

**Analyzing Income Tax From Aadhar Dataset Using Hadoop**

***in partial fulfillment for the award of the degree***

***of***

**Bachelor of Technology**

**in**

**Information Technology**

**Dr. Mahalingam College of Engineering and Technology**

**Pollachi – 642 003**

**An Autonomous Institution**

**Affiliated to Anna University, Chennai - 600 025**

**MAY 2017**

**MINI PROJECT REPORT**

**Submitted by**

**JEEVANANTHAM C (14BIT068)**

**SARAVANAN C (14BIT076)**

**JEEVANANTHAM C (14BIT068)**

**SARAVANAN C (14BIT076)**

**INTERNAL EXAMINER**

Submitted for the Autonomous End Semester Examination Mini Project

viva-voce held on \_\_\_\_\_\_\_\_\_\_\_

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

Dr.S.Ramakrishnan,

Prof. & Head,

Information Technology,

Dr. Mahalingam College of Engineering and Technology,

NPT-MCET Campus,

Pollachi – 642 003, India.

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

Mr.Balasamy,

Supervisor,

Assistant Professor,

Information Technology,

Dr. Mahalingam College of Engineering and Technology,

NPT-MCET Campus,

Pollachi – 642 003, India.

Pollachi – 642003 India

**Dr. Mahalingam College of Engineering and Technology**

**Pollachi – 642 003**

**An Autonomous Institution**

**Affiliated to Anna University, Chennai - 600 025**

**EXTERNAL EXAMINER**

who carried out the project work under my supervision.

Certified that this mini project report, **“Analyzing Income Tax From Aadhar Dataset Using Hadoop ”** is the bonafide work of

BONAFIDE CERTIFICATE

**ABSTRACT**

Our project is about analysing the Income Tax from Aadhar data set using Hadoop Big Data.Analysis of the Aadhar Data set provides valuable details. To analyze this large amount of data, we need parallel processing system and reliable data storage mechanism. So we move to Hadoop distributed file system, map reducing framework and hive database.

The Sample Aadhar Data set is taken from the Government website. Then, the dataset is to be inserted into the Hive Database table after Pre-processing using map reduce is done.

By processing the Aadhar Data set, we can Predict the total amount of Income Tax that would be collected by the government and we can eliminate the unwanted fields Registrar,District etc. The Minimized data is then stored in HDFS and processed using HIVE query Language. Finally, the data is visualized using Splunk Tool.

**ACKNOWLEDGEMENT**

We would like to praise the almighty for his grace to bless us with this project and for giving us the confidence to complete it successfully.

We wish to acknowledge with thanks for the excellent encouragement given by the management of our college and **Dr.C.Ramaswamy., M.E., F.I.V., Ph.D.,** Secretary, NIA Educational Institutions.

We sincerely thank our Director **Dr.S.Vijayarangan., M.E., Ph.D., FIE.,** for his constant support for our project.

We wish to express our hearty thanks to **Dr.M.Ramakrishnan, B.E., M.E., Ph.D.,** Principal of our college, for his constant motivation regarding our Project work.

We express our extreme gratefulness to **Dr.S.Ramakrishnan, M.E., Ph.D.,** Professor & Head of the Department, Information Technology for his direction delivered at all times required.

We express our warm and sincere thanks to our guide **Mr.Balasamy, M.E.,** Assistant Professor, Information Technology, for her tireless and meticulous efforts in bringing out this project to its logical conclusion.

We are committed to place our heartfelt thanks to our project coordinator and all our teaching and non-teaching staff members, lab technicians and friends, and all the noble hearts that gave us immense encouragement towards the completion of our Project.

TABLE OF CONTENTS

1. INTRODUCTION 6

1.1. Overview 6

1.2. Aim and Objective 6

1.3. Existing System 7

1.4. Proposed System 7

1.5. Architecture 8

2. REQUIREMENT AND SPECIFICATION 11

2.1. Software Requirements 11

2.2. Hardware Requirements 11

3. IMPLEMENTATION 12

3.1 Modules 12

3.2 Module Description 12

3.2.1. Pre-Processing 12

3.2.2. Data Storage 12

3.2.3. Analysis 13

3.2.4. Visualization 13

3.2.5. Prediction 13

3.3. Algorithm Implementation 13

3.4. Steps for Run the System 14

4. RESULT AND DESCRIPTION 15

5.REFERENCES 15

APPENDIX 16

A. Dataset Structure 16

B. Coding of Map Reduce program 17

C. Screen shots of the Running system 20

LIST OF ABBREVIATIONS (in alphabetic order)

HDFS: Hadoop Distributed File System.

