import findspark

findspark.init()

1. Clean and Transform data for proper processing and getting complete insights without any garbage values.

import pyspark

from pyspark.sql import SparkSession

spark = SparkSession.Builder().master("local[1]").appName("SparkByExamples.com").getOrCreate()

import pyspark

from pyspark.sql import SparkSession

df=spark.read.csv(r'C:\\Users\\AnilKumar\\Downloads\\ASSIGNMENT FOLDER\\compliants assignment folder\\Complaints.csv')

df.show()

df.count()

+------------+--------------------+--------------------+--------------------+--------------------+-----+--------+-------------+-------------+--------------------+--------------------+--------------------+----------------+------------------+

| \_c0| \_c1| \_c2| \_c3| \_c4| \_c5| \_c6| \_c7| \_c8| \_c9| \_c10| \_c11| \_c12| \_c13|

+------------+--------------------+--------------------+--------------------+--------------------+-----+--------+-------------+-------------+--------------------+--------------------+--------------------+----------------+------------------+

|Complaint ID| Product| Sub-product| Issue| Sub-issue|State|ZIP code|Submitted via|Date received|Date sent to company| Company| Company response|Timely response?|Consumer disputed?|

| 1114245| Debt collection| Medical|Disclosure verifi...|Not given enough ...| FL| 32219| Web| 11/13/2014| 11/13/2014|Choice Recovery, Inc|Closed with expla...| Yes| NA|

| 1114488| Debt collection| Medical|Disclosure verifi...|Right to dispute ...| TX| 75006| Web| 11/13/2014| 11/13/2014|Expert Global Sol...| In progress| Yes| NA|

| 1114255|Bank account or s...| Checking account|Deposits and with...| NA| NY| 11102| Web| 11/13/2014| 11/13/2014|FNIS (Fidelity Na...| In progress| Yes| NA|

| 1115106| Debt collection|Other (phone, hea...|Communication tac...|Frequent or repea...| GA| 31721| Web| 11/13/2014| 11/13/2014|Expert Global Sol...| In progress| Yes| NA|

| 1115890| Credit reporting| NA|Incorrect informa...|Information is no...| FL| 33461| Web| 11-12-2014| 11/13/2014| TransUnion| In progress| Yes| NA|

| 1114180| Debt collection| Credit card|Cont'd attempts c...| Debt is not mine| CA| 95035| Web| 11-12-2014| 11/13/2014|Expert Global Sol...| In progress| Yes| NA|

| 1114124| Consumer loan| Vehicle loan|Managing the loan...| NA| MI| 48430| Fax| 11-12-2014| 11/13/2014| Ally Financial Inc| In progress| Yes| NA|

| 1113211| Debt collection| Credit card|False statements ...|Indicated committ...| MS| 39056| Web| 11-12-2014| 11-12-2014|Premium Asset Ser...| In progress| Yes| NA|

| 1112443| Consumer loan| Installment loan|Managing the loan...| NA| CA| 94509| Web| 11-12-2014| 11-12-2014| Ally Financial Inc| In progress| Yes| NA|

| 1112618| Debt collection| Medical|Disclosure verifi...|Not given enough ...| FL| 34743| Web| 11-12-2014| 11-12-2014|Enhanced Recovery...|Closed with non-m...| Yes| NA|

| 1111987| Credit reporting| NA|Credit reporting ...|No notice of inve...| NJ| 8852| Web| 11-12-2014| 11-12-2014| TransUnion|Closed with expla...| Yes| NA|

| 1113072| Credit reporting| NA|Incorrect informa...| Account status| NY| 11514| Web| 11-12-2014| 11-12-2014| TransUnion|Closed with expla...| Yes| NA|

| 1112733| Debt collection| Medical|Communication tac...|Used obscene/prof...| WA| 98034| Web| 11-12-2014| 11-12-2014|Renton Collection...|Closed with expla...| Yes| NA|

| 1112115| Mortgage|Conventional fixe...|Loan modification...| NA| NJ| 8527| Web| 11-12-2014| 11-12-2014| Seterus|Closed with expla...| Yes| NA|

