**USA Consumer Forum Data Analysis**

**Flume:**

flume-ng agent --conf-file /usr/local/flume/conf/flumeproject.conf --name agent -Dflume.root.logger=INFO,console

**Inside .conf file:**

agent1.sources = mysrc

agent1.sinks = hdfsdest

agent1.channels = mychannel

agent1.sources.mysrc.type = exec

agent1.sources.mysrc.command = hadoop fs -put /home/acadgild/Downloads/Execute/flume/pro/Consumer\_Complaints /user/flume/mainproject

agent1.sinks.hdfsdest.type = hdfs

agent1.sinks.hdfsdest.hdfs.path = hdfs://localhost:9000/user/flume/mainproject

agent1.channels.mychannel.type = memory

agent1.sources.mysrc.channels = mychannel

agent1.sinks.hdfsdest.channel = mychannel

**First cleaning the raw data using mapreduce**

hadoop jar /home/acadgild/Downloads/Exefile/clean.jar cleaning.Cleandata /user/flume/mainproject cleandata

**1. Write a pig script to find no of complaints which got timely response**

**Loading data into PIG**

A = load '/user/acadgild/cleandata' using PigStorage(',') As (Date\_received:chararray,Product:chararray,Sub\_product:chararray,Issue:chararray,Sub\_issue:chararray,Consumer\_complaint\_narrative:chararray,Company\_public\_response:chararray,Company:chararray,State:chararray,ZIP\_code:int,Submitted\_via:chararray,Date\_sent\_to\_company:chararray,Company\_response\_to\_consumer:chararray,Timely\_response:chararray,Consumer\_disputed:chararray,Complaint\_ID:int);

**Filtering:**

B = Filter A by Timely\_response == 'Yes';

**Grouping:**

C = Group B ALL;

**Foreach:**

D = Foreach C Generate COUNT($0);

Dump D;

**Storing:**

Store D into '/user/acadgild/mainproject/time' Using PigStorage(',');

**Exporting to SQL using sqoop:**

sqoop export --connect jdbc:mysql://localhost/proj2 --table mainproject\_time --export-dir /user/acadgild/mainproject/udf\_result/part-m-00000 --fields-terminated-by ',' --username root -P

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Write a pig script to find no**

**of complaints where consumer forum forwarded the complaint**

**same day they received to respective company**

**Filtering:**

E = Filter A By Date\_received == Date\_sent\_to\_company;

**Grouping:**

F = Group E by Company;

**Foreach:**

G = Foreach F Generate Group, COUNT(E.$0);

**Storing:**

Store G into '/user/acadgild/mainproject/rese' Using PigStorage(',');

**Exporting to SQL using sqoop:**

sqoop export --connect jdbc:mysql://localhost/proj2 --table mainproject\_rese --export-dir /user/acadgild/mainproject/udf\_result/part-m-00000 --fields-terminated-by ',' --username root -P

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Write a pig script to find list of companies toping in complaint chart (companies with**

**maximum number of complaints)**

**Filtering:**

H = Filter A By Consumer\_complaint\_narrative != '';

**Grouping:**

I = Group H by Company;

**Foreach:**

J = Foreach I Generate Group, COUNT(H.$0);

**OrderBy:**

K = Order J By $1 DESC;

**Limit:**

L = Limit K 1;

**Storing:**

Store L into '/user/acadgild/mainproject/null' Using PigStorage(',');

**Exporting to SQL using sqoop:**

sqoop export --connect jdbc:mysql://localhost/proj2 --table mainproject\_null --export-dir /user/acadgild/mainproject/udf\_result/part-m-00000 --fields-terminated-by ',' --username root -P

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Write a pig script to find no**

**of complaints filed with product type has &quot;Debt**

**collection&quot; for the year 2015**

**Filtering:**

M = Filter A by Product == 'Debt collection';

**Filtering:**

N = FILTER M BY (Date\_received matches '.\*2015.\*');

**Grouping:**

O = Group N ALL;

**Foreach:**

P = Foreach O Generate COUNT(N.$0);

**Storing:**

Store P into '/user/acadgild/mainproject' Using PigStorage(',');

**Exporting to SQL using sqoop:**

sqoop export --connect jdbc:mysql://localhost/proj2 --table mainproject\_2015 --export-dir /user/acadgild/mainproject/udf\_result/part-m-00000 --fields-terminated-by ',' --username root -P