**Google Play Store Analysis**

**Project By: Krunali Patel**

**Step 1: Start pyspark**

krunali@ubuntu:~$ pyspark

**Step 2: Import Library**

>>> import pyspark

>>> from pyspark.sql import SparkSession

>>> from pyspark.sql.types import StructType,StructField,StringType,IntegerType

>>> from pyspark.sql.functions import \*

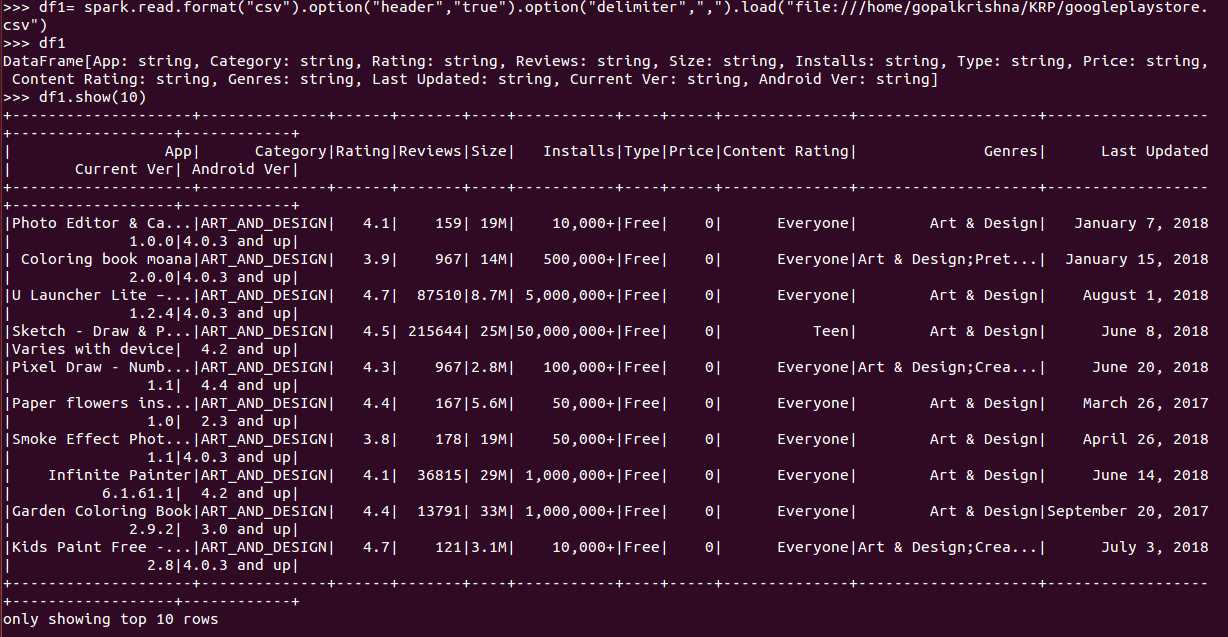
**Step 3: Create Dataframe**

>>> df1= spark.read.format("csv").option("header","true").option("delimiter",",").load("file:///home/krunali/KRP/googleplaystore.csv")

>>> df1

DataFrame[App: string, Category: string, Rating: string, Reviews: string, Size: string, Installs: string, Type: string, Price: string, Content Rating: string, Genres: string, Last Updated: string, Current Ver: string, Android Ver: string]

>>> df1.show(10)



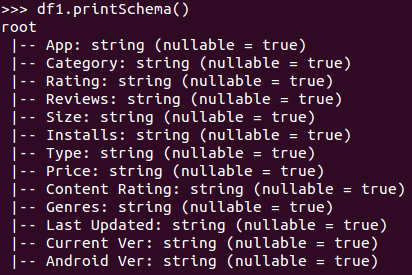
>>> df1.count()

10841



**Step 4: Check Schema**

>>> df1.printSchema()



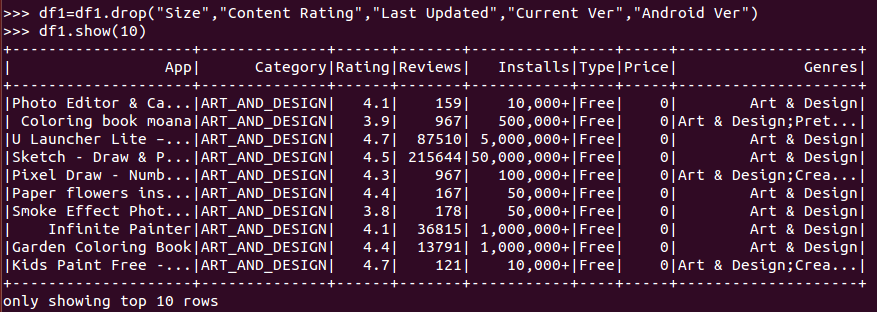
**Remark: - If we see above schema, all columns’ datatype are string type because( e.g. here some NULL values in Price and in Install also + symbol is there ) like that.**