| 1112705| Credit reporting| NA|Unable to get cre...|Problem getting r...| CA| 90716| Web| 11-12-2014| 11-12-2014| TransUnion|Closed with expla...| Yes| NA|

| 1113116| Credit reporting| NA|Incorrect informa...| Account status| CT| 6242| Web| 11-12-2014| 11-12-2014| TransUnion|Closed with expla...| Yes| NA|

| 1112181| Debt collection| Credit card|Cont'd attempts c...| Debt is not mine| FL| 34481| Web| 11-12-2014| 11-12-2014|Commonwealth Fina...|Closed with expla...| Yes| NA|

| 1113153| Debt collection|Other (phone, hea...|Cont'd attempts c...| Debt was paid| GA| 30345| Web| 11-12-2014| 11-12-2014| EOS Holdings, Inc|Closed with expla...| Yes| NA|

| 1112011| Debt collection| Medical|Cont'd attempts c...|Debt resulted fro...| SC| 29360| Web| 11-12-2014| 11-12-2014|Revenue Recovery ...|Closed with expla...| Yes| NA|

+------------+--------------------+--------------------+--------------------+--------------------+-----+--------+-------------+-------------+--------------------+--------------------+--------------------+----------------+------------------+

only showing top 20 rows

Out[12]:

312913

2.Find the number of complaints for which the Company response is currently "in progress".

df.createOrReplaceTempView("complaints")

df2 = spark.sql("select count(complaints.\_c0) from complaints where complaints.\_c11='In progress'")

df2.show()

output :-

+----------+

|count(\_c0)|

+----------+

| 2612|

+----------+

3.Which company has the maximum consumer complaints.

from pyspark.sql.types import IntegerType

df=df.withColumn('\_c0',df['\_c0'].cast(IntegerType()))

df.groupBy('\_c10').max('\_c0').show()

+--------------------+--------+

| \_c10|max(\_c0)|

+--------------------+--------+

|Praxis Financial ...| 1083515|

|Worldwide Process...| 1076364|

|Lincoln Mortgage ...| 1071017|

|Law Office of Joe...| 1064107|

|Array Services Gr...| 1058623|

| Loan To Learn| 1035595|

| PlusFour, Inc| 998737|

|South West Recove...| 910341|

|Investigation & R...| 887926|

|F&L Marketing Ent...| 662753|

|Summit Mortgage C...| 526370|

|Greenlight Financial| 410462|

| Security Finance| 1091472|

|Capital Markets C...| 1079608|

|Law Office of Har...| 1059214|

|Holloway & Moxley...| 1055073|

| Toll Brothers Inc| 1056117|

|Brazos Higher Edu...| 1046185|

| BAM Financial, LLC| 934807|

|FirsTrust Mortgag...| 760821|

+--------------------+--------+

only showing top 20 rows

4.Which Companies is able to solve issues of customers (on the terms of Company response and timely response)

df.createOrReplaceTempView("complaints")

df3=spark.sql("select \_c10 from complaints where complaints.\_c11='Closed with explanation' and complaints.\_c12 ='Yes'").show()

+--------------------+

| \_c10|

+--------------------+

|Choice Recovery, Inc|

| TransUnion|

| TransUnion|

|Renton Collection...|

| Seterus|

| TransUnion|

| TransUnion|

|Commonwealth Fina...|

| EOS Holdings, Inc|

|Revenue Recovery ...|

| TransUnion|

|GLA Collection Co...|

| Wells Fargo|

| TransUnion|

| TransUnion|

| TransUnion|

|Stellar Recovery Inc|

| TransUnion|

|Convergent Resour...|

|Southwest Recover...|

+--------------------+

5.Which companies are having least response time for a complaint raised?

df=spark.sql("select max(\_c10) from complaints where \_c12='Yes'").

