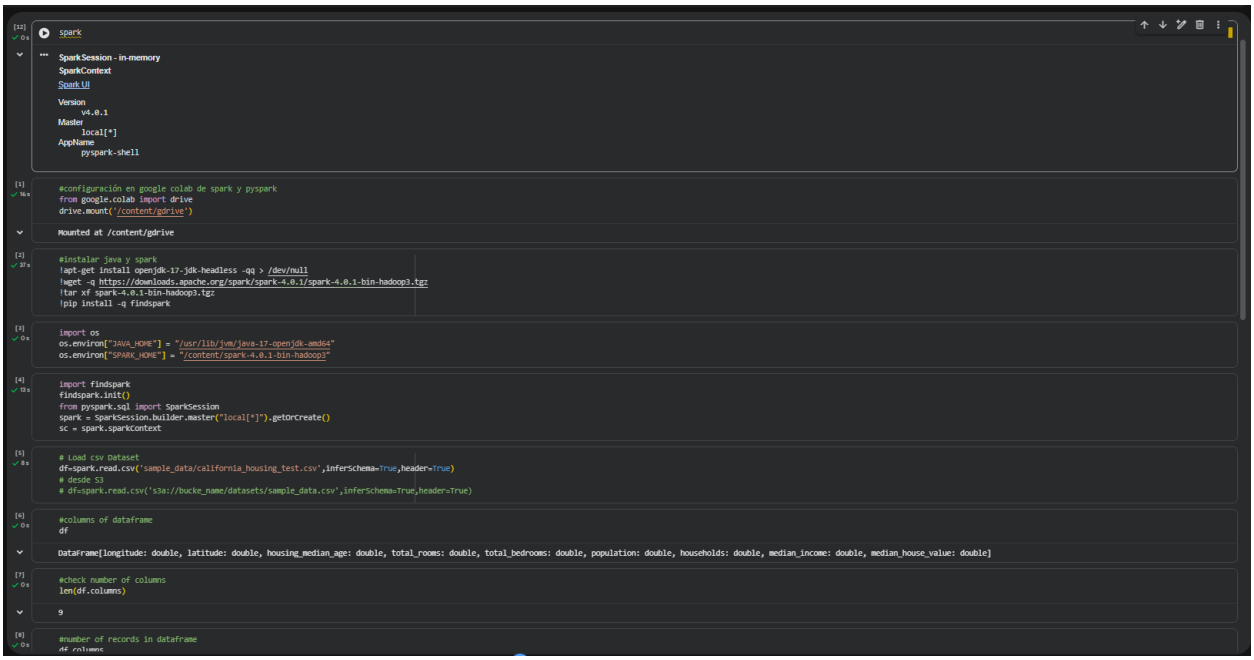
Spark en EMR y Google Colab



```
spark
```

```
SparkSession - in-memory
SparkContext
Spark UI

Version
    v4.0.1
Master
    local[*]
AppName
    pyspark-shell
```

```
#configuración en google colab de spark y pyspark
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
#instalar java y spark
!apt-get install openjdk-17-jdk-headless -qq > /dev/null
!wget -q https://downloads.apache.org/spark/spark-4.0.1/spark-4.0.1-bin-hadoop3.tgz
!tar xf spark-4.0.1-bin-hadoop3.tgz
!pip install -q findspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-17-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-4.0.1-bin-hadoop3"
```

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
```

```
# Load csv Dataset
df=spark.read.csv('sample_data/california_housing_test.csv',inferSchema=True,header=True)
# desde S3
# df=spark.read.csv('s3a://bucke_name/datasets/sample_data.csv',inferSchema=True,header=True)
```

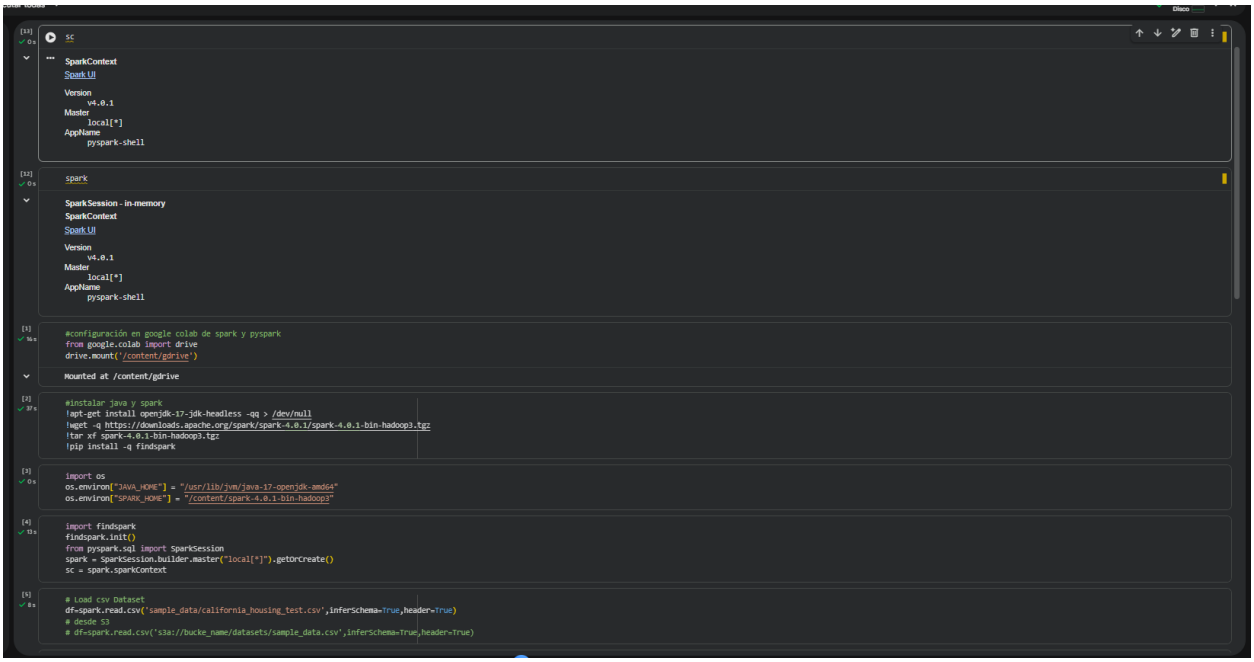
```
#columns of dataframe
df
```

```
DataFrame[longitude: double, latitude: double, housing_median_age: double, total_rooms: double, total_bedrooms: double, population: double, households: double, median_income: double, median_house_value: double]
```

