**Extracting data from charts.csv**

>>> **df = spark.read.csv('charts.csv',header=True,inferSchema=True)**

(to read the csv file )

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

>>> **newdf = df.select('title','rank',year(df.date).alias(‘date’),'artist','chart','trend','streams')**

(convert date into year and storing ina datafram)

>>> **newdf.show()**

>>> **df1 = newdf.filter(newdf['date']=='2020')**

(this is used to filter the data having date 2020)

>>> **df1.show()**

>>> **df2 = newdf.filter(newdf['date']=='2021')**

>>> **df2.show()**

(then we convert toPandas and )

**>>>df1.toPandas()**

**>>>df2.toPandas()**

**Import pandas as pd**

**Both = pd.concat([df1,df2],axis=1)**

**Both.to\_csv(‘both.csv’)**

**DATA CLEANING**

**Import pyspark.pandas as ps** (import pandas api)

>>> **df = ps.read\_csv('both.csv')** #reading data

>>> **df.head()**

>>> **df = df.drop(['url','Unnamed: 0','\_c0'],axis=1)** #dropping useless columns

**>>> type(df)** #getting type of dataframe

**<class 'pyspark.pandas.frame.DataFrame'>**

>>> **df.isnull().sum()** #checking null values

title 0

rank 0

date 0

artist 18

region 0

chart 0

trend 0

streams 2508926

dtype: int64

>>> **df.dtypes #**checking datatype

title object

rank object

date object

artist object

region object

chart object

trend object

streams object

dtype: object

**>>> df['streams']=df['streams'].astype('int64')** #converting datatype

>>>**df.describe() #getting information about data**

**>>>df['streams']=df['streams'].fillna(56740.87)** #replacing null value with mean(mean will get from above cmd)

**>>> df.isnull().sum()**

title 0

rank 0

date 0

artist 18

region 0

chart 0

trend 0

streams 0

dtype: int64

**>>> df=df.dropna()** #dropping null rows

**>>> df.isnull().sum()**

title 0

rank 0

date 0

artist 0

region 0

chart 0

trend 0

streams 0

dtype: int64

**>>>df.rename(columns={'date':'year'},inplace=True)** #renaming column date to year

**>>>df=df.sort\_values(by='rank',ascending=True)** #sorting data according to rank

**To save as excel**

**Procedure(we used):first we convert pyspark.pandas as csv so we got the partitioned file then we load into spark dataframe then we convert into pandas dataframe then from panda we convert into csv file**

**Df.to\_csv(‘sorted.csv’) #this will give you an partition file for spark only**

**Pyspark->spark->pandas->csv**

**Df=spark.read.csv(‘sorted.csv’)**

**Df.toPandas() this cmd will not work now**

**Df.to\_csv(‘sorted.csv’)**

**Now you will have csv file**

***IMP NOTE:-*** (diff between pyspark pandas and pandas)

if ypu import pyspark.pandas then it will export csv as partitions and if you use pandas only in spark then it will import as a csv file

Pyspark. pandas is an API that allows you to use pandas functions and operations on "spark data frames"

df1.shape ==> rows are 11496684 and 8 columns

in pyspark.pandas type is <class 'pyspark.pandas.frame.DataFrame'>

in pandas in spark type is <class 'pandas.core.frame.DataFrame'>

you can directly access the data using the folder name

pd.set\_option('compute.ops\_on\_diff\_frames', True)

Cannot combine the series or dataframe because it comes from a different dataframe.

toPandas() in my pc does not work because spark uses inmemory computation and here my pc memory is not enough to process it for that more memory is required