**Step 5: Data Cleaning**

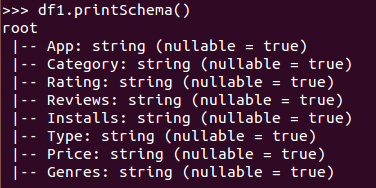
**Remark:- Here we will remove the columns which are not important.**

>>> df1=df1.drop("Size","Content Rating","Last Updated","Current Ver","Android Ver")

>>> df1.show(10)



>>> df1.printSchema()  **====> (Here we can see that the data types are corrected)**



**Step 6: Import library regexp\_replace to replace string data type by Integer data Type & correct the data types of columns**

from pyspark.sql.functions import regexp\_replace, col

df1=df1.withColumn("Reviews",col("Reviews").cast(IntegerType()))\

.withColumn("Installs",regexp\_replace(col("Installs"),"[^0-9]",""))\

.withColumn("Installs",col("Installs").cast(IntegerType()))\

.withColumn("Price",regexp\_replace(col("Price"),"[$]",""))\

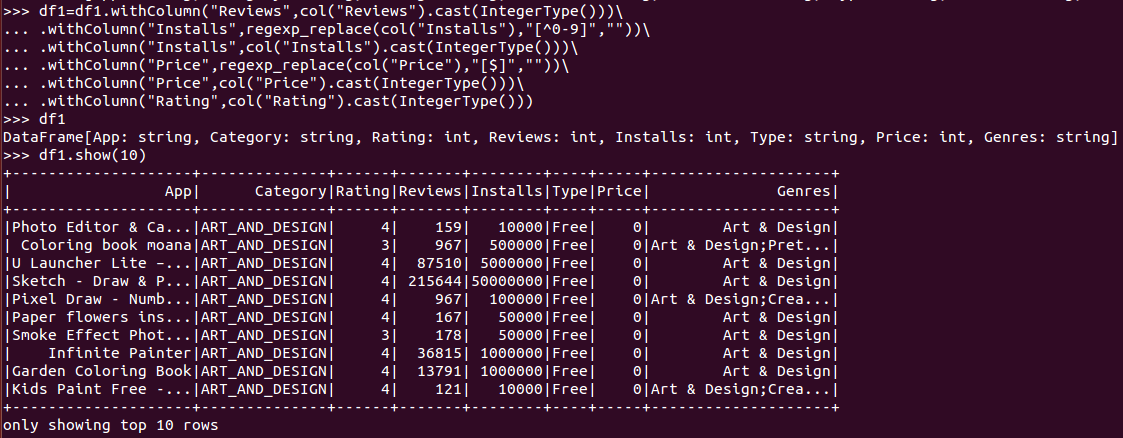
.withColumn("Price",col("Price").cast(IntegerType()))\

.withColumn("Rating",col("Rating").cast(IntegerType()))

>>> df1

DataFrame[App: string, Category: string, Rating: int, Reviews: int, Installs: int, Type: string, Price: int, Genres: string]

>>> df1.show(10)

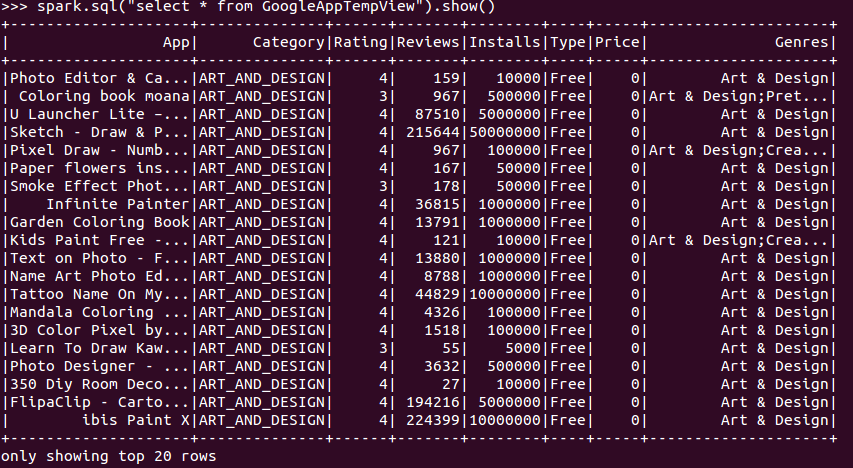


**Step 7: Create a temporary View to do analysis on that.**

>>> df1.createOrReplaceTempView("GoogleAppTempView")

