:

df = spark.read.csv('challenge.csv',header=True,escape="\"")

Note that we defined an escape character to avoid commas in the .csv file when parsing.

Let’s take a look at the head of the DataFrame using the show() function:

df.show(5,0)

**Step 3: Exploratory data analysis**

Now that we have seen the variables present in this dataset, let’s perform some exploratory data analysis to further understand these data points:

1. Let’s start by counting the number of rows in the DataFrame :

df.count()

2.Try the below one

df.distinct().count()

3.Try with one more CSV File

df2=spark.read.csv('original.csv',header=True,escape="\"")

df2.show()

4.Select concept

df\_select=df2.select("first\_name","last\_name")

df\_select.show()

5.Filter concept

df\_select2=df2.filter(df2.first\_name=='Alvera')

df\_select2.show()

6.Filter with like

df\_filter=df2.filter(df2.first\_name.like("%lvera"))

df\_filter.show()

7.Filter with endswith

df\_filter=df2.filter(df2.first\_name.endswith('din'))

df\_filter.show()

**Step4: PySpark using sql**

**First example**

from pyspark.sql.types import \*

myschema=StructType([

StructField('id',IntegerType()),

StructField('first\_name',StringType()),

StructField('last\_name',StringType()),

StructField('gender',StringType()),

StructField('city',StringType()),

StructField('job\_title',StringType()),

StructField('Salary',StringType()),

StructField('latitude',FloatType()),

StructField('longitude',FloatType())

])

df3=spark.read.csv("original.csv",header=True,schema=myschema)

df3.show()

**Secondexample**

df3.createOrReplaceTempView("original")

query1=spark.sql('select \* from original')

query1.show()

**Thirdexample**

query2=spark.sql('select concat(first\_name," ",last\_name) as full\_name from original where gender="Female"')

query2.show()