```
#check number of columns
len(df.columns)
```

```
9
```

```
#number of records in dataframe
df.columns
```



```
sc
```

```
SparkContext
Spark UI

Version
    v4.0.1
Master
    local[*]
AppName
    pyspark-shell
```

```
spark
```

```
SparkSession - in-memory
SparkContext
Spark UI

Version
    v4.0.1
Master
    local[*]
AppName
    pyspark-shell
```

```
#configuración en google colab de spark y pyspark
from google.colab import drive
drive.mount('/content/gdrive')
```

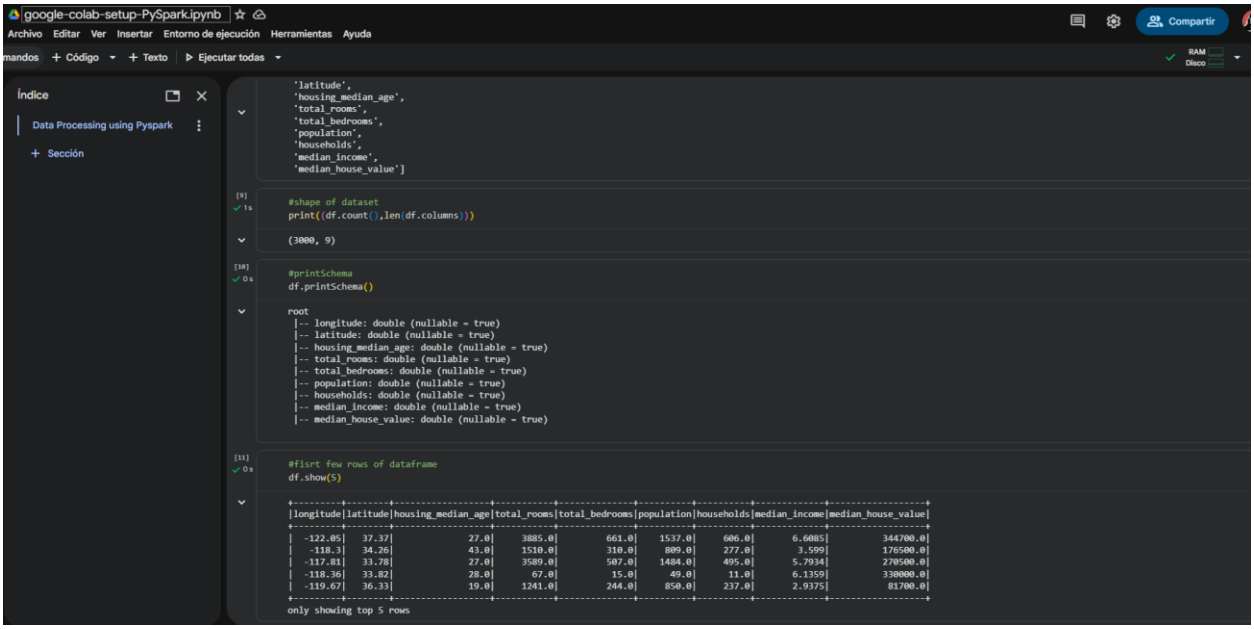
```
Mounted at /content/gdrive
```

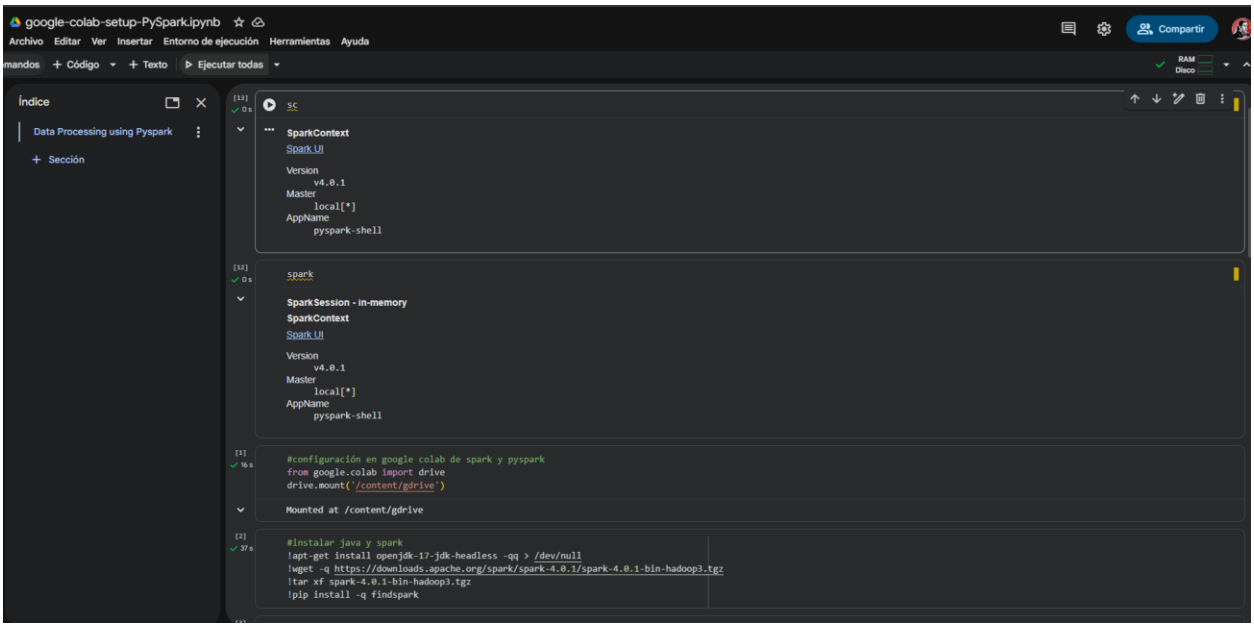
```
#instalar java y spark
!apt-get install openjdk-17-jdk-headless -qq > /dev/null
!wget -q https://downloads.apache.org/spark/spark-4.0.1/spark-4.0.1-bin-hadoop3.tgz
!tar xf spark-4.0.1-bin-hadoop3.tgz
!pip install -q findspark
```

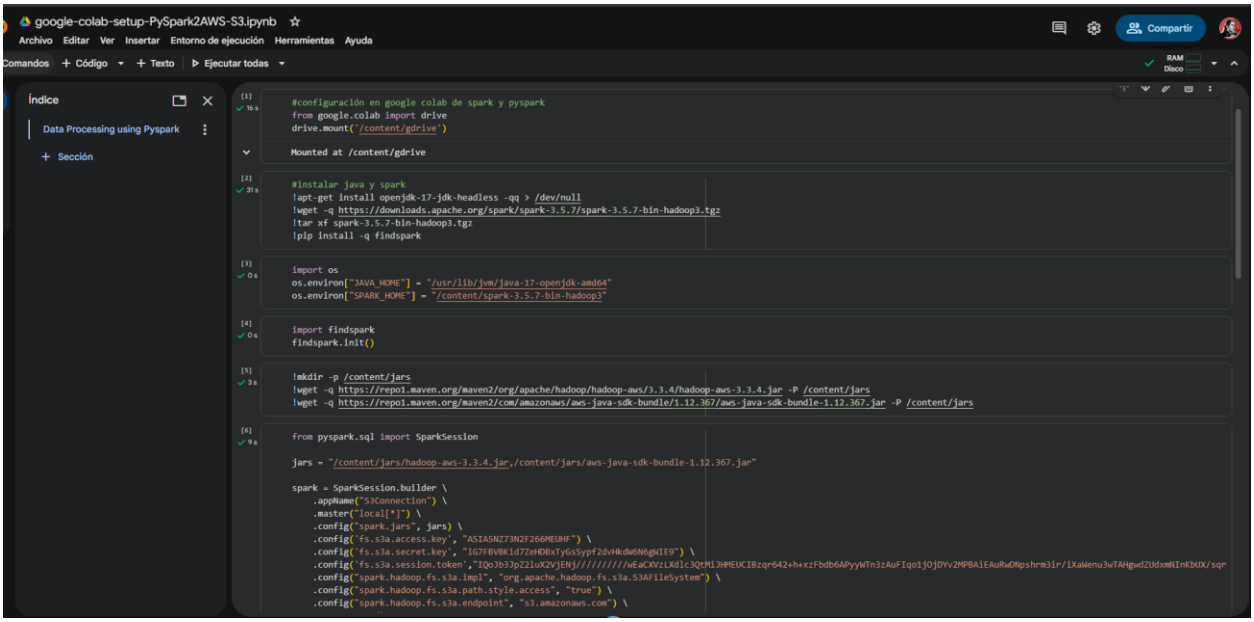
```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-17-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-4.0.1-bin-hadoop3"
```

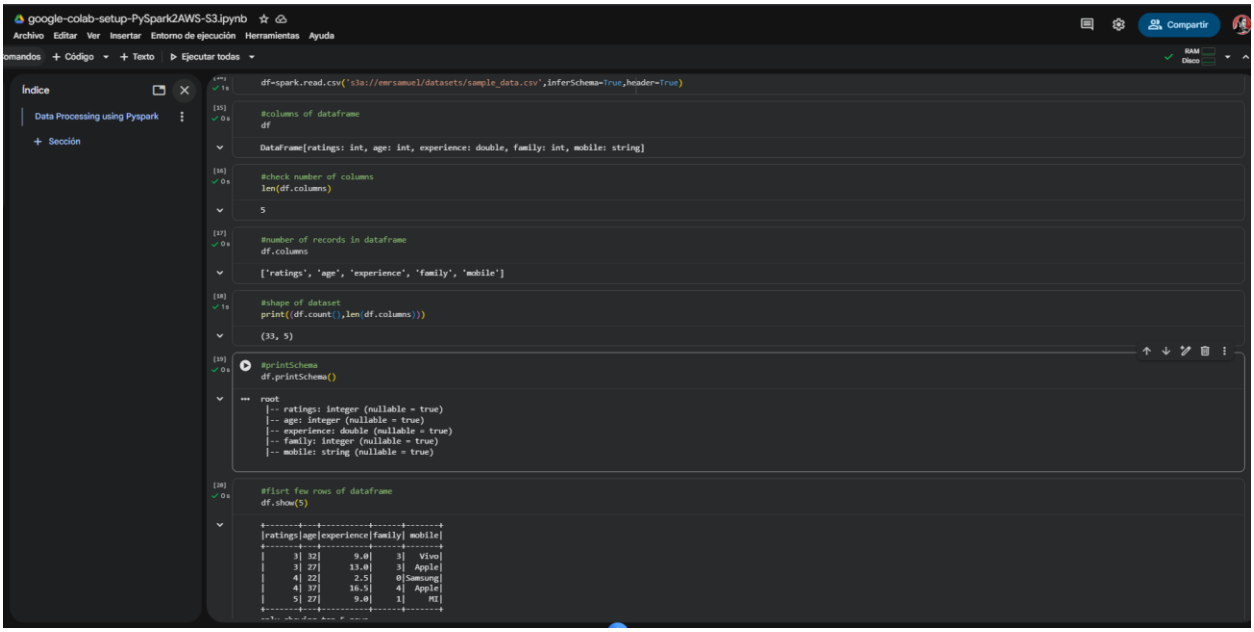
```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
```

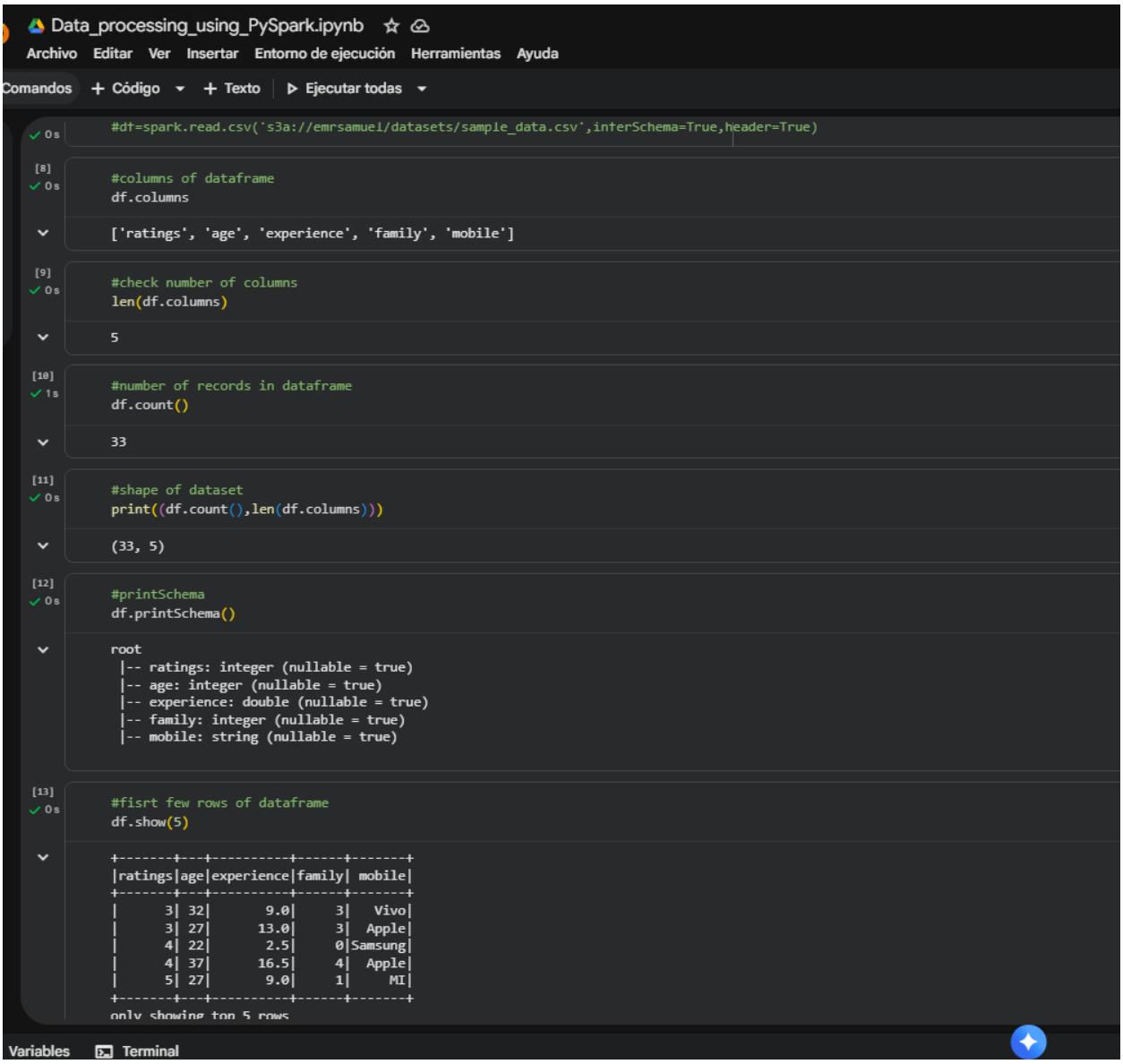
```
# Load csv Dataset
df=spark.read.csv('sample_data/california_housing_test.csv',inferSchema=True,header=True)
# desde S3
# df=spark.read.csv('s3a://bucke_name/datasets/sample_data.csv',inferSchema=True,header=True)
```

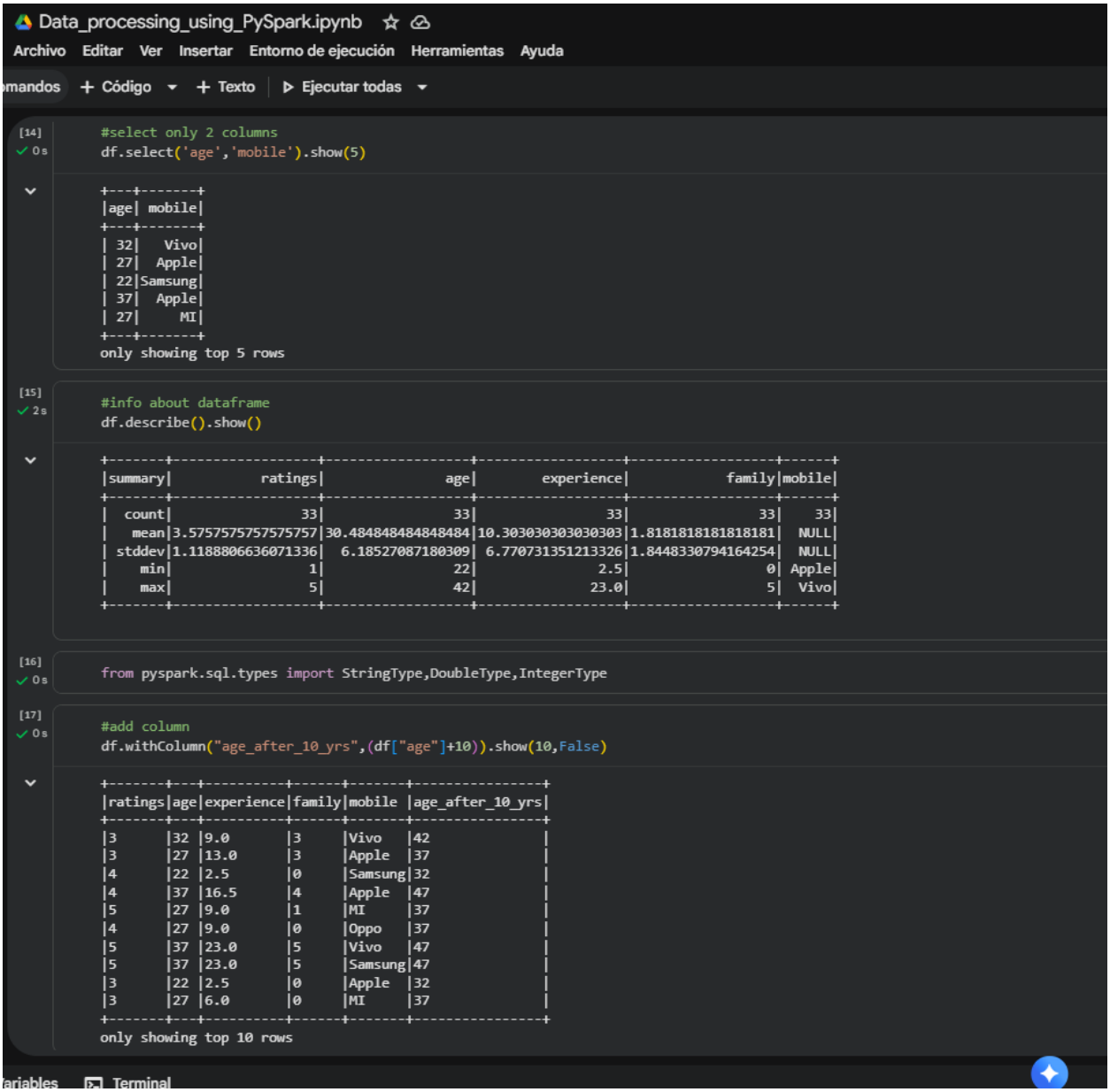

google-colab-setup-PySpark.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Compartir
+ Código + Texto Ejecutar todas
RAM
Disco
Índice
Data Processing using Pyspark
+ Sección
'latitude',
'housing_median_age',
'total_rooms',
'total_bedrooms',
'population',
'households',
'median_income',
'median_house_value']
#shape of dataset
print((df.count(),len(df.columns)))
(3000, 9)
#printSchema
df.printSchema()
root
|-- longitude: double (nullable = true)
|-- latitude: double (nullable = true)
|-- housing_median_age: double (nullable = true)
|-- total_rooms: double (nullable = true)
|-- total_bedrooms: double (nullable = true)
|-- population: double (nullable = true)
|-- households: double (nullable = true)
|-- median_income: double (nullable = true)
|-- median_house_value: double (nullable = true)
#fisrt few rows of dataframe
df.show(5)
+---------+--------+------------------+----------+-------------+----------+----------+-------------+------------------+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|median_house_value|
+---------+--------+------------------+----------+-------------+----------+----------+-------------+------------------+
-122.05	37.37	27.0	3885.0	661.0	1537.0	606.0	6.6085	344700.0
-118.3	34.26	43.0	1510.0	310.0	809.0	277.0	3.599	176500.0
-117.81	33.78	27.0	3589.0	507.0	1484.0	495.0	5.7934	270500.0
-118.36	33.82	28.0	67.0	15.0	49.0	11.0	6.1359	330000.0
-119.67	36.33	19.0	1241.0	244.0	850.0	237.0	2.9375	81700.0
+---------+--------+------------------+----------+-------------+----------+----------+-------------+------------------+
only showing top 5 rows

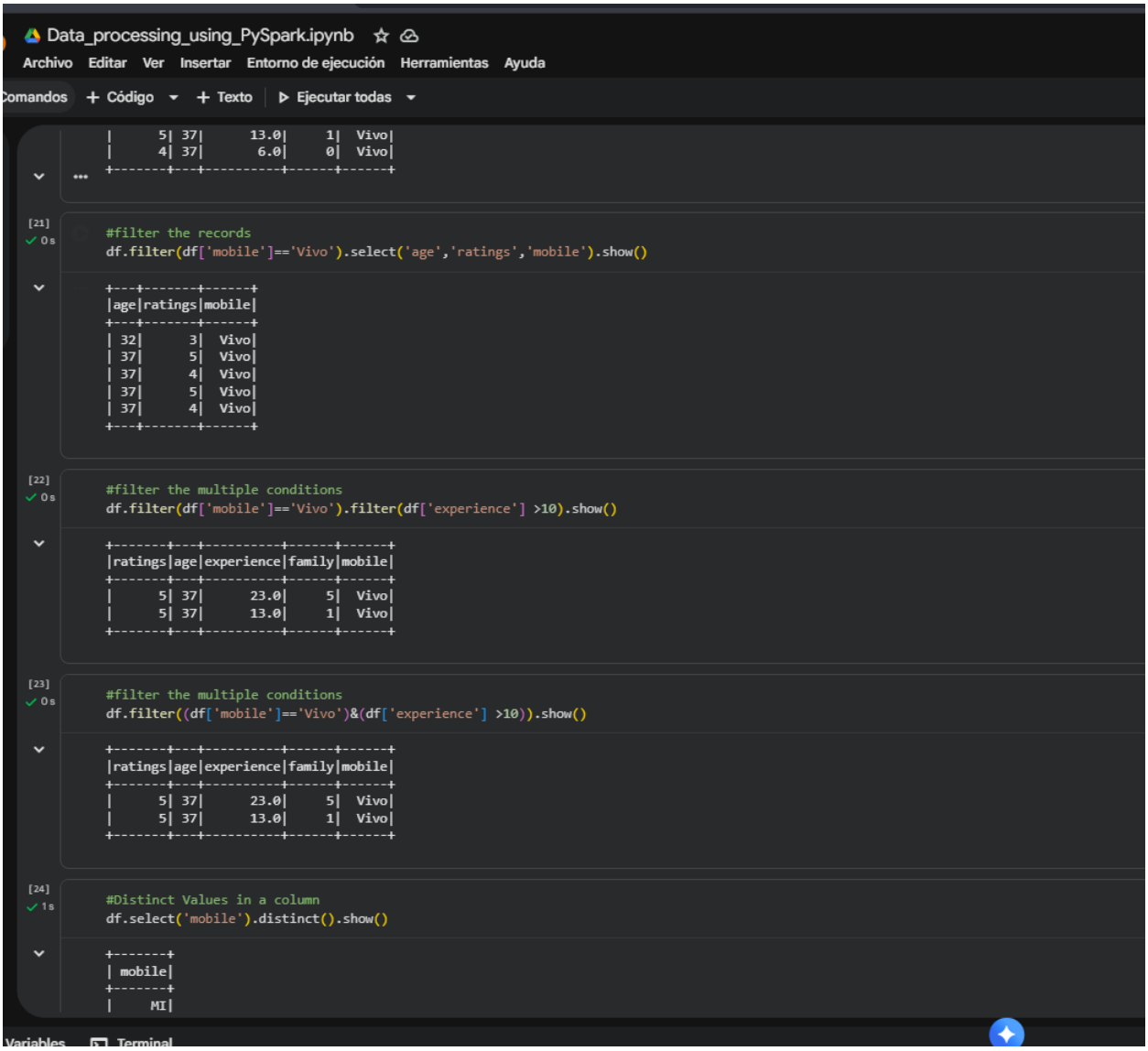

google-colab-setup-PySpark.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Compartir
+ Código + Texto Ejecutar todas
RAM
Disco
Índice
Data Processing using Pyspark
+ Sección
sc
SparkContext
Spark UI
Version
v4.0.1
Master
local[*]
AppName
pyspark-shell
spark
SparkSession - in-memory
SparkContext
Spark UI
Version
v4.0.1
Master
local[*]
AppName
pyspark-shell
#configuración en google colab de spark y pyspark
from google.colab import drive
drive.mount('/content/gdrive')
Mounted at /content/gdrive
#instalar java y spark
!apt-get install openjdk-17-jdk-headless -qq > /dev/null
!wget -q https://downloads.apache.org/spark/spark-4.0.1/spark-4.0.1-bin-hadoop3.tgz
!tar xf spark-4.0.1-bin-hadoop3.tgz
!pip install -q findspark

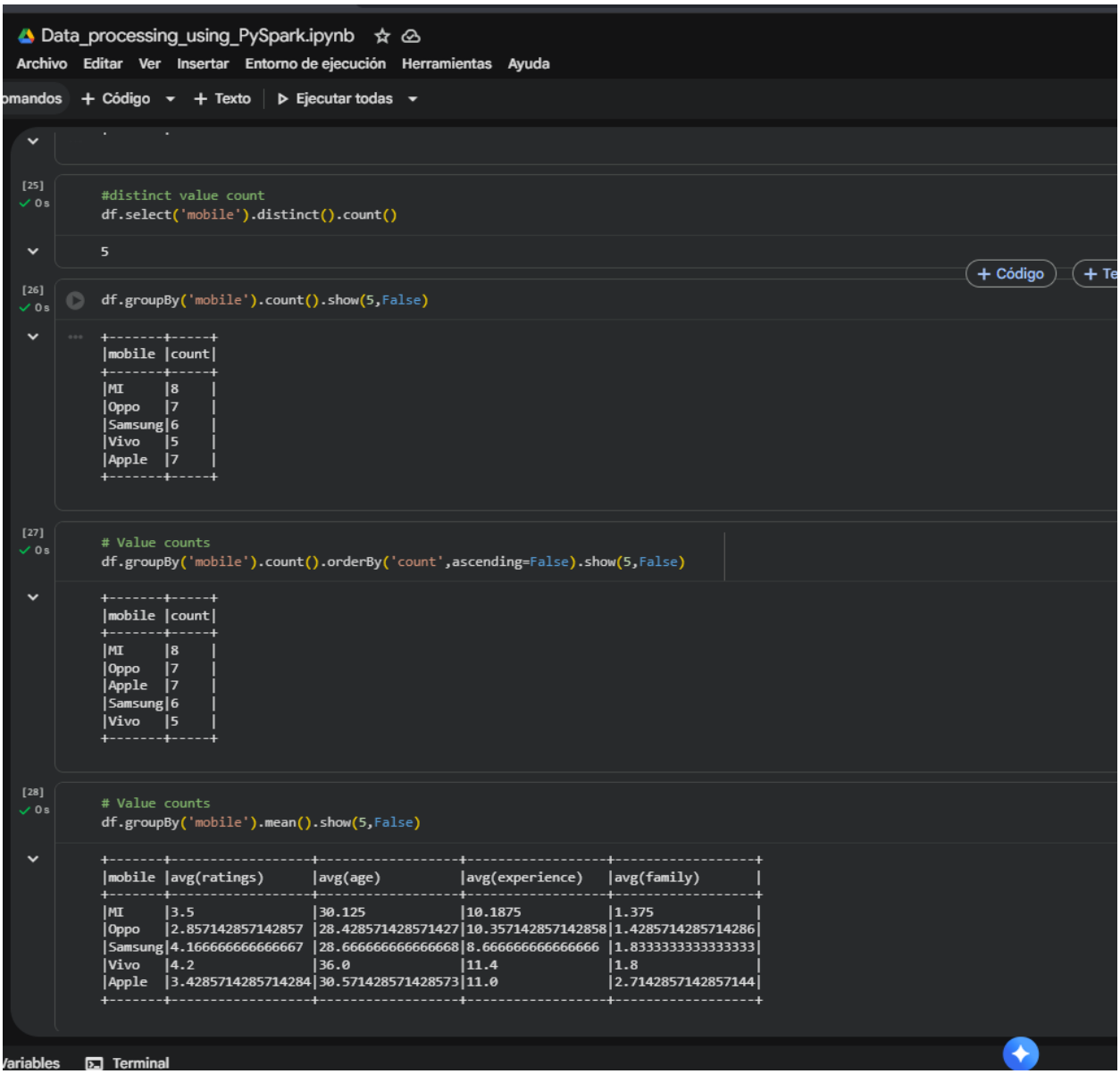

google-colab-setup-PySpark2AWS-S3.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Compartir
Comandos + Código + Texto Ejecutar todas
Índice
Data Processing using Pyspark
+ Sección
#configuración en google colab de spark y pyspark
from google.colab import drive
drive.mount('/content/gdrive')
Mounted at /content/gdrive
#instalar java y spark
!apt-get install openjdk-17-jdk-headless -qq > /dev/null
!wget -q https://downloads.apache.org/spark/spark-3.5.7/spark-3.5.7-bin-hadoop3.tgz
!tar xf spark-3.5.7-bin-hadoop3.tgz
!pip install -q findspark
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-17-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.7-bin-hadoop3"
import findspark
findspark.init()
!mkdir -p /content/jars
!wget -q https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-aws/3.3.4/hadoop-aws-3.3.4.jar -P /content/jars
!wget -q https://repo1.maven.org/maven2/com/amazonaws/aws-java-sdk-bundle/1.12.367/aws-java-sdk-bundle-1.12.367.jar -P /content/jars
from pyspark.sql import SparkSession
jars = "/content/jars/hadoop-aws-3.3.4.jar,/content/jars/aws-java-sdk-bundle-1.12.367.jar"
spark = SparkSession.builder \
.appName("S3Connection") \
.master("local[*]") \
.config("spark.jars", jars) \
.config('fs.s3a.access.key', "ASIASNZ73N2F266MEUHF") \
.config('fs.s3a.secret.key', "IG7FBVBKId7ZeHDBxTyGsSypf2dvHkdaGN6gWIE9") \
.config('fs.s3a.session.token',"IQoJb3JpZ2luX2VjENj/////////wEaCXVzLXd1c3QtMiJHMEUCIBzqr642+h+xzFbdb6APyyWTn3zAuFIqoijOjDYv2MPBAiEAuRwDNpshrm3ir/iXaWenu3wTAHgwdZUdxmNInKbUX/sqr
.config("spark.hadoop.fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem") \
.config("spark.hadoop.fs.s3a.path.style.access", "true") \
.config("spark.hadoop.fs.s3a.endpoint", "s3.amazonaws.com") \


google-colab-setup-PySpark2AWS-S3.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Compartir
Índice
Data Processing using Pyspark
+ Sección
df=spark.read.csv('s3a://emrsamuel/datasets/sample_data.csv',inferSchema=True,header=True)
#columns of dataframe
df
DataFrame[ratings: int, age: int, experience: double, family: int, mobile: string]
#check number of columns
len(df.columns)
5
#number of records in dataframe
df.columns
['ratings', 'age', 'experience', 'family', 'mobile']
#shape of dataset
print((df.count(),len(df.columns)))
(33, 5)
#printSchema
df.printSchema()
root
|-- ratings: integer (nullable = true)
|-- age: integer (nullable = true)
|-- experience: double (nullable = true)
|-- family: integer (nullable = true)
|-- mobile: string (nullable = true)
#fisrt few rows of dataframe
df.show(5)
+-------+---+----------+------+-------+
|ratings|age|experience|family| mobile|
+-------+---+----------+------+-------+
3	32	9.0	3	Vivo
3	27	13.0	3	Apple
4	22	2.5	0	Samsung
4	37	16.5	4	Apple
5	27	9.0	1	MI
+-------+---+----------+------+-------+


Data_processing_using_PySpark.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Comandos + Código + Texto Ejecutar todas
0 s
#df=spark.read.csv('s3a://emrsamuel/datasets/sample_data.csv',interSchema=True,header=True)
[8]
0 s
#columns of dataframe
df.columns
['ratings', 'age', 'experience', 'family', 'mobile']
[9]
0 s
#check number of columns
len(df.columns)
5
[10]
1 s
#number of records in dataframe
df.count()
33
[11]
0 s
#shape of dataset
print((df.count(),len(df.columns)))
(33, 5)
[12]
0 s
#printSchema
df.printSchema()
root
|-- ratings: integer (nullable = true)
|-- age: integer (nullable = true)
|-- experience: double (nullable = true)
|-- family: integer (nullable = true)
|-- mobile: string (nullable = true)
[13]
0 s
#fisrt few rows of dataframe
df.show(5)
+-------+---+----------+------+-------+
|ratings|age|experience|family| mobile|
+-------+---+----------+------+-------+
3	32	9.0	3	Vivo
3	27	13.0	3	Apple
4	22	2.5	0	Samsung
4	37	16.5	4	Apple
5	27	9.0	1	MI
+-------+---+----------+------+-------+
only showing top 5 rows
Variables
Terminal


Data_processing_using_PySpark.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
mandos + Código + Texto Ejecutar todas
[14]
0 s
#select only 2 columns
df.select('age','mobile').show(5)
+---+-------+
|age| mobile|
+---+-------+
32	Vivo
27	Apple
22	Samsung
37	Apple
27	MI
+---+-------+
only showing top 5 rows
[15]
2 s
#info about dataframe
df.describe().show()
+-------+------------------+------------------+------------------+------------------+------+
|summary| ratings| age| experience| family|mobile|
+-------+------------------+------------------+------------------+------------------+------+
count	33	33	33	33	33
mean	3.5757575757575757	30.484848484848484	10.303030303030303	1.8181818181818181	NULL
stddev	1.1188806636071336	6.18527087180309	6.770731351213326	1.8448330794164254	NULL
min	1	22	2.5	0	Apple
max	5	42	23.0	5	Vivo
+-------+------------------+------------------+------------------+------------------+------+
[16]
0 s
from pyspark.sql.types import StringType,DoubleType,IntegerType
[17]
0 s
#add column
df.withColumn("age_after_10_yrs",(df["age"]+10)).show(10,False)
+-------+---+----------+------+-------+---------------+
|ratings|age|experience|family|mobile |age_after_10_yrs|
+-------+---+----------+------+-------+---------------+
3	32	9.0	3	Vivo	42
3	27	13.0	3	Apple	37
4	22	2.5	0	Samsung	32
4	37	16.5	4	Apple	47
5	27	9.0	1	MI	37
4	27	9.0	0	Oppo	37
5	37	23.0	5	Vivo	47
5	37	23.0	5	Samsung	47
3	22	2.5	0	Apple	32
3	27	6.0	0	MI	37
+-------+---+----------+------+-------+---------------+
only showing top 10 rows
ariables
Terminal


Data_processing_using_PySpark.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Comandos + Código + Texto ▷ Ejecutar todas
| 5| 37| 13.0| 1| Vivo|
| 4| 37| 6.0| 0| Vivo|
[21] 0 s
#filter the records
df.filter(df['mobile']=='Vivo').select('age','ratings','mobile').show()
age	ratings	mobile
32	3	Vivo
37	5	Vivo
37	4	Vivo
37	5	Vivo
37	4	Vivo
[22] 0 s
#filter the multiple conditions
df.filter(df['mobile']=='Vivo').filter(df['experience'] >10).show()
ratings	age	experience	family	mobile
5	37	23.0	5	Vivo
5	37	13.0	1	Vivo
[23] 0 s
#filter the multiple conditions
df.filter((df['mobile']=='Vivo')&(df['experience'] >10)).show()
ratings	age	experience	family	mobile
5	37	23.0	5	Vivo
5	37	13.0	1	Vivo
[24] 1 s
#Distinct Values in a column
df.select('mobile').distinct().show()
| mobile|
| MI|
Variables Terminal


Data_processing_using_PySpark.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
omandos + Código + Texto Ejecutar todas
[25] 0 s
#distinct value count
df.select('mobile').distinct().count()
5
+ Código
[26] 0 s
df.groupBy('mobile').count().show(5,False)
+------+-----+
|mobile |count|
+------+-----+
MI	8
Oppo	7
Samsung	6
Vivo	5
Apple	7
+------+-----+
[27] 0 s
# Value counts
df.groupBy('mobile').count().orderBy('count',ascending=False).show(5,False)
+------+-----+
|mobile |count|
+------+-----+
MI	8
Oppo	7
Apple	7
Samsung	6
Vivo	5
+------+-----+
[28] 0 s
# Value counts
df.groupBy('mobile').mean().show(5,False)
+------+-----------------+-----------------+-----------------+-----------------+
|mobile |avg(ratings) |avg(age) |avg(experience) |avg(family) |
+------+-----------------+-----------------+-----------------+-----------------+
MI	3.5	30.125	10.1875	1.375
Oppo	2.857142857142857	28.428571428571427	10.357142857142858	1.4285714285714286
Samsung	4.166666666666667	28.666666666666668	8.666666666666666	1.8333333333333333
Vivo	4.2	36.0	11.4	1.8
Apple	3.4285714285714284	30.571428571428573	11.0	2.7142857142857144
+------+-----------------+-----------------+-----------------+-----------------+
Variables
Terminal

Data_processing_using_PySpark.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

mandos + Código + Texto ▷ Ejecutar todas