MR: Map Reduce.

**CHAPTER 1**

**INTRODUCTION**

**1.INTRODUCTION:**

The data is too big, moves too fast and is unstructured and doesn't fit the structures of the architectures. To gain value from this data we need an alternative way to process it . Various fields for example that generate such large amounts of huge data are Facebook, Twitter ,Weather stations ,New York Stock Exchange , Worldwide electric transmissions etc. Thus in our project we are dealing with huge amount of unstructured Aadhar Data. Our paper focuses on the shifting of processes from single node data processing to Hadoop distributed file system for faster processing and the best technique to process the queries.

**1**.**1 Overview:**

Analyzing the Aadhar data set will give valuable details. To analyze this huge amount of data we need parallel processing system and reliable data storage mechanism. So we move to Hadoop distributed file system, map reducing framework and hive database. The sample aadhar data set was taken from the government website. That the dataset was inserted into hive database table after pre-processing using map reduce.

By processing the Aadhar data set we can predict the total amount of income tax that would collected by the government and we can minimize the unwanted data like register ,district etc. And then The minimized data is then stored in hdfs and processed using hive query language. Finally the data is visualized using splunk tool.

**1.2 Aim and Objective of the Project :**

* To calculate the total amount of income tax that collected by Government by individual State.
* To provide visualization of the obtained data and compare the income tax among the states.

**1.3 Existing System:**

In the Existing system Income tax is calculated manually which takes more time. Evaluating the citizen’s Income tax Department from their Bank Accounts if it crosses 2.5 lakhs transaction per annum. Grouping the Income Tax of State wise is tedious process. Individual Citizen’s Taxation is a necessary and calculating individual’s is more complicated .

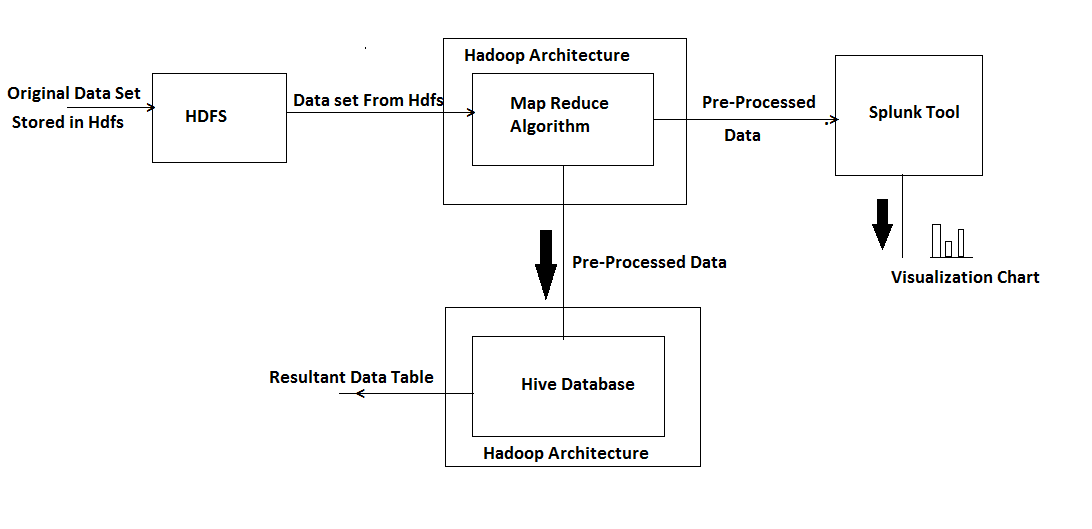
**1.4 Proposed System:**

Hadoop is an opensource large-scale data processing framework that supports distributed processing of large chunks of data using simple programming models. The Apache Hadoop project consists of the HDFS and Hadoop Map Reduce in addition to other modules, that provides high speed clustered processing for the analysis of large set of data smoothly and efficiently.so we Proposed the system for calculating the income tax for all states using aadhar dataset.