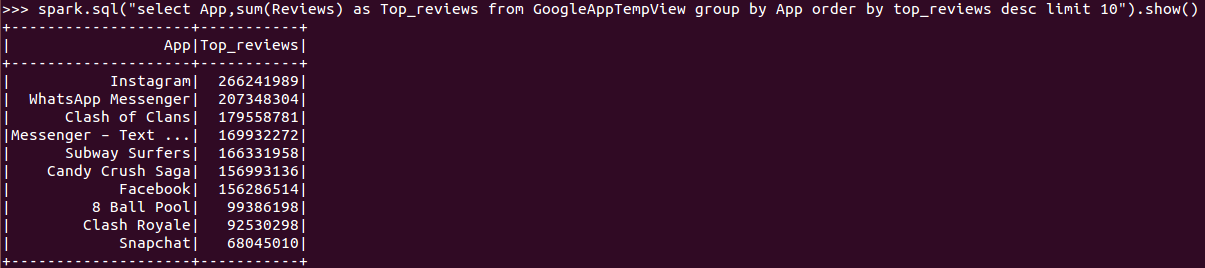


>>> spark.sql("select \* from GoogleAppTempView ").show()



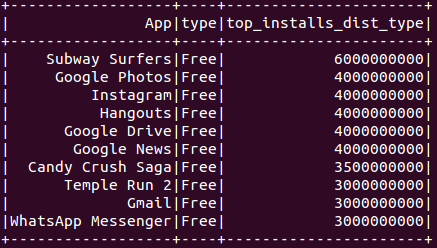
**Q.1 Find out the Top 10 review given to the app**

>>> spark.sql("select App,sum(Reviews) as Top\_reviews from GoogleAppTempView group by App order by top\_reviews desc limit 10").show()



**Q.2 Find out the Top 10 installs apps & distribution of type (free / paid)**

>>> spark.sql("select App,type,sum(Installs) as top\_installs\_dist\_type from GoogleAppTempView group by 1,2 order by 3 desc limit 10").show()

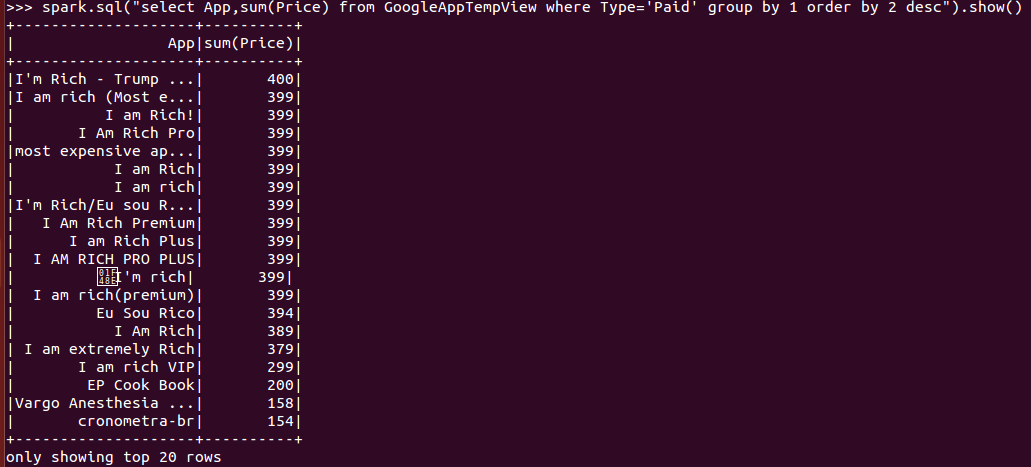


**Q.3 Find out category wise distribution of installed apps**

>>> spark.sql("select category,sum(Installs) as categorywise\_installs from GoogleAppTempView group by category order by categorywise\_installs desc limit 10”).show()



**Q.4 Find out the Top paid apps**

>>> spark.sql("select App,sum(Price) from GoogleAppTempView where Type='Paid' group by 1 order by 2 desc").show()

**Q.5 Find out the Top paid rating apps**

>>> spark.sql("select App,sum(Rating) from GoogleAppTempView where Type='Paid' group by 1 order by 2 desc").show()