```
|5      |27 |9.0       |1     |MI     |45.0   |
|4      |27 |9.0       |0     |Oppo   |36.0   |
|5      |37 |23.0      |5     |Vivo   |115.0  |
|5      |37 |23.0      |5     |Samsung|115.0  |
|3      |22 |2.5       |0     |Apple  |7.5    |
|3      |27 |6.0       |0     |MI     |18.0   |
+-------+---+----------+------+-------+-------+
only showing top 10 rows
```

[43] ✓ 0s

```
#duplicate values
df.count()
```

```
33
```

[44] ✓ 0s

```
#drop duplicate values
df=df.dropDuplicates()
```

[45] ✓ 0s

```
#validate new count
df.count()
```

```
26
```

[46] ✓ 0s

```
#drop column of dataframe
df_new=df.drop('mobile')
```

[47] ✓ 0s

```
df_new.show(10)
```

```
+-------+---+----------+------+
|ratings|age|experience|family|
+-------+---+----------+------+
|      3| 32|       9.0|     3|
|      4| 22|       2.5|     0|
|      5| 27|       6.0|     0|
|      4| 22|       6.0|     1|
|      3| 27|       6.0|     0|
|      2| 32|      16.5|     2|
|      4| 27|       9.0|     0|
|      2| 27|       9.0|     2|
|      3| 37|      16.5|     5|
|      4| 27|       6.0|     1|
+-------+---+----------+------+
only showing top 10 rows
```

[48] ✓ 0s

```
# saving file to csv
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
3	application_1763344109923_0006	pyspark	idle	Link	Link	None	✓

```
SparkSession available as 'spark'.
```

In [2]:
```
spark
```

```
<pyspark.sql.session.SparkSession object at 0xffffa58c2730>
```

In [*]:
```
# Load csv Dataset
# desde gdrvie
#df=spark.read.csv('gdrive/MyDrive/st0263-252/bigdata/datasets/sample_data.csv',inferSchema=True,header=True)

# desde local
#df=spark.read.csv('../datasets/sample_data.csv',inferSchema=True,header=True)
# desde S3
df=spark.read.csv('s3://emrsamuel/datasets/sample_data.csv',inferSchema=True,header=True)
```

In [*]:
```
#columns of dataframe
df.columns
```

In [*]:
```
#check number of columns
len(df.columns)
```

In [*]:
```
#number of records in dataframe
df.count()
```

In [*]:
```
#shape of dataset
print((df.count(),len(df.columns)))
```

In [*]:
```
#printSchema
df.printSchema()
```

In [ ]:
```
#fisrt few rows of dataframe
df.show(5)
```

In [ ]:
```
#select only 2 columns
df.select('age','mobile').show(5)
```

jupyterhub Data_processing_using_PySpark Last Checkpoint: hace 3 minutos (autosaved)

Logout | Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted | PySpark

Run | Code

```
df=spark.read.csv('s3://emrsamuel/datasets/sample_data.csv',inferSchema=True,header=True)
```

In [3]:
```
#columns of dataframe
df.columns
```

```
['ratings', 'age', 'experience', 'family', 'mobile']
```

In [4]:
```
#check number of columns
len(df.columns)
```

```
5
```

In [5]:
```
#number of records in dataframe
df.count()
```

```
33
```

In [6]:
```
#shape of dataset
print((df.count(),len(df.columns)))
```

```
(33, 5)
```

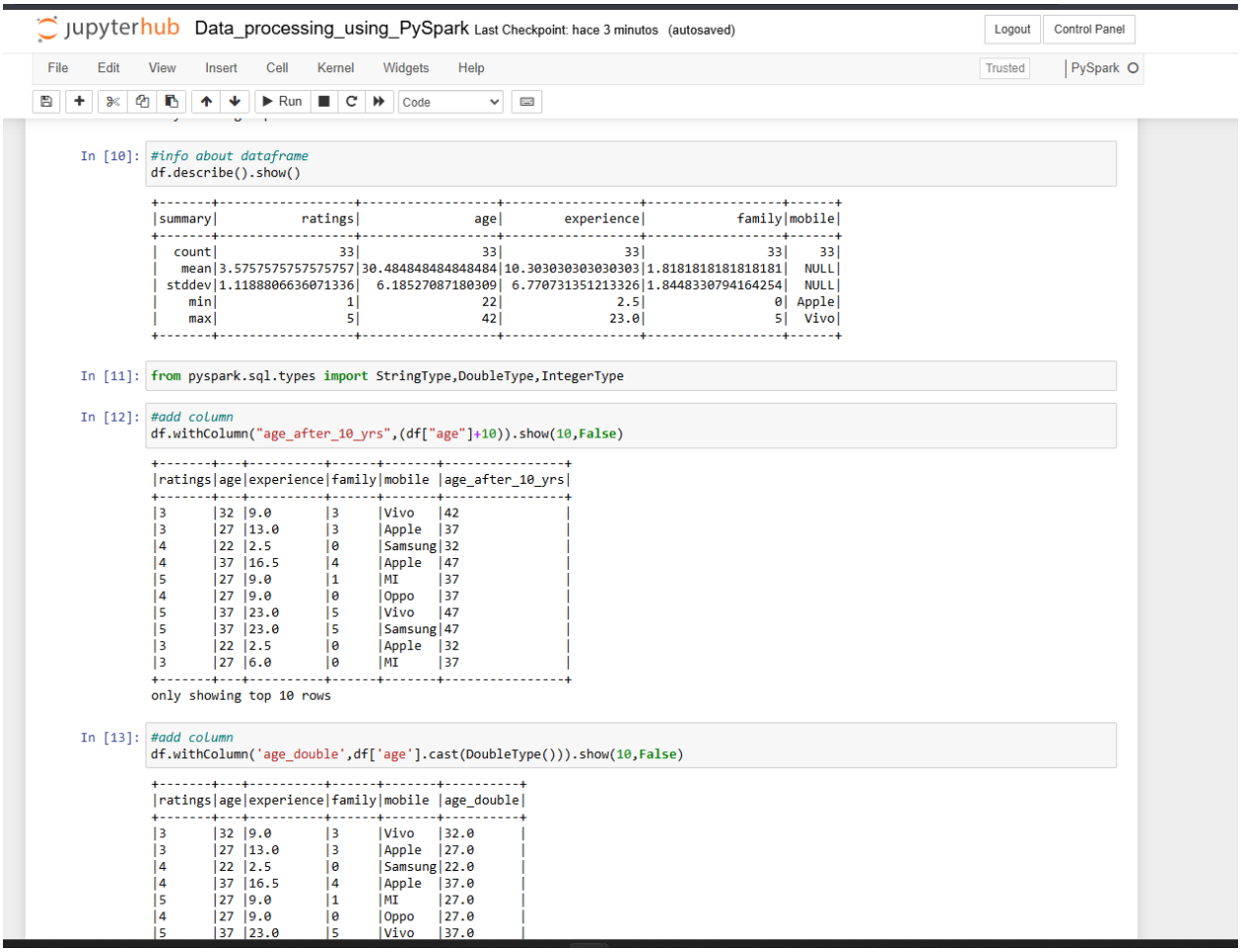
In [7]:
```
#printSchema
df.printSchema()
```