There are five modules are available. First the original Dataset was preprocessed using map reduce algorithm.In pre processing the unwanted fields like District , subdistrict etc. are eliminated and the new dataset was created.the resultant data set is stored into the Hive database table and also stored in to the Splunk file system.the analysis is performed in two different ways.In hive using hive queriy the total income tax was calculated.In Splunk the loaded data fields are separated and in visualization part provide apropirate fields to X - axis and Y – axis to Visualize it.And finally the Prediction of 30 % of Income wasdone using Splunk Machine learning Toolkit.

**1.5 Architecture**

**Workflow**:



**Name Node:**

The name node is the suitable hardware that contains the GNU/ LINUX OS and name node framework software. The name node in the file system acts as a master server and it does its tasks as follows:

• The execution operation of the file system such as renaming, closing and opening files and directories is done by name node.

• The name space of the file system is managed by name node.

• Regulates the client’s access to files.

**Data Node:**

As in the name node, the data node is a suitable hardware that resides the GNU/ LINUX operating system and data node software. The cluster contains many nodes and each node in that cluster will be a data node. The data storage of the system is maintained by these nodes. Tasks of data node are:

Based upon the client request, it performs the react and writes operation. • The name node will give certain instructions to data node such as creation, deletion and replication and it performs all these operations.

These are the two architechtures used in hadoop clustering and processing,when the hadoop system gets configured it needs the mapreduce program for the running our analytics ,where the hadoop process peta bytes of data easier and make the output simpler in simple data format.

**Map Reduce :**

The basic definition of map reduce is defined in previous section. In Map Reduce, we are able to write application to process the huge amount of data in parallel.

The Map Reduce mainly contains two main tasks, namely “MAP” and “REDUCE”. The task of the map is to take a set of data and converting this data to another set of data, where the individual data elements are broken down into tuples i.e., key/ value pairs. The work of reduce task is to take the O/O of map task as input and reduce those data tuples into smaller tuples. As the name indicates the reduce task is always done after map task.

**Yarn Manager:**

Yarn consist of central resource manager which arbitrates all available cluster resources area and as per node manager which takes direction from the Resource Manager and responsible for managing resources available on a single node. The resource manager is the master that judges all the available cluster resources and thus helps manage the distributed application running on the YARN system.

**Description:**

The resource manager works together with node manager and application manager. Node manager will take the instructions from the resource manager and will manage resource available on a single node. Application masters are responsible for negotiating resources with resource manager and for working with node manager to start the containers. The Node Manager (NM) is also YARN’s prenode agent, and takes care of the individual compute nodes in Hadoop cluster.

**Chapter 2**

**2. Requirement Specifications:**

**2.1 Software Requirements:**

* Ubuntu OS (used Ubuntu16.04).
* Java jdk (used jdk version 1.8).
* Hadoop Architecture (used Hadoop 2.6.0).
* Hive Database (used Hive 2.1.1).
* Splunk Platform(used splunk 6.5.3).
  1. **Hardware Requirements:**

**Minimum Requirements:**

* GB of RAM
* 3rd generation Processor
* 500 GB of Hard Disk
* Virtualization Enabled System.

**Chapter 3**

**3. IMPLEMENTATION:**

**3.1 MODULES:**

1. Pre-Processing
2. Data Storage
3. Analysis
4. Visualization
5. Prediction

**3.2 MODULE DESCRIPTION:**

**3.2.1 Pre-Processing:**

Technology used Machine learning program jar file. From the Original data set using the field named Annual income calculate the Income tax amount.If(Income > 250000) then Tax=(0.05)\*(Income-250000)Using the above formula calculate the total income tax amount according to Individual State.

**3.2.2 Data Storage:**

After Hive Installation go to the hive Database by using hive command.Create the Database that the procedure same as the MySql Database commands. Create external table that stored as the text file.The reason for the command stored as the text file is to load the text document directly from the local machine and easy access.After creating a table then load the data directly from the local machine by using **“load data local inpath ‘path’ overwrite into table table\_name”** command.

**3.2.3 Analysis:**

Using Hive Db Query we can generate different type of output as an Information. Calculating the total amount of Income tax that collected by government for individual State using the command “**select state,sum(tax) from details group by state”.** It produces the result that the total amount of Income tax that collected by government for individual State .

**3.2.4 Visualization.**

Tools used for visualization is Splunk. The splunk is a platform that specifically for monitoring, searching, analyzing the machine generated Big Data via web style Interface. The Pre-processed data was imported into the Splunk File system. By assigning the Tax field to Y-axis and State field to X-axis the visualization was made in the form of bar chart.