Show()

Output :-

+------------+

| max(\_c10)|

+------------+

|iServe Trust|

+------------+

6.Find the issue that occurred mostly.

import org.apache.spark.sql.SparkSession  
import org.apache.spark.sql.functions.*desc*//import scala.util.control.Breaks.break  
//import spark.implicits.\_  
//import org.apache.spark.sql.functions.\_  
//import scala.collection.immutable.Nil.groupBy  
object IssueOccured extends App{  
 val *spark*: SparkSession = SparkSession.*builder*().master("local[\*]").appName("SparkByExamples.com")  
 .getOrCreate()  
 val *df*=*spark*.read.option("header",true).csv(path = "src/main/data/complaints.csv")  
 *df*.show()  
 *df*.createOrReplaceTempView(viewName="facts")  
 val *table* = *spark*.sql(sqlText = "select issue,count(issue) from facts group by issue order by 2 desc limit 1")  
 *println*(*table*.show())  
}

output :-

+--------------------+------------+

| issue|count(issue)|

+--------------------+------------+

|Loan modification...| 70487|

+--------------------+------------+

7. Which are the Top 5 states having the highest volume of complaints coming.

import org.apache.spark.sql.SparkSession  
import org.apache.spark.sql.functions.*desc*//import scala.util.control.Breaks.break  
//import spark.implicits.\_  
//import org.apache.spark.sql.functions.\_  
//import scala.collection.immutable.Nil.groupBy  
object IssueOccured extends App{  
 val *spark*: SparkSession = SparkSession.*builder*().master("local[\*]").appName("SparkByExamples.com")  
 .getOrCreate()  
 val *df*=*spark*.read.option("header",true).csv(path = "src/main/data/complaints.csv")  
 *df*.show()  
 *df*.createOrReplaceTempView(viewName="facts")  
 val *table* = *spark*.sql(sqlText = "select state,count(issue) from facts group by state order by 2 desc limit 5")  
 *println*(*table*.show())  
}

+-----+------------+

|state|count(issue)|

+-----+------------+

| CA| 47076|

| FL| 30164|

| TX| 21711|

| NY| 21519|

| GA| 13514|

+-----+------------+

8.Which are the Top 5 companies people complaining the most.

import org.apache.spark.sql.SparkSession  
import org.apache.spark.sql.functions.*desc*//import scala.util.control.Breaks.break  
//import spark.implicits.\_  
//import org.apache.spark.sql.functions.\_  
//import scala.collection.immutable.Nil.groupBy  
object IssueOccured extends App{  
 val *spark*: SparkSession = SparkSession.*builder*().master("local[\*]").appName("SparkByExamples.com")  
 .getOrCreate()  
 val *df*=*spark*.read.option("header",true).csv(path = "src/main/data/complaints.csv")  
 *df*.show()  
 *df*.createOrReplaceTempView(viewName="facts")  
 val *table* = *spark*.sql(sqlText = "select company,count(issue) from facts group by company,issue order by 2 desc limit 5")  
 *println*(*table*.show())  
}

output :-

+---------------+------------+

| company|count(issue)|

+---------------+------------+

|Bank of America| 20038|

| Wells Fargo| 10874|

| Experian| 10377|

| Equifax| 9358|

| TransUnion| 8047|

+---------------+------------+

9. Which product has the most number of complaints.

import org.apache.spark.sql.SparkSession  
import org.apache.spark.sql.functions.*desc*//import scala.util.control.Breaks.break  
//import spark.implicits.\_  
//import org.apache.spark.sql.functions.\_  
//import scala.collection.immutable.Nil.groupBy  
object IssueOccured extends App{  
 val *spark*: SparkSession = SparkSession.*builder*().master("local[\*]").appName("SparkByExamples.com")  
 .getOrCreate()  
 val *df*=*spark*.read.option("header",true).csv(path = "src/main/data/complaints.csv")  
 *df*.show()  
 *df*.createOrReplaceTempView(viewName="facts")  
 val *table* = *spark*.sql(sqlText = "select product ,count(issue) from facts group by product,issue order by 2 desc limit 5")  
 *println*(*table*.show())  
}

output :-

+--------------------+------------+

| product|count(issue)|

+--------------------+------------+

| Mortgage| 70487|

| Mortgage| 36767|

| Credit reporting| 29069|

| Debt collection| 17972|

|Bank account or s...| 16205|

+--------------------+------------+