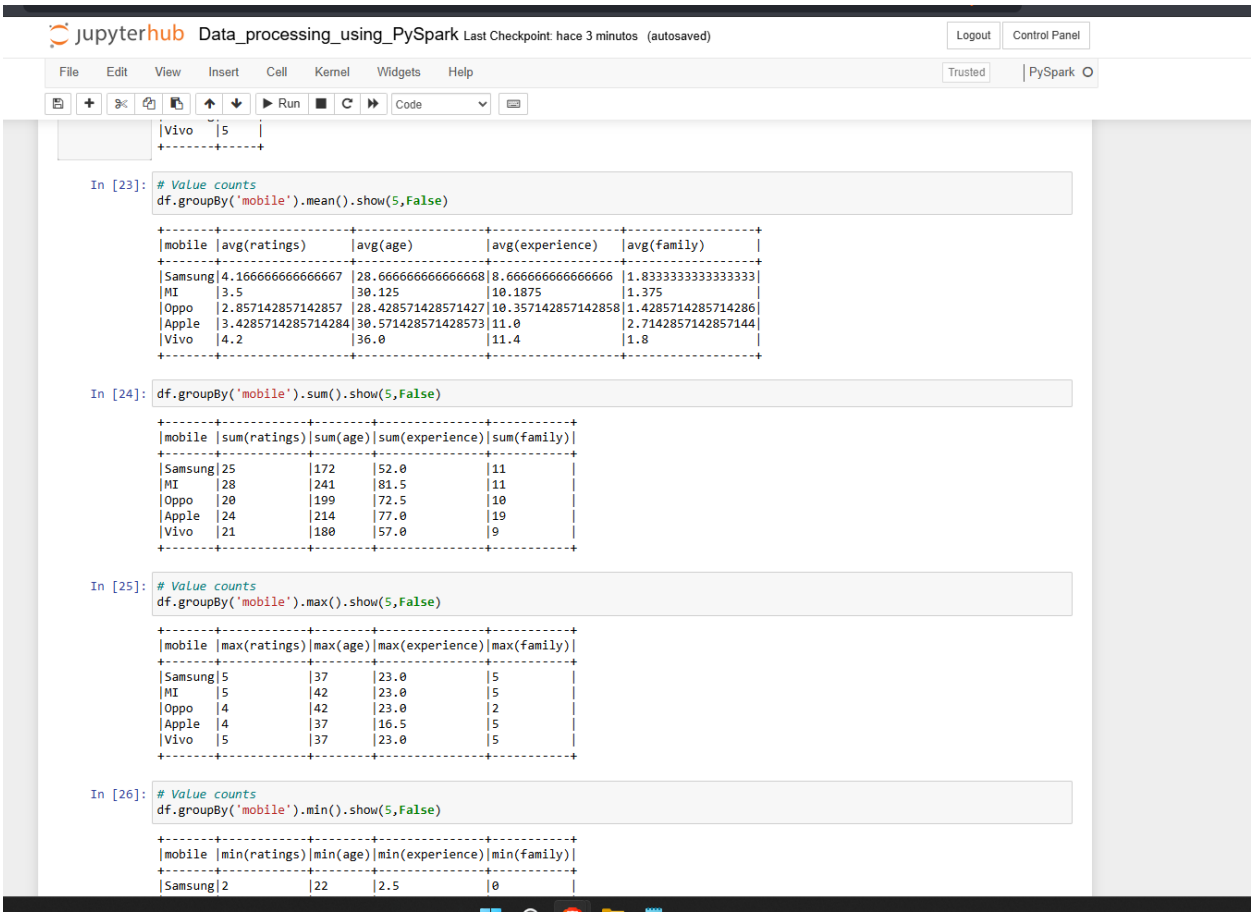
```
root
 |-- ratings: integer (nullable = true)
 |-- age: integer (nullable = true)
 |-- experience: double (nullable = true)
 |-- family: integer (nullable = true)
 |-- mobile: string (nullable = true)
```

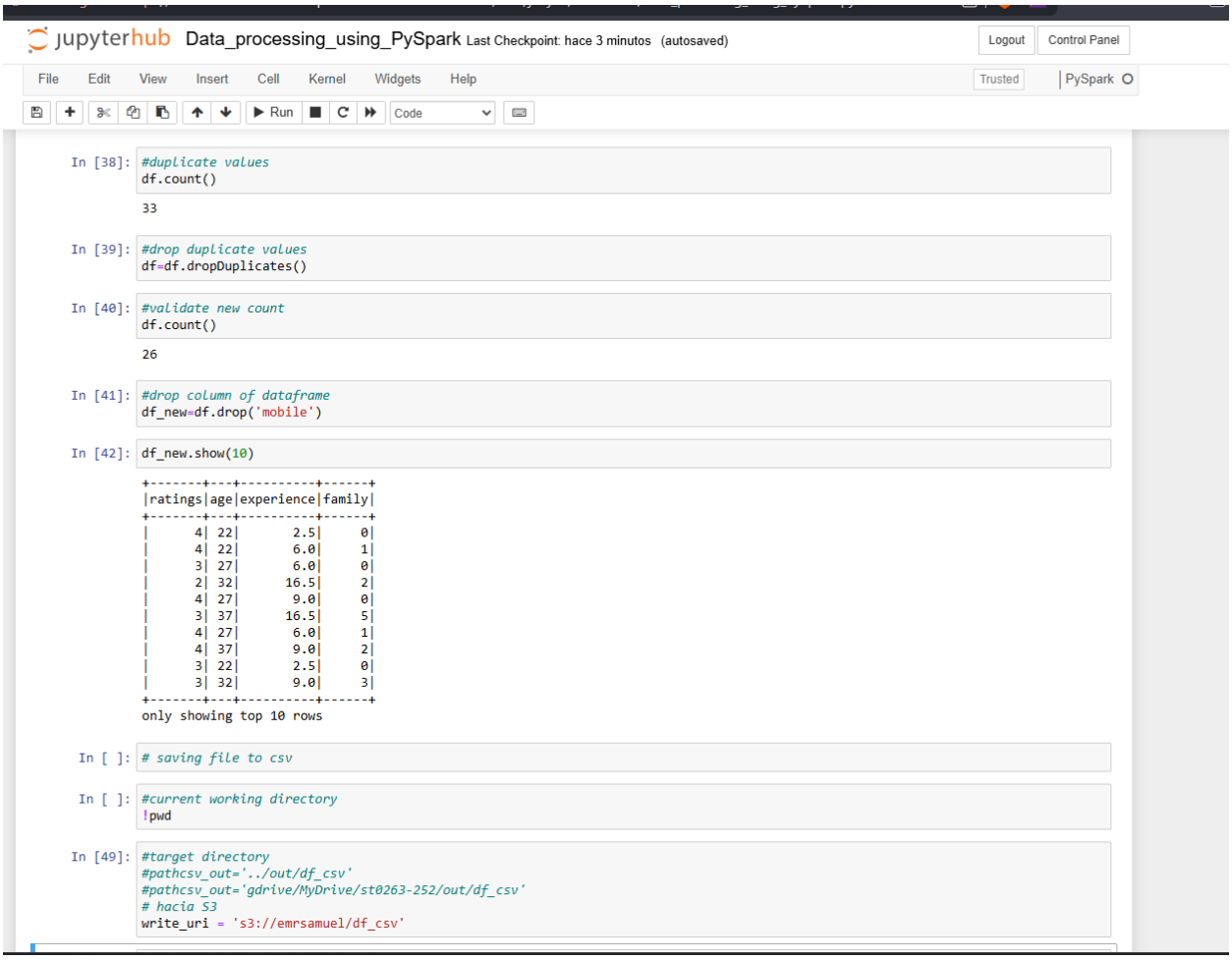
In [8]:
```
#fisrt few rows of dataframe
df.show(5)
```

```
+-------+---+----------+------+-------+
|ratings|age|experience|family| mobile|
+-------+---+----------+------+-------+
|      3| 32|       9.0|     3|   Vivo|
|      3| 27|      13.0|     3|  Apple|
|      4| 22|       2.5|     0|Samsung|
|      4| 37|      16.5|     4|  Apple|
|      5| 27|       9.0|     1|     MI|
+-------+---+----------+------+-------+
only showing top 5 rows
```

In [9]:
```
#select only 2 columns
```


jupyterhub Data_processing_using_PySpark Last Checkpoint: hace 3 minutos (autosaved)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
PySpark
Run
Code
In [10]:
#info about dataframe
df.describe().show()
+-------+------------------+------------------+------------------+------------------+------+
|summary| ratings| age| experience| family|mobile|
+-------+------------------+------------------+------------------+------------------+------+
count	33	33	33	33	33
mean	3.5757575757575757	30.484848484848484	10.303030303030303	1.8181818181818181	NULL
stddev	1.1188806636071336	6.18527087180309	6.770731351213326	1.8448330794164254	NULL
min	1	22	2.5	0	Apple
max	5	42	23.0	5	Vivo
+-------+------------------+------------------+------------------+------------------+------+
In [11]:
from pyspark.sql.types import StringType,DoubleType,IntegerType
In [12]:
#add column
df.withColumn("age_after_10_yrs",(df["age"]+10)).show(10,False)
+-------+---+----------+------+-------+---------------+
|ratings|age|experience|family|mobile |age_after_10_yrs|
+-------+---+----------+------+-------+---------------+
3	32	9.0	3	Vivo	42
3	27	13.0	3	Apple	37
4	22	2.5	0	Samsung	32
4	37	16.5	4	Apple	47
5	27	9.0	1	MI	37
4	27	9.0	0	Oppo	37
5	37	23.0	5	Vivo	47
5	37	23.0	5	Samsung	47
3	22	2.5	0	Apple	32
3	27	6.0	0	MI	37
+-------+---+----------+------+-------+---------------+
only showing top 10 rows
In [13]:
#add column
df.withColumn('age_double',df['age'].cast(DoubleType())).show(10,False)
+-------+---+----------+------+-------+----------+
|ratings|age|experience|family|mobile |age_double|
+-------+---+----------+------+-------+----------+
3	32	9.0	3	Vivo	32.0
3	27	13.0	3	Apple	27.0
4	22	2.5	0	Samsung	22.0
4	37	16.5	4	Apple	37.0
5	27	9.0	1	MI	27.0
4	27	9.0	0	Oppo	27.0
5	37	23.0	5	Vivo	37.0


jupyterhub Data_processing_using_PySpark Last Checkpoint: hace 3 minutos (autosaved)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
PySpark
Run
Code
|Vivo |5 |
+-------+-----+
In [23]:
# Value counts
df.groupBy('mobile').mean().show(5,False)
+-------+------------------+------------------+------------------+------------------+
|mobile |avg(ratings) |avg(age) |avg(experience) |avg(family) |
+-------+------------------+------------------+------------------+------------------+
Samsung	4.166666666666667	28.666666666666668	8.666666666666666	1.8333333333333333
MI	3.5	30.125	10.1875	1.375
Oppo	2.857142857142857	28.428571428571427	10.357142857142858	1.4285714285714286
Apple	3.4285714285714284	30.571428571428573	11.0	2.7142857142857144
Vivo	4.2	36.0	11.4	1.8
+-------+------------------+------------------+------------------+------------------+
In [24]:
df.groupBy('mobile').sum().show(5,False)
+-------+------------+--------+---------------+-----------+
|mobile |sum(ratings)|sum(age)|sum(experience)|sum(family)|
+-------+------------+--------+---------------+-----------+
Samsung	25	172	52.0	11
MI	28	241	81.5	11
Oppo	20	199	72.5	10
Apple	24	214	77.0	19
Vivo	21	180	57.0	9
+-------+------------+--------+---------------+-----------+
In [25]:
# Value counts
df.groupBy('mobile').max().show(5,False)
+-------+------------+--------+---------------+-----------+
|mobile |max(ratings)|max(age)|max(experience)|max(family)|
+-------+------------+--------+---------------+-----------+
Samsung	5	37	23.0	5
MI	5	42	23.0	5
Oppo	4	42	23.0	2
Apple	4	37	16.5	5
Vivo	5	37	23.0	5
+-------+------------+--------+---------------+-----------+
In [26]:
# Value counts
df.groupBy('mobile').min().show(5,False)
+-------+------------+--------+---------------+-----------+
|mobile |min(ratings)|min(age)|min(experience)|min(family)|
+-------+------------+--------+---------------+-----------+
|Samsung|2 |22 |2.5 |0 |


jupyterhub Data_processing_using_PySpark Last Checkpoint: hace 3 minutos (autosaved)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
PySpark
Run
Code
In [38]:
#duplicate values
df.count()
33
In [39]:
#drop duplicate values
df=df.dropDuplicates()
In [40]:
#validate new count
df.count()
26
In [41]:
#drop column of dataframe
df_new=df.drop('mobile')
In [42]:
df_new.show(10)
+-------+---+----------+------+
|ratings|age|experience|family|
+-------+---+----------+------+
4	22	2.5	0
4	22	6.0	1
3	27	6.0	0
2	32	16.5	2
4	27	9.0	0
3	37	16.5	5
4	27	6.0	1
4	37	9.0	2
3	22	2.5	0
3	32	9.0	3
+-------+---+----------+------+
only showing top 10 rows
In [ ]:
saving file to csv
In [ ]:
#current working directory
!pwd
In [49]:
#target directory
#pathcsv_out='../out/df_csv'
#pathcsv_out='gdrive/MyDrive/st0263-252/out/df_csv'
# hacia S3
write_uri = 's3://emrsamuel/df_csv'

jupyterhub Data_processing_using_PySpark Last Checkpoint: hace 4 minutos (autosaved)

Logout | Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted | PySpark

Run | Code

In [42]:
```
df_new.show(10)
```

```
+-------+---+----------+------+
|ratings|age|experience|family|
+-------+---+----------+------+
|      4| 22|       2.5|     0|
|      4| 22|       6.0|     1|
|      3| 27|       6.0|     0|
|      2| 32|      16.5|     2|
|      4| 27|       9.0|     0|
|      3| 37|      16.5|     5|
|      4| 27|       6.0|     1|
|      4| 37|       9.0|     2|
|      3| 22|       2.5|     0|
|      3| 32|       9.0|     3|
+-------+---+----------+------+
only showing top 10 rows
```

In [ ]:
```
# saving file to csv
```

In [ ]:
```
#current working directory
!pwd
```

In [49]:
```
#target directory
#pathcsv_out='../out/df_csv'
#pathcsv_out='gdrive/MyDrive/st0263-252/out/df_csv'
# hacia S3
write_uri = 's3://emrsamuel/df_csv'
```

In [51]:
```
#save the dataframe as single csv
df.coalesce(1).write.format("csv").option("header","true").save(write_uri)
```

In [ ]:
```
# parquet
```

In [52]:
```
#target location
#pathparquet_out='../out/df_parquet'
#pathparquet_out='gdrive/MyDrive/st0263-252/out/df_parquet'

# hacia S3
write_uri='s3://emrsamuel/df_parquet'
```

In [54]:
```
#save the data into parquet format
df.write.mode("overwrite").format("parquet").save(write_uri)
```

emrsamuel > df_csv/

df_csv/

Objetos | Propiedades

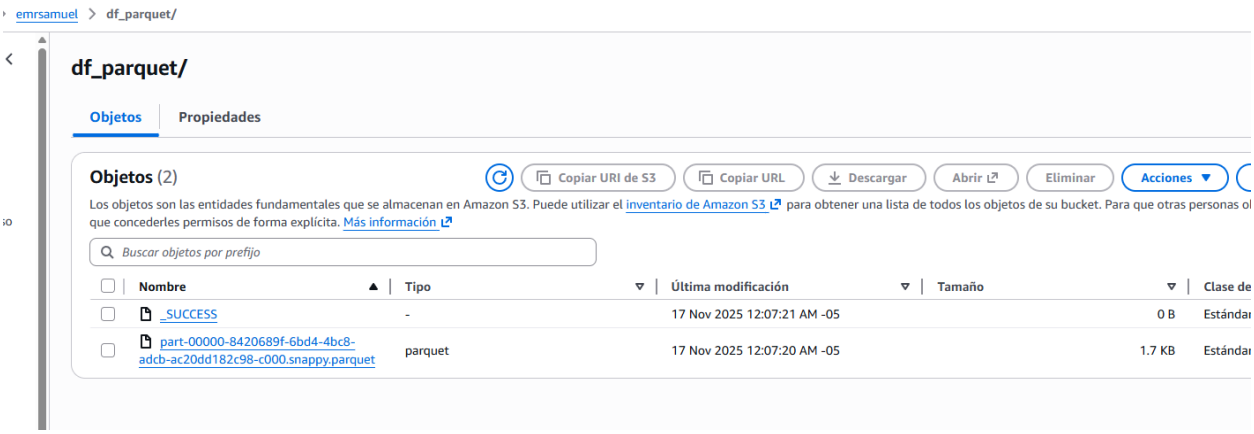
Objetos (2)