**3.2.5 Prediction:**

The tool used for Prediction is Splunt Machine learning Toolkit. The Pre-processed Dataset was imported into the Slunk file system. First we choose the Machine learning algorithm for our Data set. Then choose the Prediction Field and the fields are used to prediction. After selecting the necessary details the choose the 70% of data was learning and 30% of data was prediction. Finally the prediction was done by clicking fit model.

**3.3. Algorithm Implementation.**

Installing the Eclipse Luna.

Create New project.

Create the Mapper class, Reducer Class,Driver class.

Add dependent Jar files.

Use the java class and mapper program to cut sort the columns from the dataset and calculate the Income tax.

Using Reducer it reduces the common datasets in the fields.

**Dataset descreption:**

A complex mapreduce program on Aadhar dataset. Here we use the Aadhar data set that downloaded from [www.gov.in.From](http://www.gov.in.From) that we calculate the income tax by using map reduce program.

**3.4 .Steps for Running the Hadoop Aadhar data analytics project:**

Create the project code for our convenience of output needed.

Import the project in eclipse IDE in the same way it was told in earlier guide and change the jar paths with the jar files present in the lib directory of this project.

When the project is not having any error, we will export it as a jar file, same as we did in word count map reduce guide. Right Click on the Project file and click on Export. Select jar file.

Give the path where you want to save the file.

Select the main file by clicking on browse.

Click on Finish to export.

dataset.csv is our Dataset for our project it size measures up to 50mb . we can extent up to tb’s and more.

Before running the map reduce program to check what it does, see that your cluster is up and all the Hadoop daemons are running.

Send the dataset on to HDFS.

**Command:** hdfs dfs –put /Downloads/dataset.csv /sample

**Command:** hdfs dfs -ls /

**Command:** hadoop jar final.jar /sample/dataset.csv /output

Check output file directory in HDFS.

Here it contains the part-00000 success file ,this contains our analytical output file.

**Chapter 4**

**4. Result and Discussions:**

Thus the Income tax of citizens is calculated with respect to the Person Annual Income and it is visualized by the use of splunk tool.

**Chapter 5**

**5. References:**

[1]. P.Agarwal, S.Das and A.E.Abbadi, “Bigdata and cloud computing: Current state and future opportunities” in Proc Int Conf Extending Database Technol.

[2]. J.Cohen, B.Dolan, M.Dunlap, J.M.Hellenstein and C.Welton, “Mad Skills: New Analysis practices for Bigdata”.

[3]. J.Dean and S.Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters” Commun.

[4]. K.Michael and K.W.Millen, “Bigdata: New Opportunities and New Challenges”.

[5]. M.Olson, “Hadoop: Scaleable, Flexible Data Storage and Analysis”.

[6]. Andue Riberio, Afonso Silva, Alberto Rodrigues De Silva, “Data Modelling and Data Analytics: A Survey from a big data perspective”.

**APPENDIX:**

A: Dataset Structure.

Field#  Name                           Units

———————————————

1 STATE String

2 ANNUAL\_INCOME Bigint

3 GENDER String

4 AGE Int

5 DISTRICT String

6 SUB DISTRICT String

**B: Coding of Map Reduce program:**

Import java.io.IOException;  
import java.io.Writer;  
import java.util.Iterator;  
import java.util.StringTokenizer;  
  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.Mapper;  
import org.apache.hadoop.mapreduce.Reducer;  
import org.apache.hadoop.conf.Configuration;  
  
public class Tax {   
public static class MaxTemperatureMapper extends  
 Mapper<LongWritable, Text, Text, Text> {

@Override  
public void map(LongWritable arg0, Text Value, Context context)  
 throws IOException, InterruptedException {   
String Tax="";  
String file\_arr[]=Value.toString().split("\n");  
for(int i=0;i<file\_arr.length;i++){  
String data[]=file\_arr[i].split(",");  
if(Long.parseLong(data[1])>250000){  
Long temp=(long)((0.05)\*(Long.parseLong(data[1])-250000));  
 context.write(new Text(data[0]+","+temp+","+data[2]+","+data[3]), new Text(""));