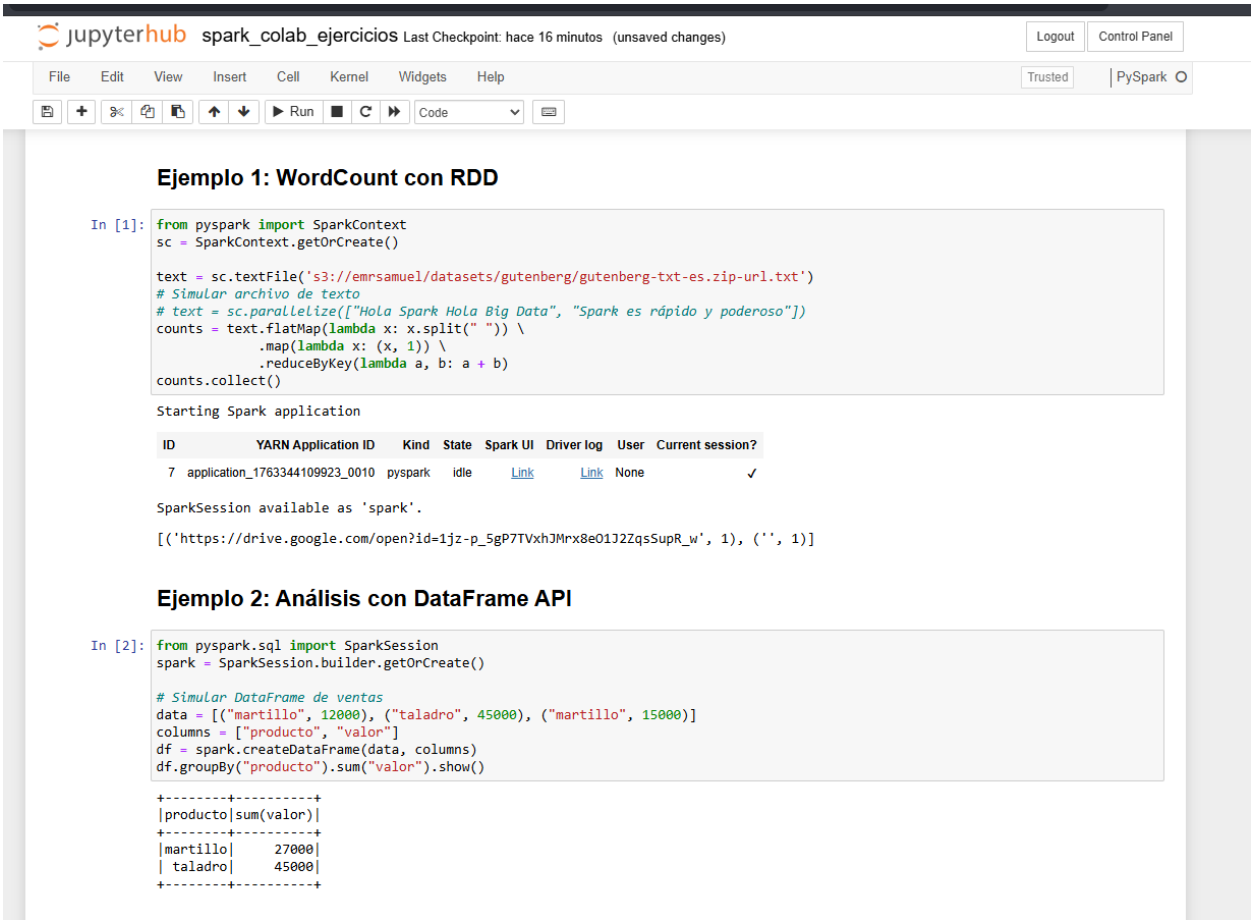
Copiar URI de S3 | Copiar URL | Descargar | Abrir | Eliminar | Acciones | Crear carpet

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el inventario de Amazon S3 para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acces
que concederles permisos de forma explícita. Más información

Buscar objetos por prefijo

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenami
_SUCCESS	-	17 Nov 2025 12:06:34 AM -05	0 B	Estándar
part-00000-94c7f10f-b96b-4e8b-97a6-055176b5a4f6-c000.csv	csv	17 Nov 2025 12:06:34 AM -05	474.0 B	Estándar





Ejemplo 1: WordCount con RDD

In [1]:
```
from pyspark import SparkContext
sc = SparkContext.getOrCreate()

text = sc.textFile('s3://emrsamuel/datasets/gutenberg/gutenberg-txt-es.zip-url.txt')
# Simular archivo de texto
# text = sc.parallelize(["Hola Spark Hola Big Data", "Spark es rápido y poderoso"])
counts = text.flatMap(lambda x: x.split(" ")) \
             .map(lambda x: (x, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.collect()
```

```
Starting Spark application
```

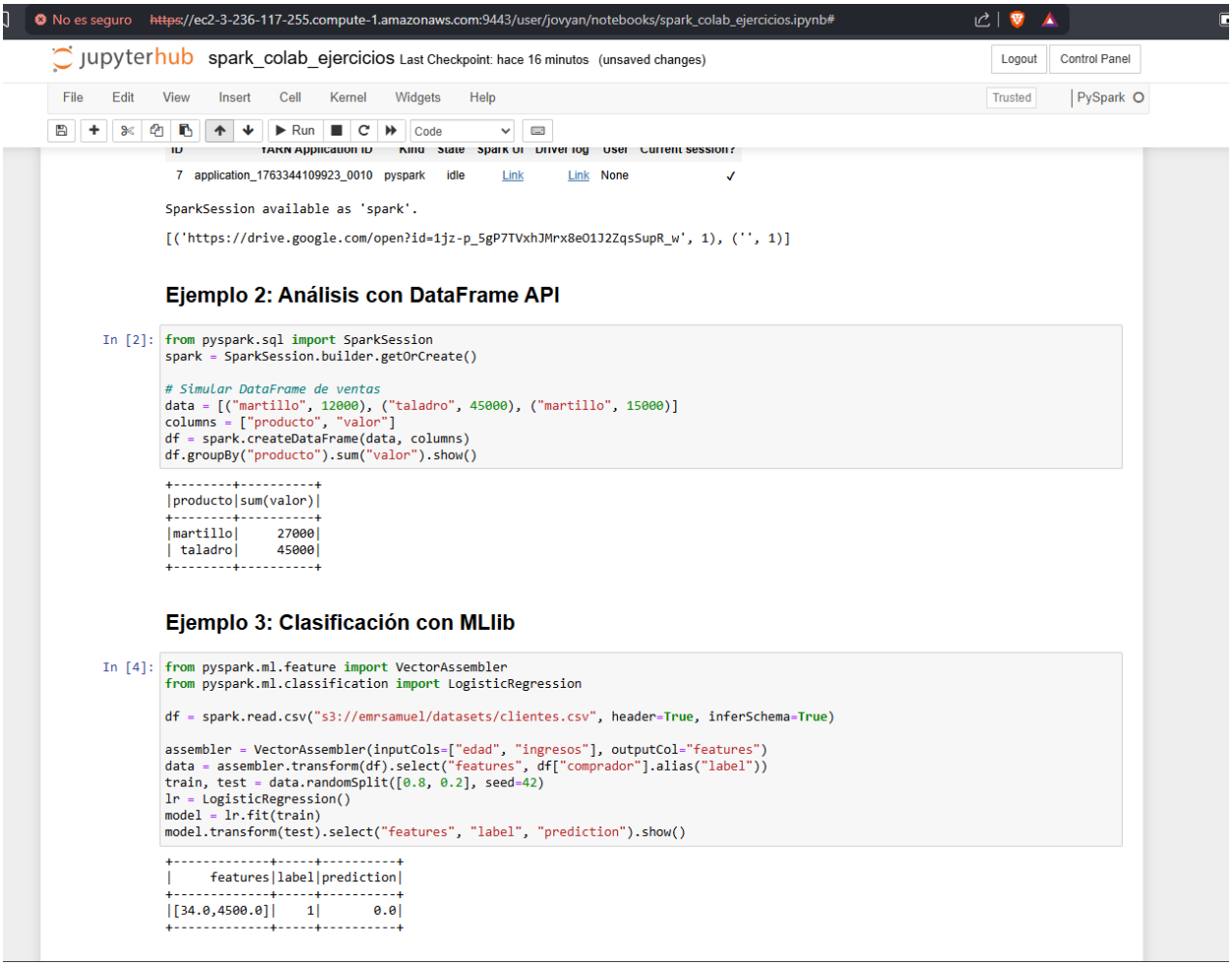
ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
7	application_1763344109923_0010	pyspark	idle	Link	Link	None	✓

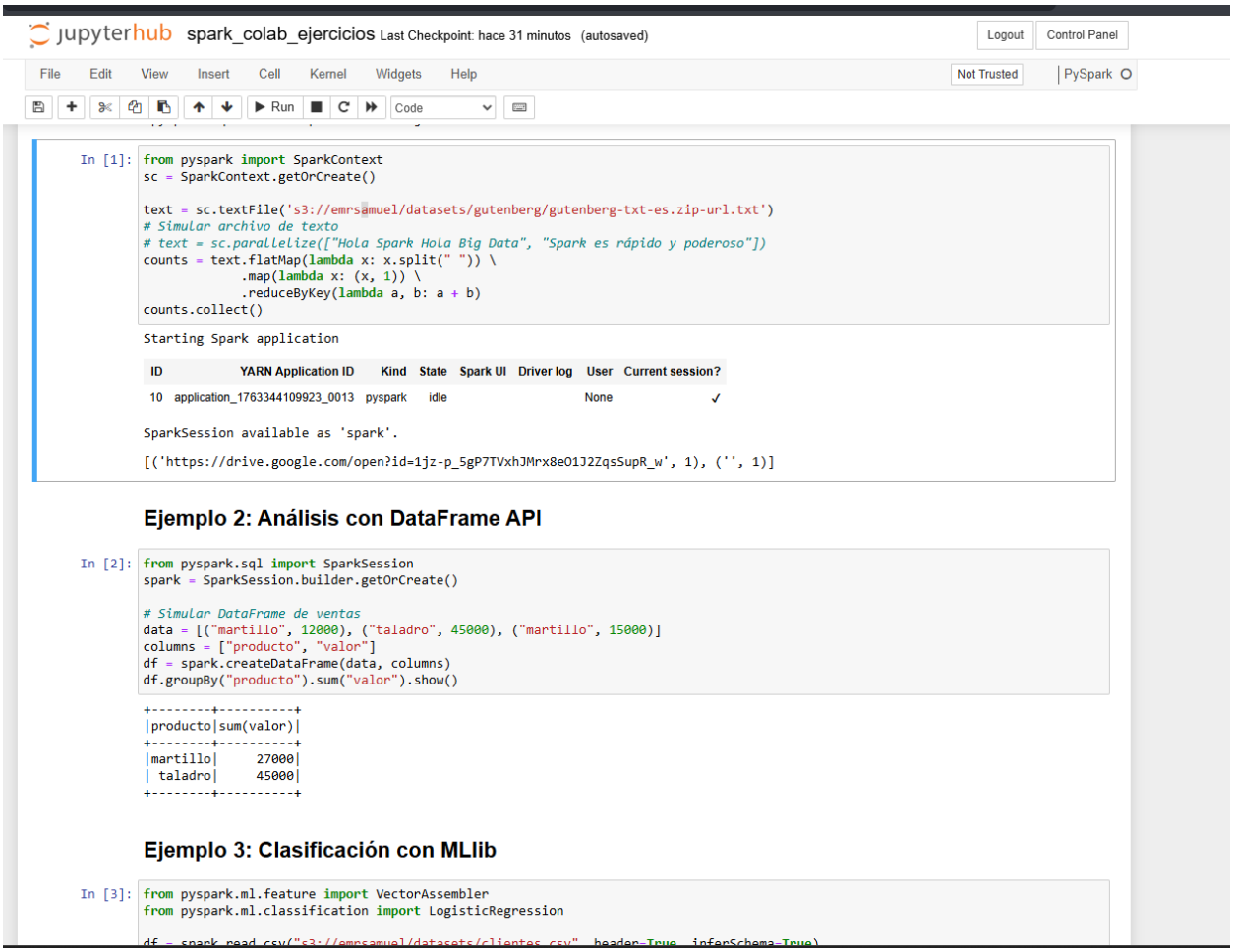
```
SparkSession available as 'spark'.

[('https://drive.google.com/open?id=1jz-p_5gP7TVxhJMrx8eO1J2ZqsSupR_w', 1), ('', 1)]
```

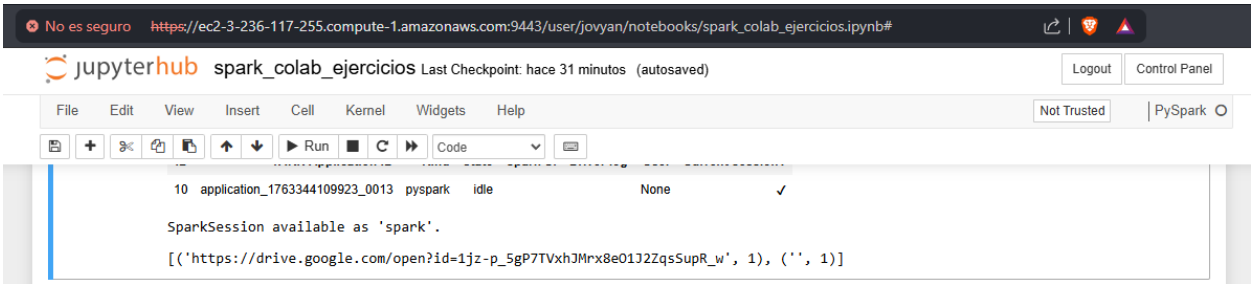
Ejemplo 2: Análisis con DataFrame API

In [2]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

# Simular DataFrame de ventas
data = [("martillo", 12000), ("taladro", 45000), ("martillo", 15000)]
columns = ["producto", "valor"]
df = spark.createDataFrame(data, columns)
df.groupBy("producto").sum("valor").show()
```

```
+--------+----------+
|producto|sum(valor)|
+--------+----------+
|martillo|     27000|
| taladro|     45000|
+--------+----------+
```


No es seguro https://ec2-3-236-117-255.compute-1.amazonaws.com:9443/user/jovyan/notebooks/spark_colab_ejercicios.ipynb#
jupyterhub spark_colab_ejercicios Last Checkpoint: hace 16 minutos (unsaved changes)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
PySpark
Run
Code
ID YARN Application ID Kind State Spark UI Driver log User Current session?
7 application_1763344109923_0010 pyspark idle Link Link None ✓
SparkSession available as 'spark'.
[('https://drive.google.com/open?id=1jz-p_5gP7TVxhJMrx8eO1J2ZqsSupR_w', 1), ('', 1)]
Ejemplo 2: Análisis con DataFrame API
In [2]:
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
# Simular DataFrame de ventas
data = [("martillo", 12000), ("taladro", 45000), ("martillo", 15000)]
columns = ["producto", "valor"]
df = spark.createDataFrame(data, columns)
df.groupBy("producto").sum("valor").show()
+--------+----------+
|producto|sum(valor)|
+--------+----------+
|martillo| 27000|
| taladro| 45000|
+--------+----------+
Ejemplo 3: Clasificación con MLlib
In [4]:
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LogisticRegression
df = spark.read.csv("s3://emrsamuel/datasets/clientes.csv", header=True, inferSchema=True)
assembler = VectorAssembler(inputCols=["edad", "ingresos"], outputCol="features")
data = assembler.transform(df).select("features", df["comprador"].alias("label"))
train, test = data.randomSplit([0.8, 0.2], seed=42)
lr = LogisticRegression()
model = lr.fit(train)
model.transform(test).select("features", "label", "prediction").show()
+-------------+-----+----------+
| features|label|prediction|
+-------------+-----+----------+
|[34.0,4500.0]| 1| 0.0|
+-------------+-----+----------+


jupyterhub spark_colab_ejercicios Last Checkpoint: hace 31 minutos (autosaved)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
PySpark
Run
Code
In [1]:
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
text = sc.textFile('s3://emrsamuel/datasets/gutenberg/gutenberg-txt-es.zip-url.txt')
# Simular archivo de texto
# text = sc.parallelize(["Hola Spark Hola Big Data", "Spark es rápido y poderoso"])
counts = text.flatMap(lambda x: x.split(" ")) \
.map(lambda x: (x, 1)) \
.reduceByKey(lambda a, b: a + b)
counts.collect()
Starting Spark application
ID YARN Application ID Kind State Spark UI Driver log User Current session?
10 application_1763344109923_0013 pyspark idle None ✓
SparkSession available as 'spark'.
[('https://drive.google.com/open?id=1jz-p_5gP7TVxhJMrx8eO1J2ZqsSupR_w', 1), ('', 1)]
Ejemplo 2: Análisis con DataFrame API
In [2]:
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
# Simular DataFrame de ventas
data = [("martillo", 12000), ("taladro", 45000), ("martillo", 15000)]
columns = ["producto", "valor"]
df = spark.createDataFrame(data, columns)
df.groupBy("producto").sum("valor").show()
+--------+----------+
|producto|sum(valor)|
+--------+----------+
|martillo| 27000|
| taladro| 45000|
+--------+----------+
Ejemplo 3: Clasificación con MLlib
In [3]:
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LogisticRegression



Ejemplo 2: Análisis con DataFrame API

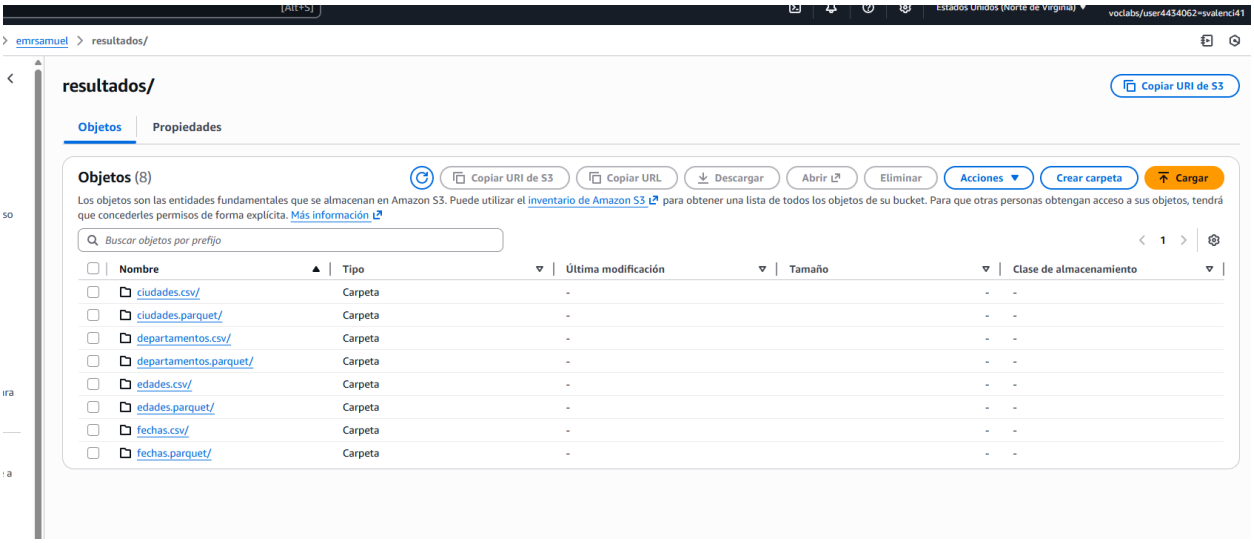
In [2]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

# Simular DataFrame de ventas
data = [("martillo", 12000), ("taladro", 45000), ("martillo", 15000)]
columns = ["producto", "valor"]
df = spark.createDataFrame(data, columns)
df.groupBy("producto").sum("valor").show()
```

```
+--------+----------+
|producto|sum(valor)|
+--------+----------+
|martillo|     27000|
| taladro|     45000|
+--------+----------+
```

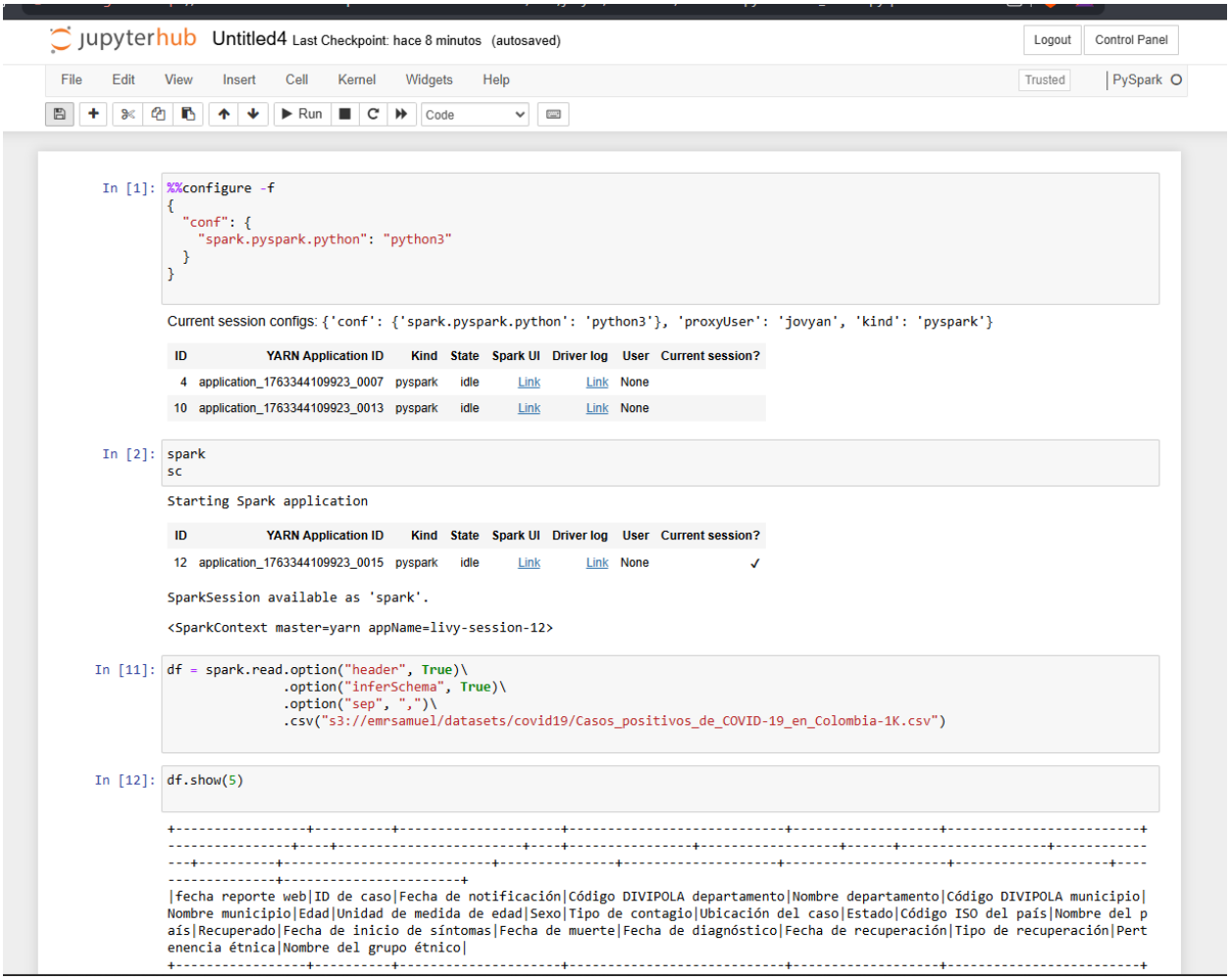
Ejemplo 3: Clasificación con MLlib

In [3]:
```
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LogisticRegression

df = spark.read.csv("s3://emrsamuel/datasets/clientes.csv", header=True, inferSchema=True)

assembler = VectorAssembler(inputCols=["edad", "ingresos"], outputCol="features")
data = assembler.transform(df).select("features", df["comprador"].alias("label"))
train, test = data.randomSplit([0.8, 0.2], seed=42)
lr = LogisticRegression()
model = lr.fit(train)
model.transform(test).select("features", "label", "prediction").show()
```

```
+-------------+-----+----------+
|     features|label|prediction|
+-------------+-----+----------+
|[34.0,4500.0]|    1|       0.0|
+-------------+-----+----------+
```



Ejercicio adicional propuesto en el Readme



jupyterhub Untitled4 Last Checkpoint: hace 8 minutos (autosaved)

Logout | Control Panel

File Edit View Insert Cell Kernel Widgets Help | Trusted | PySpark

Run | Code

In [1]:
```
%%configure -f
{
  "conf": {
    "spark.pyspark.python": "python3"
  }
}
```

Current session configs: {'conf': {'spark.pyspark.python': 'python3'}, 'proxyUser': 'jovyan', 'kind': 'pyspark'}

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
4	application_1763344109923_0007	pyspark	idle	Link	Link	None	
10	application_1763344109923_0013	pyspark	idle	Link	Link	None	

In [2]:
```
spark
sc
```

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
12	application_1763344109923_0015	pyspark	idle	Link	Link	None	✓

SparkSession available as 'spark'.

<SparkContext master=yarn appName=livy-session-12>

In [11]:
```
df = spark.read.option("header", True)\
               .option("inferSchema", True)\
               .option("sep", ",")\
               .csv("s3://emrsamuel/datasets/covid19/Casos_positivos_de_COVID-19_en_Colombia-1K.csv")
```

In [12]:
```
df.show(5)
```

```
+-----------------+----------+--------------------+--------------------------+------------------+------------------------+
----------------+----+----------------------+----+---------------+-----------------+------+------------------+-----------
---+----------+--------------------------+---------------+--------------------+--------------------+--------------------+----
-------------+-----------------------+
|fecha reporte web|ID de caso|Fecha de notificación|Código DIVIPOLA departamento|Nombre departamento|Código DIVIPOLA municipio|
Nombre municipio|Edad|Unidad de medida de edad|Sexo|Tipo de contagio|Ubicación del caso|Estado|Código ISO del país|Nombre del p
aís|Recuperado|Fecha de inicio de síntomas|Fecha de muerte|Fecha de diagnóstico|Fecha de recuperación|Tipo de recuperación|Pert
enencia étnica|Nombre del grupo étnico|
+-----------------+----------+--------------------+--------------------------+------------------+------------------------+
```

jupyterhub Untitled4 Last Checkpoint: hace 9 minutos (autosaved)

Logout | Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted | PySpark

Run | Code

```
----------------+----+------------------------+----+----------------+------------------+------+-------------------+------------
---+----------+-------------------------+---------------+--------------------+--------------------+--------------------+----
--------------+-----------------------+
only showing top 5 rows
```

In [13]:
```
df.printSchema()
```

```
root
 |-- fecha reporte web: string (nullable = true)
 |-- ID de caso: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- Nombre departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- Nombre municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Nombre del grupo étnico: string (nullable = true)
```

In [14]:
```
df = df.toDF(*[c.strip().lower().replace(" ", "_").replace("ó","o").replace("á","a").replace("í","i").replace("é","e").replace("
df.show(5)
```

```
+-----------------+----------+---------------------+---------------------------+------------------+-------------------------+
----------------+----+------------------------+----+----------------+-----------------+------+-------------------+------------
---+----------+---------------------------+---------------+--------------------+--------------------+--------------------+----
-------------+-----------------------+
|fecha_reporte_web|id_de_caso|fecha_de_notificación|código_divipola_departamento|nombre_departamento|código_divipola_municipio|
nombre_municipio|edad|unidad_de_medida_de_edad|sexo|tipo_de_contagio|ubicación_del_caso|estado|código_iso_del_país|nombre_del_p
aís|recuperado|fecha_de_inicio_de_síntomas|fecha_de_muerte|fecha_de_diagnóstico|fecha_de_recuperación|tipo_de_recuperación|pert
enencia_étnica|nombre_del_grupo_étnico|
+-----------------+----------+---------------------+---------------------------+------------------+-------------------------+
----------------+----+------------------------+----+----------------+-----------------+------+-------------------+------------
---+----------+---------------------------+---------------+--------------------+--------------------+--------------------+----
```

jupyterhub Untitled4 Last Checkpoint: hace 9 minutos (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

Run Code

```
df = df.withColumn(
    "edad_grupo",
    when(col("edad") < 18, "menor_18")
    .when((col("edad") >= 18) & (col("edad") <= 40), "18-40")
    .when((col("edad") > 40) & (col("edad") <= 60), "41-60")
    .otherwise("mayor_60")
)

df.select("edad", "edad_grupo").show(10)
```

```
+----+----------+
|edad|edad_grupo|
+----+----------+
|  19|     18-40|
|  34|     18-40|
|  50|     41-60|
|  55|     41-60|
|  25|     18-40|
|  27|     18-40|
|  85|  mayor_60|
|  22|     18-40|
|  28|     18-40|
|  36|     18-40|
+----+----------+
only showing top 10 rows
```

In [16]:

```
df_activos = df.filter(col("estado") == "Leve")  # o el valor que indique activo
df_activos.show(5)
```

```
+----------------+---------+--------------------+---------------------------+-------------------+-------------------------+
---------------+----+----------------------+----+----------------+-----------------+------+-------------------+-----------
---+----------+---------------------+-------------+-------------------+---------------------+--------------------+----
-------------+----------------------+----------+
|fecha_reporte_web|id_de_caso|fecha_de_notificación|código_divipola_departamento|nombre_departamento|código_divipola_municipio|
nombre_municipio|edad|unidad_de_medida_de_edad|sexo|tipo_de_contagio|ubicación_del_caso|estado|código_iso_del_país|nombre_del_p
aís|recuperado|fecha_de_inicio_de_síntomas|fecha_de_muerte|fecha_de_diagnóstico|fecha_de_recuperación|tipo_de_recuperación|pert
enencia_étnica|nombre_del_grupo_étnico|edad_grupo|
+----------------+---------+--------------------+---------------------------+-------------------+-------------------------+
---------------+----+----------------------+----+----------------+-----------------+------+-------------------+-----------
---+----------+---------------------+-------------+-------------------+---------------------+--------------------+----
-------------+----------------------+----------+
| 6/3/2020 0:00:00|         1|     2/3/2020 0:00:00|                         11|             BOGOTA|                    11001|
BOGOTA|  19|                     1|   F|       Importado|             Casa|  Leve|                380|          ITALIA|Recupe
rado|          27/2/2020 0:00:00|           NULL|    6/3/2020 0:00:00|    13/3/2020 0:00:00|                 PCR|
6|                 NULL|     18-40|
| 9/3/2020 0:00:00|         2|     6/3/2020 0:00:00|                         76|              VALLE|                    76111|
```

jupyterhub Untitled4 Last Checkpoint: hace 9 minutos (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

Run Code

In [17]:
```
df_departamento = df.groupBy("nombre_departamento")\
                    .count()\
                    .orderBy("count", ascending=False)
df_departamento.show(10)
```

```
+------------------+-----+
|nombre_departamento|count|
+------------------+-----+
|            BOGOTA|  401|
|             VALLE|  148|
|          ANTIOQUIA|  106|
|       CUNDINAMARCA|   49|
|          CARTAGENA|   39|
|          RISARALDA|   34|
|       BARRANQUILLA|   31|
|             HUILA|   30|
|            QUINDIO|   23|
|    NORTE SANTANDER|   19|
+------------------+-----+
only showing top 10 rows
```

In [18]:
```
df_ciudad = df.groupBy("nombre_municipio")\
              .count()\
              .orderBy("count", ascending=False)
df_ciudad.show(10)
```

```
+---------------+-----+
|nombre_municipio|count|
+---------------+-----+
|          BOGOTA|  401|
|           CALI|  101|
|       MEDELLIN|   63|
|       CARTAGENA|   39|
|    BARRANQUILLA|   31|
|           NEIVA|   27|
|         PEREIRA|   25|
|         PALMIRA|   22|
|      VALLEDUPAR|   16|
|          CUCUTA|   15|
+---------------+-----+
only showing top 10 rows
```

In [19]:
```
df_fecha = df.groupBy("fecha_reporte_web")\
             .count()\
             .orderBy("count", ascending=False)
df_fecha.show(10)
```

jupyterhub Untitled4 Last Checkpoint: hace 9 minutos (autosaved)

Logout | Control Pan

File Edit View Insert Cell Kernel Widgets Help

Trusted | PySpark

Run | Code

In [20]:
```
df_edad = df.groupBy("edad_grupo")\
            .count()\
            .orderBy("edad_grupo")
df_edad.show()
```

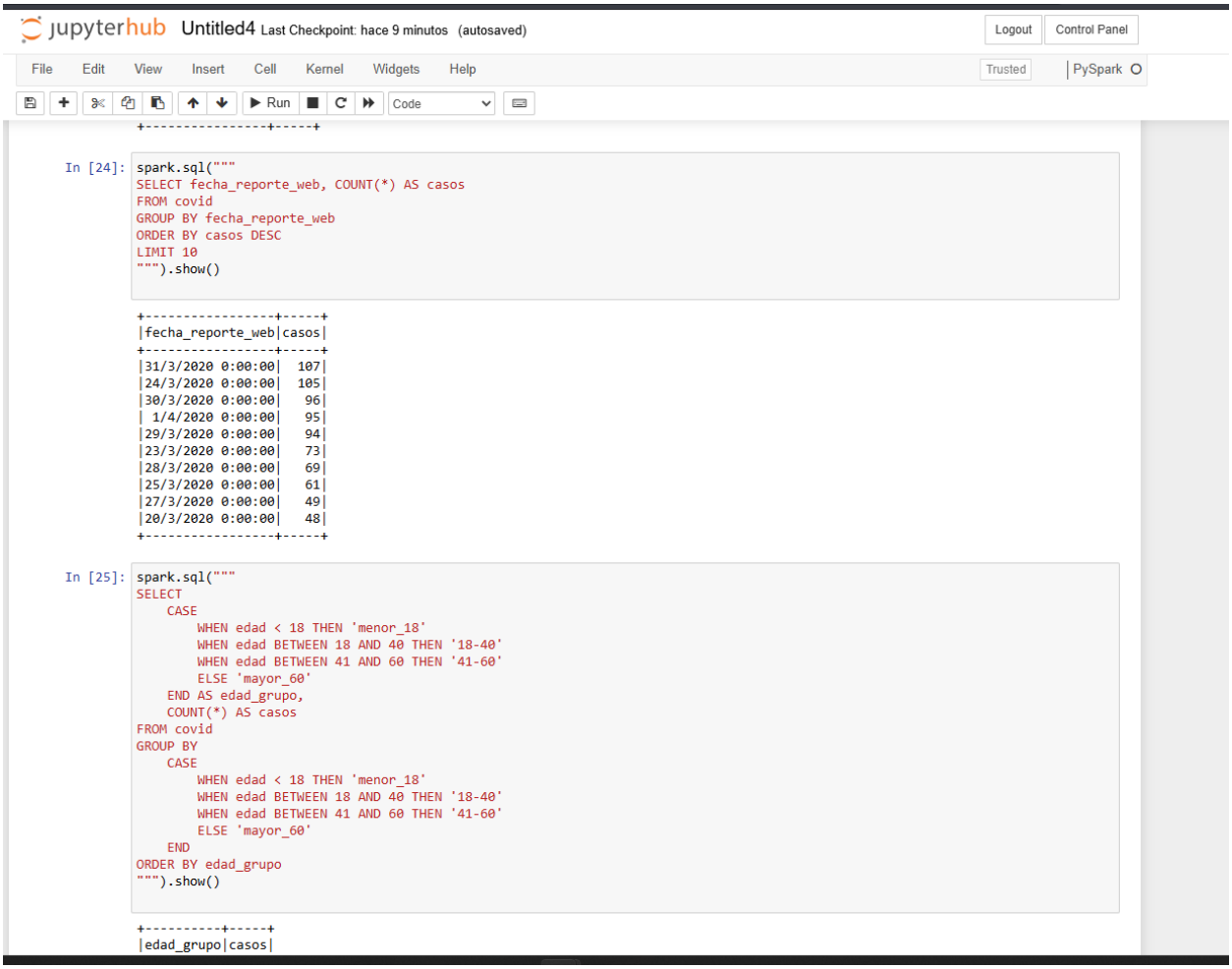
```
+----------+-----+
|edad_grupo|count|
+----------+-----+
|     18-40|  476|
|     41-60|  332|
|  mayor_60|  159|
|  menor_18|   33|
+----------+-----+
```