//Tax=Tax+"\n"+data[0]+","+Long.toString(temp);  
}

}  
}  
}  
//Reducer

public static class MaxTemperatureReducer extends  
 Reducer<Text, Text, Text, Text> {   
public void reduce(Text Key, Iterator<Text> Values, Context context)  
 throws IOException, InterruptedException {   
String temperature = Values.next().toString();  
 context.write(Key, new Text(""));  
}  
}   
public static void main(String[] args) throws Exception {  
//reads the default configuration of cluster from the configuration xml files  
Configuration conf = new Configuration();   
//Initializing the job with the default configuration of the cluster   
Job job = new Job(conf, "weather example");   
//Assigning the driver class name  
job.setJarByClass(Tax.class);  
//Key type coming out of mapper  
job.setMapOutputKeyClass(Text.class);   
//value type coming out of mapper  
job.setMapOutputValueClass(Text.class);  
//Defining the mapper class name  
job.setMapperClass(MaxTemperatureMapper.class);   
//Defining the reducer class name  
job.setReducerClass(MaxTemperatureReducer.class);  
//Defining input Format class which is responsible to parse the dataset into a key value pair  
job.setInputFormatClass(TextInputFormat.class);

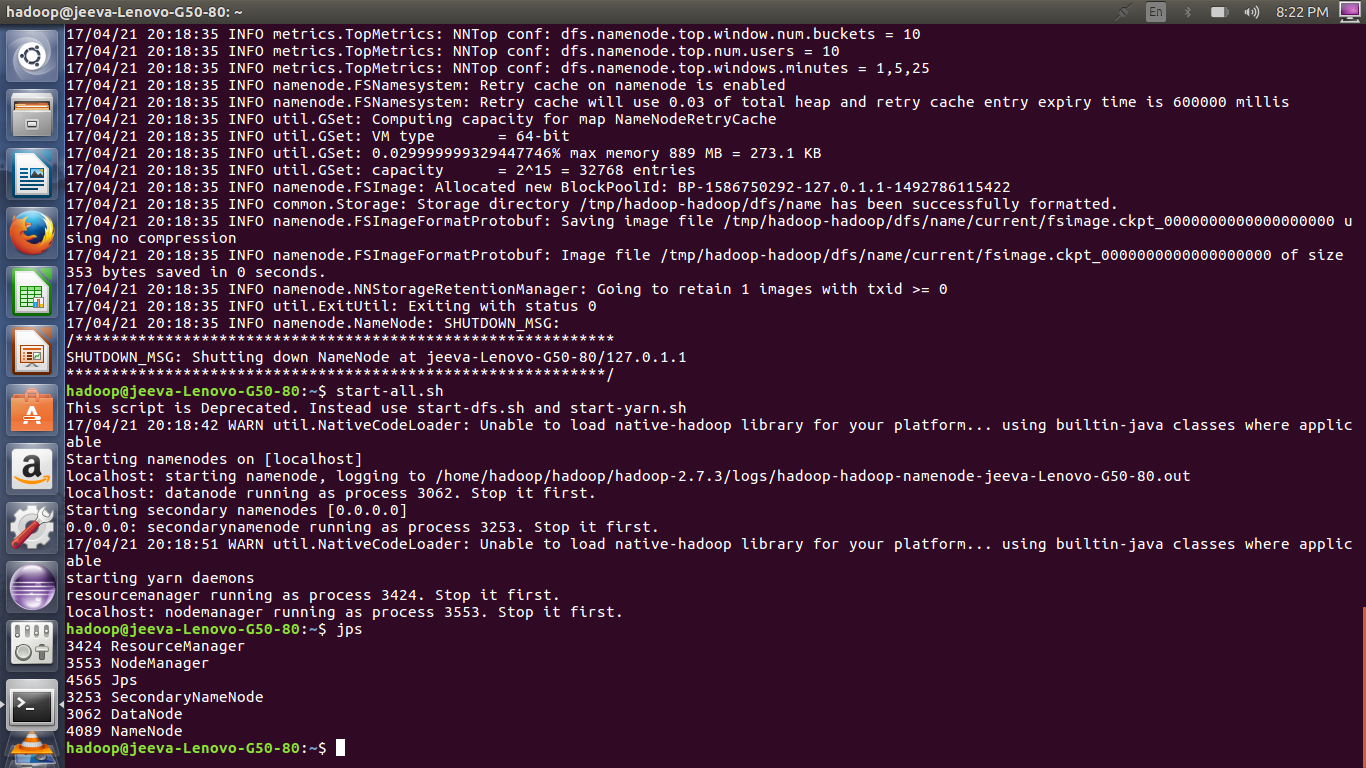
//Defining output Format class which is responsible to parse the dataset into a key value pair