In [21]:
```
df.createOrReplaceTempView("covid")
```

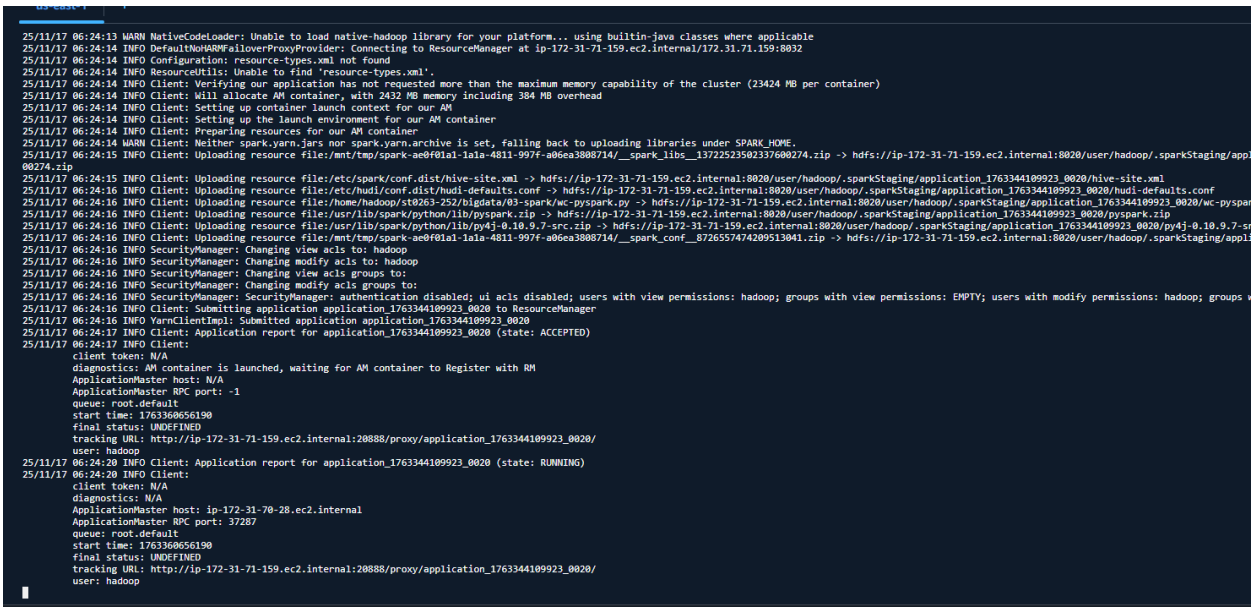
In [22]:
```
spark.sql("""
SELECT nombre_departamento, COUNT(*) AS casos
FROM covid
GROUP BY nombre_departamento
ORDER BY casos DESC
LIMIT 10
""").show()
```

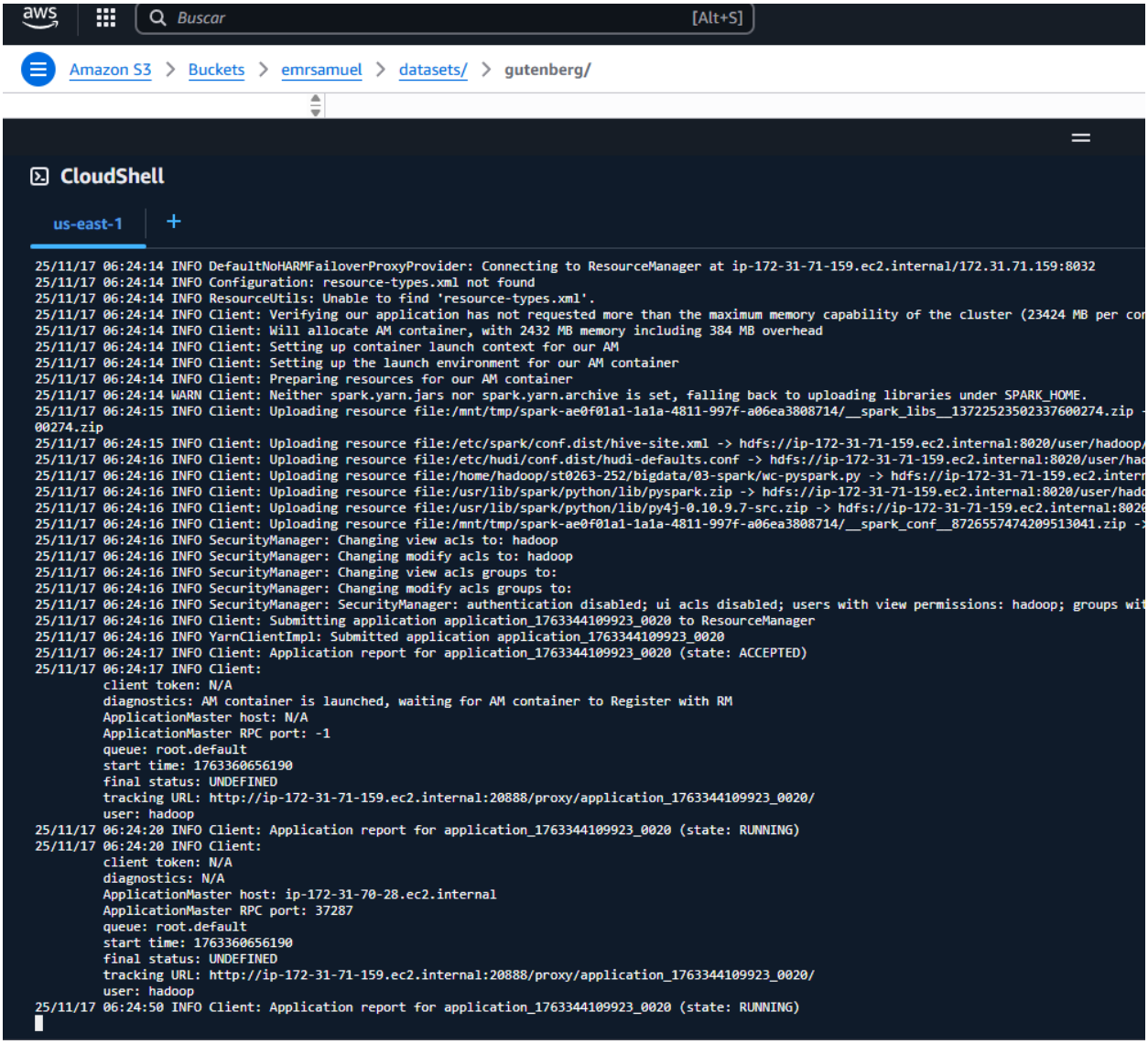
```
+------------------+-----+
|nombre_departamento|casos|
+------------------+-----+
|            BOGOTA|  401|
|             VALLE|  148|
|          ANTIOQUIA|  106|
|       CUNDINAMARCA|   49|
|          CARTAGENA|   39|
|          RISARALDA|   34|
|       BARRANQUILLA|   31|
|              HUILA|   30|
|            QUINDIO|   23|
|    NORTE SANTANDER|   19|
+------------------+-----+
```

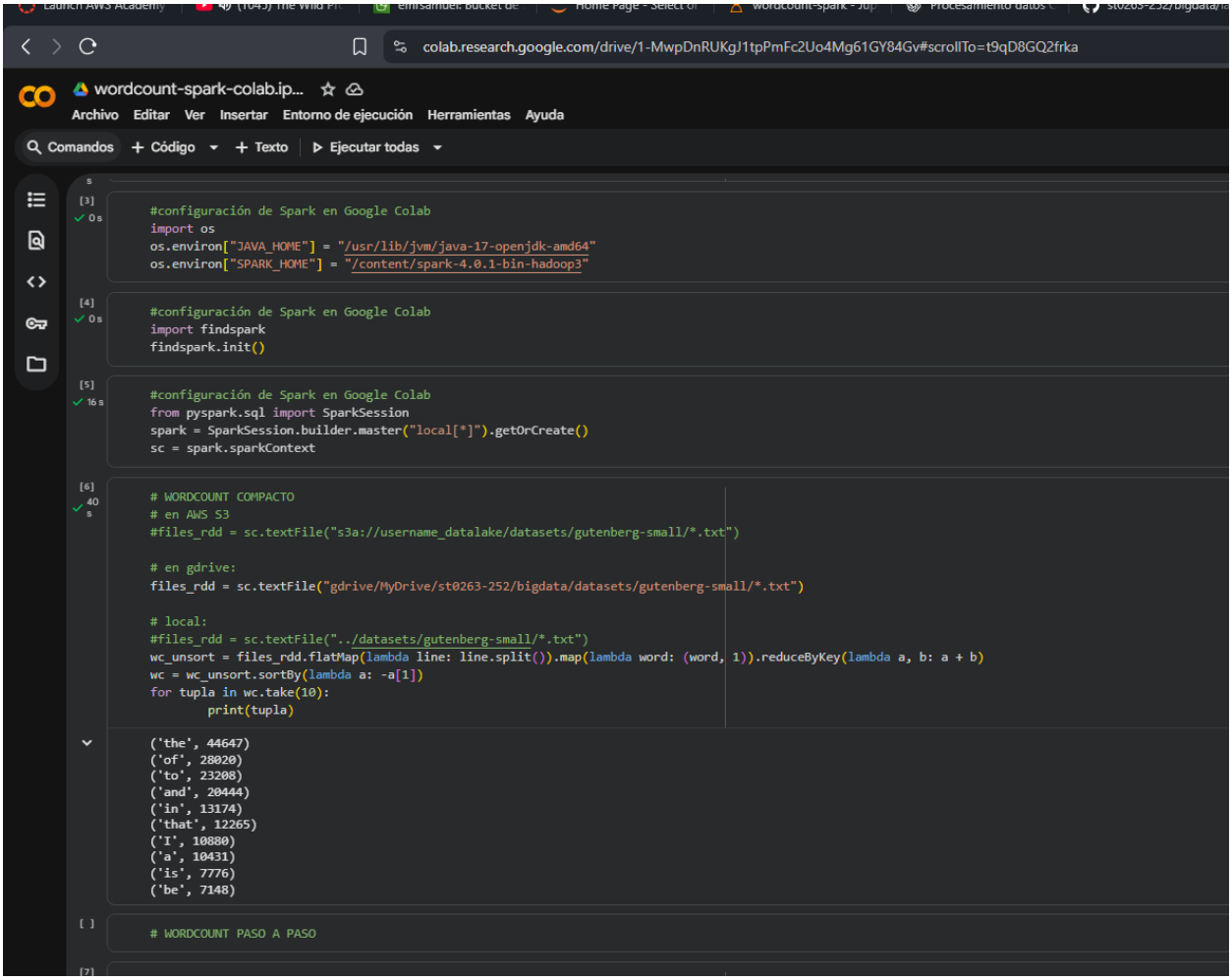
In [23]:
```
spark.sql("""
SELECT nombre_municipio, COUNT(*) AS casos
FROM covid
GROUP BY nombre_municipio
```

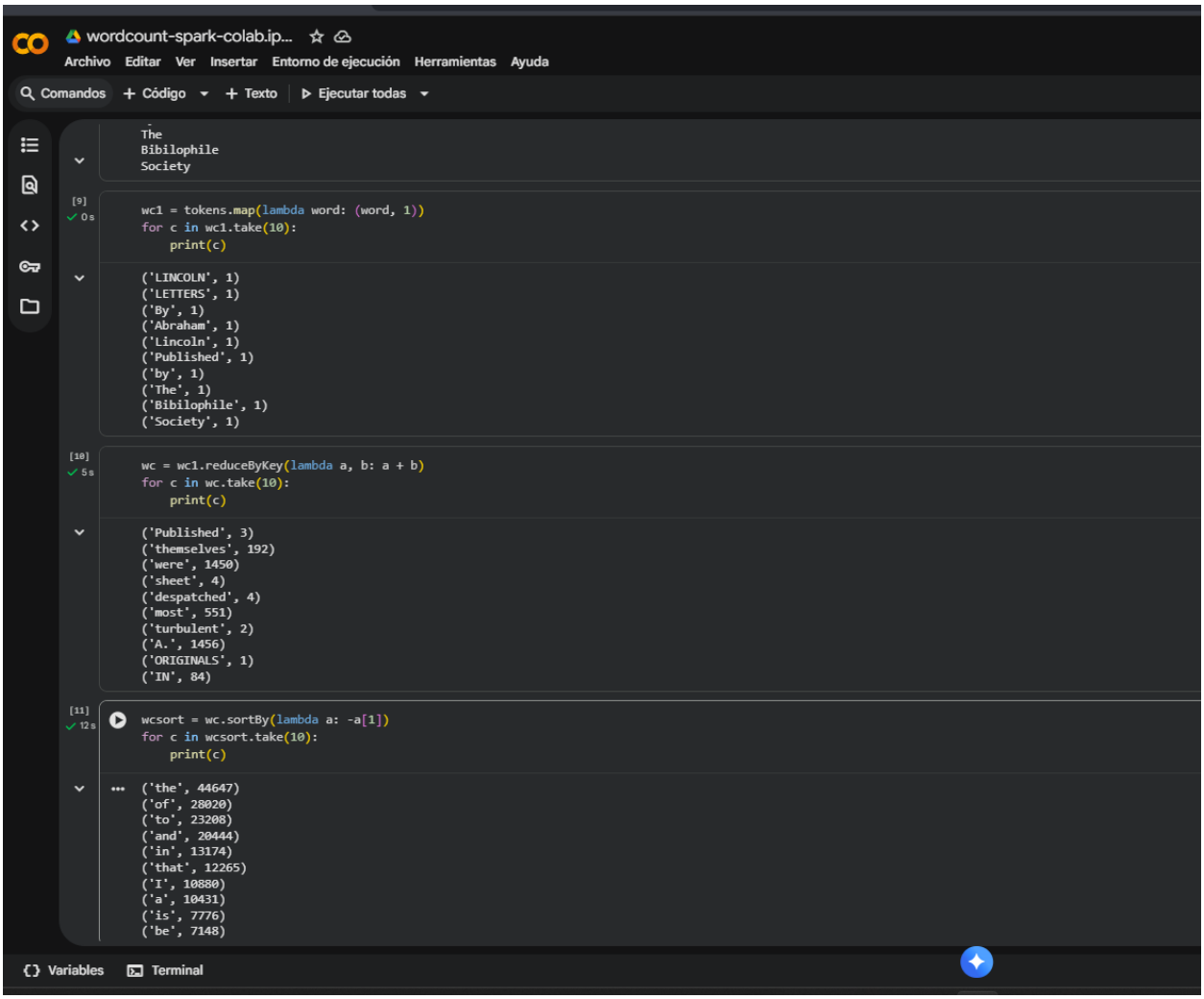


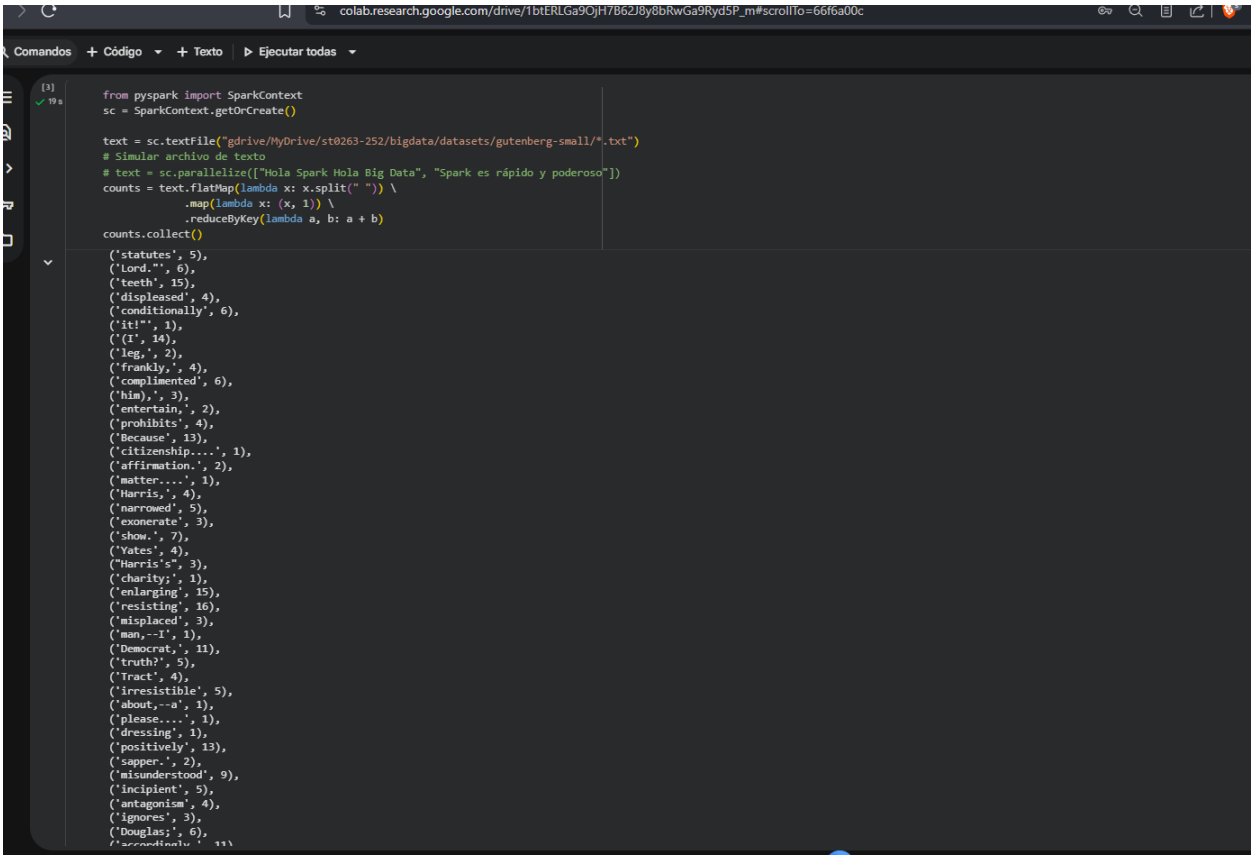
Ejecución de .py en main node

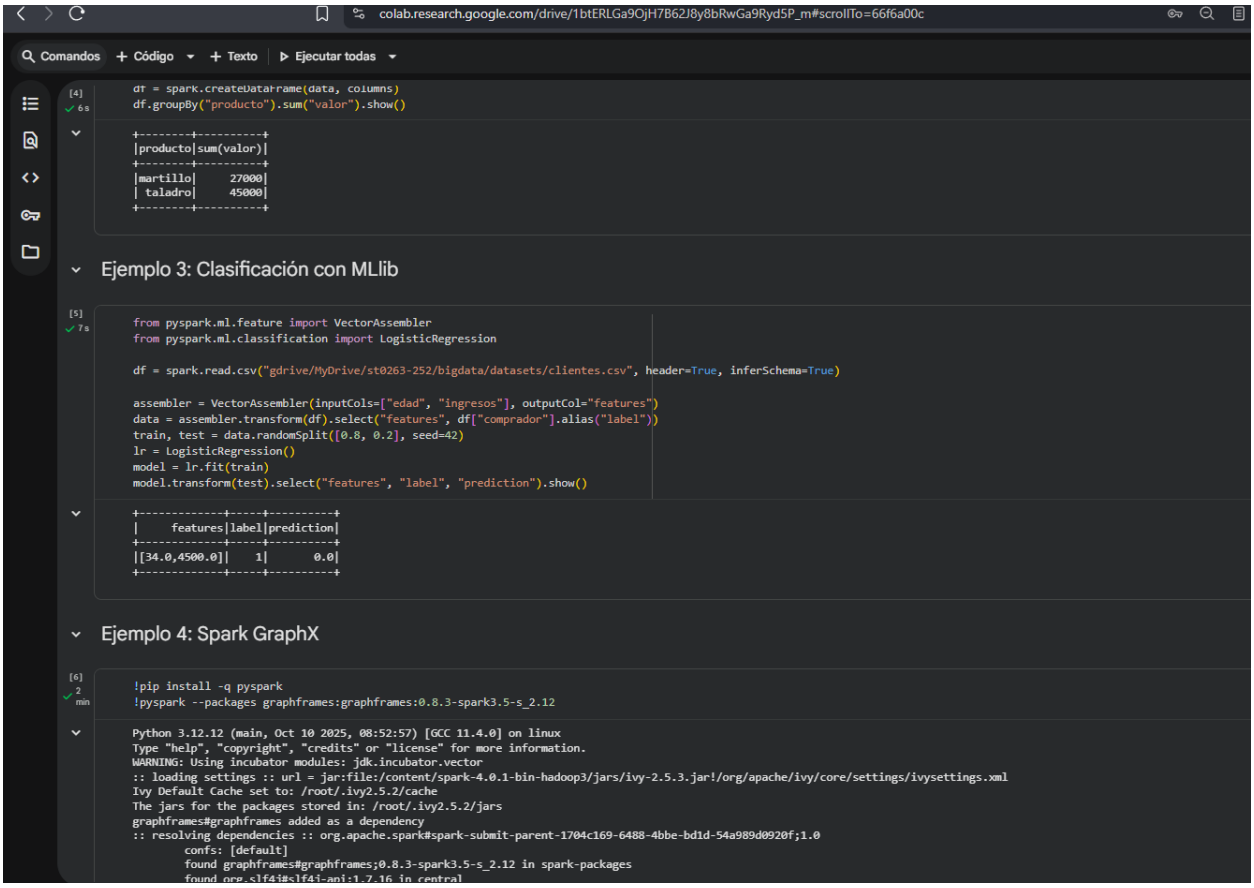



aws
Buscar
[Alt+S]
Amazon S3 > Buckets > emrsamuel > datasets/ > gutenberg/
CloudShell
us-east-1
25/11/17 06:24:14 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at ip-172-31-71-159.ec2.internal/172.31.71.159:8032
25/11/17 06:24:14 INFO Configuration: resource-types.xml not found
25/11/17 06:24:14 INFO ResourceUtils: Unable to find 'resource-types.xml'.
25/11/17 06:24:14 INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (23424 MB per co
25/11/17 06:24:14 INFO Client: Will allocate AM container, with 2432 MB memory including 384 MB overhead
25/11/17 06:24:14 INFO Client: Setting up container launch context for our AM
25/11/17 06:24:14 INFO Client: Setting up the launch environment for our AM container
25/11/17 06:24:14 INFO Client: Preparing resources for our AM container
25/11/17 06:24:14 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
25/11/17 06:24:15 INFO Client: Uploading resource file:/mnt/tmp/spark-ae0f01a1-1a1a-4811-997f-a06ea3808714/__spark_libs__1372523502337600274.zip
00274.zip
25/11/17 06:24:15 INFO Client: Uploading resource file:/etc/spark/conf.dist/hive-site.xml -> hdfs://ip-172-31-71-159.ec2.internal:8020/user/hadoop/
25/11/17 06:24:16 INFO Client: Uploading resource file:/etc/hudi/conf.dist/hudi-defaults.conf -> hdfs://ip-172-31-71-159.ec2.internal:8020/user/had
25/11/17 06:24:16 INFO Client: Uploading resource file:/home/hadoop/st0263-252/bigdata/03-spark/wc-pyspark.py -> hdfs://ip-172-31-71-159.ec2.intern
25/11/17 06:24:16 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/pyspark.zip -> hdfs://ip-172-31-71-159.ec2.internal:8020/user/hadd
25/11/17 06:24:16 INFO Client: Uploading resource file:/usr/lib/spark/python/lib/py4j-0.10.9.7-src.zip -> hdfs://ip-172-31-71-159.ec2.internal:8020
25/11/17 06:24:16 INFO Client: Uploading resource file:/mnt/tmp/spark-ae0f01a1-1a1a-4811-997f-a06ea3808714/__spark_conf__872655747420951304 1.zip -
25/11/17 06:24:16 INFO SecurityManager: Changing view acls to: hadoop
25/11/17 06:24:16 INFO SecurityManager: Changing modify acls to: hadoop
25/11/17 06:24:16 INFO SecurityManager: Changing view acls groups to:
25/11/17 06:24:16 INFO SecurityManager: Changing modify acls groups to:
25/11/17 06:24:16 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: hadoop; groups wit
25/11/17 06:24:16 INFO Client: Submitting application application_1763344109923_0020 to ResourceManager
25/11/17 06:24:16 INFO YarnClientImpl: Submitted application application_1763344109923_0020
25/11/17 06:24:17 INFO Client: Application report for application_1763344109923_0020 (state: ACCEPTED)
25/11/17 06:24:17 INFO Client:
client token: N/A
diagnostics: AM container is launched, waiting for AM container to Register with RM
ApplicationMaster host: N/A
ApplicationMaster RPC port: -1
queue: root.default
start time: 1763360656190
final status: UNDEFINED
tracking URL: http://ip-172-31-71-159.ec2.internal:20888/proxy/application_1763344109923_0020/
user: hadoop
25/11/17 06:24:20 INFO Client: Application report for application_1763344109923_0020 (state: RUNNING)
25/11/17 06:24:20 INFO Client:
client token: N/A
diagnostics: N/A
ApplicationMaster host: ip-172-31-70-28.ec2.internal
ApplicationMaster RPC port: 37287
queue: root.default
start time: 1763360656190
final status: UNDEFINED
tracking URL: http://ip-172-31-71-159.ec2.internal:20888/proxy/application_1763344109923_0020/
user: hadoop
25/11/17 06:24:50 INFO Client: Application report for application_1763344109923_0020 (state: RUNNING)


colab.research.google.com/drive/1-MwpDnRUKgJ1tpPmFc2Uo4Mg61GY84Gv#scrollTo=t9qD8GQ2frka
wordcount-spark-colab.ip...
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Comandos + Código + Texto Ejecutar todas
[3] 0 s
#configuración de Spark en Google Colab
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-17-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-4.0.1-bin-hadoop3"
[4] 0 s
#configuración de Spark en Google Colab
import findspark
findspark.init()
[5] 16 s
#configuración de Spark en Google Colab
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
[6] 40 s
WORDCOUNT COMPACTO
en AWS S3
#files_rdd = sc.textFile("s3a://username_datalake/datasets/gutenberg-small/*.txt")
# en gdrive:
files_rdd = sc.textFile("gdrive/MyDrive/st0263-252/bigdata/datasets/gutenberg-small/*.txt")
# local:
#files_rdd = sc.textFile("../datasets/gutenberg-small/*.txt")
wc_unsort = files_rdd.flatMap(lambda line: line.split()).map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
wc = wc_unsort.sortBy(lambda a: -a[1])
for tupla in wc.take(10):
print(tupla)
('the', 44647)
('of', 28020)
('to', 23208)
('and', 20444)
('in', 13174)
('that', 12265)
('I', 10880)
('a', 10431)
('is', 7776)
('be', 7148)
[ ]
WORDCOUNT PASO A PASO
[7]


wordcount-spark-colab.ip...
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Comandos + Código + Texto Ejecutar todas
The
Bibilophile
Society
[9] 0 s
wc1 = tokens.map(lambda word: (word, 1))
for c in wc1.take(10):
print(c)
('LINCOLN', 1)
('LETTERS', 1)
('By', 1)
('Abraham', 1)
('Lincoln', 1)
('Published', 1)
('by', 1)
('The', 1)
('Bibilophile', 1)
('Society', 1)
[10] 5 s
wc = wc1.reduceByKey(lambda a, b: a + b)
for c in wc.take(10):
print(c)
('Published', 3)
('themselves', 192)
('were', 1450)
('sheet', 4)
('despatched', 4)
('most', 551)
('turbulent', 2)
('A.', 1456)
('ORIGINALS', 1)
('IN', 84)
[11] 12 s
wcsort = wc.sortBy(lambda a: -a[1])
for c in wcsort.take(10):
print(c)
('the', 44647)
('of', 28020)
('to', 23208)
('and', 20444)
('in', 13174)
('that', 12265)
('I', 10880)
('a', 10431)
('is', 7776)
('be', 7148)
Variables Terminal


colab.research.google.com/drive/1btERLGa9OjH7B62J8y8bRwGa9Ryd5P_m#scrollTo=66f6a00c
Comandos
+ Código
+ Texto
Ejecutar todas
[3]
19 s
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
text = sc.textFile("gdrive/MyDrive/st0263-252/bigdata/datasets/gutenberg-small/*.txt")
# Simular archivo de texto
# text = sc.parallelize(["Hola Spark Hola Big Data", "Spark es rápido y poderoso"])
counts = text.flatMap(lambda x: x.split(" ")) \
.map(lambda x: (x, 1)) \
.reduceByKey(lambda a, b: a + b)
counts.collect()
('statutes', 5),
('Lord."', 6),
('teeth', 15),
('displeased', 4),
('conditionally', 6),
('it!"', 1),
('(I', 14),
('leg,', 2),
('frankly,', 4),
('complimented', 6),
('him),', 3),
('entertain,', 2),
('prohibits', 4),
('Because', 13),
('citizenship....', 1),
('affirmation.', 2),
('matter....', 1),
('Harris,', 4),
('narrowed', 5),
('exonerate', 3),
('show.', 7),
('Yates', 4),
("Harris's", 3),
('charity;', 1),
('enlarging', 15),
('resisting', 16),
('misplaced', 3),
('man,--I', 1),
('Democrat,', 11),
('truth?', 5),
('Tract', 4),
('irresistible', 5),
('about,--a', 1),
('please....', 1),
('dressing', 1),
('positively', 13),
('sapper.', 2),
('misunderstood', 9),
('incipient', 5),
('antagonism', 4),
('ignores', 3),
('Douglas;', 6),


colab.research.google.com/drive/1btERLGa9OjH7B62J8y8bRwGa9Ryd5P_m#scrollTo=66f6a00c
Comandos
+ Código
+ Texto
Ejecutar todas
[4]
6 s
df = spark.createDataFrame(data, columns)
df.groupBy("producto").sum("valor").show()
+--------+----------+
|producto|sum(valor)|
+--------+----------+
|martillo| 27000|
| taladro| 45000|
+--------+----------+
Ejemplo 3: Clasificación con MLlib
[5]
7 s
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.classification import LogisticRegression
df = spark.read.csv("gdrive/MyDrive/st0263-252/bigdata/datasets/clientes.csv", header=True, inferSchema=True)
assembler = VectorAssembler(inputCols=["edad", "ingresos"], outputCol="features")
data = assembler.transform(df).select("features", df["comprador"].alias("label"))
train, test = data.randomSplit([0.8, 0.2], seed=42)
lr = LogisticRegression()
model = lr.fit(train)
model.transform(test).select("features", "label", "prediction").show()
+-------------+-----+----------+
| features|label|prediction|
+-------------+-----+----------+
|[34.0,4500.0]| 1| 0.0|
+-------------+-----+----------+
Ejemplo 4: Spark GraphX
[6]
2 min
!pip install -q pyspark
!pyspark --packages graphframes:graphframes:0.8.3-spark3.5-s_2.12
Python 3.12.12 (main, Oct 10 2025, 08:52:57) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
WARNING: Using incubator modules: jdk.incubator.vector
:: loading settings :: url = jar:file:/content/spark-4.0.1-bin-hadoop3/jars/ivy-2.5.3.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /root/.ivy2.5.2/cache
The jars for the packages stored in: /root/.ivy2.5.2/jars
graphframes#graphframes added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-1704c169-6488-4bbe-bd1d-54a989d0920f;1.0
confs: [default]
found graphframes#graphframes;0.8.3-spark3.5-s_2.12 in spark-packages
found org.slf4j#slf4j-api;1.7.16 in central

jupyterhub wordcount-spark (unsaved changes)

Logout | Control Panel

File Edit View Insert Cell Kernel Widgets Help

Trusted | PySpark

Run | Code

```
<pyspark.sql.session.SparkSession object at 0xffff89dee730>
```

In [2]:
```
# si esta en EMR o Databricks, estos objetos ya están preconstruidos:
sc
```

```
<SparkContext master=yarn appName=livy-session-14>
```

In [*]:
```
# WORDCOUNT COMPACTO
# en AWS S3
files_rdd = sc.textFile('s3a://emrsamuel/datasets/gutenberg-small/AbrahamLincoln___LincolnLetters.txt')

# en gdrive:
#files_rdd = sc.textFile("gdrive/MyDrive/st0263-252/bigdata/datasets/gutenberg-small/*.txt")

# Local:
#files_rdd = sc.textFile("../datasets/gutenberg-small/*.txt")
wc_unsort = files_rdd.flatMap(lambda line: line.split()).map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
wc = wc_unsort.sortBy(lambda a: -a[1])
for tupla in wc.take(10):
        print(tupla)
```

Progress:

In [ ]:
```
# WORDCOUNT PASO A PASO
```

In [*]:
```
files = sc.textFile("gdrive/MyDrive/st0263-252/bigdata/datasets/gutenberg-small/*.txt")
for f in files.take(10):
    print(f)
```

In [*]:
```
tokens = files.flatMap(lambda line: line.split())
for t in tokens.take(10):
    print(t)
```

In [*]:
```
wc1 = tokens.map(lambda word: (word, 1))
for c in wc1.take(10):
    print(c)
```

In [*]:
```
wc = wc1.reduceByKey(lambda a, b: a + b)
for c in wc.take(10):
    print(c)
```

In [*]:
```
wcsort = wc.sortBy(lambda a: -a[1])
for c in wcsort.take(10):
    print(c)
```