job.setOutputFormatClass(TextOutputFormat.class);  
//setting the second argument as a path in a path variable  
Path OutputPath = new Path(args[1]);  
//Configuring the input path from the filesystem into the job  
FileInputFormat.addInputPath(job, new Path(args[0]));  
//Configuring the output path from the filesystem into the job  
FileOutputFormat.setOutputPath(job, new Path(args[1]));  
//deleting the context path automatically from hdfs so that we don't have delete it explicitly  
OutputPath.getFileSystem(conf).delete(OutputPath);  
//exiting the job only if the flag value becomes false  
System.exit(job.waitForCompletion(true) ? 0 : 1);  
}  
}

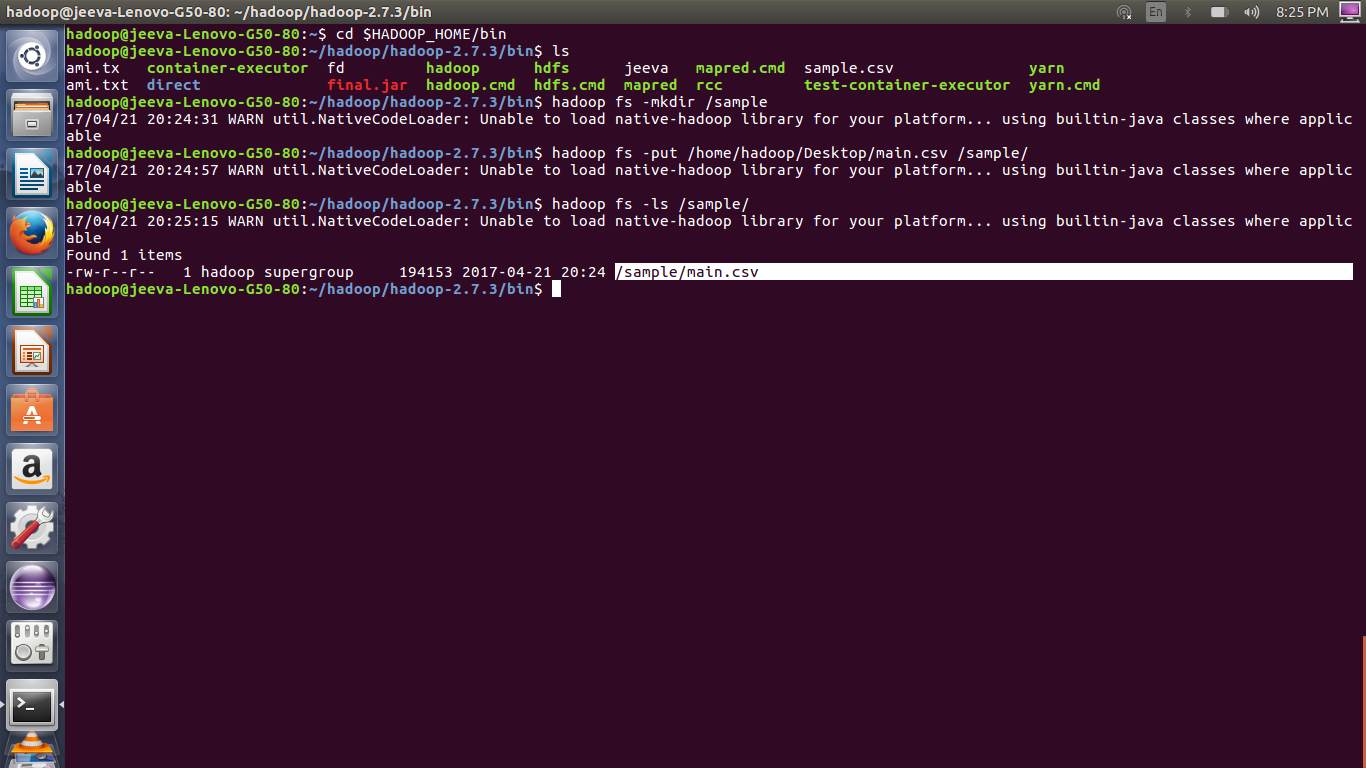
**APPENDIX :**

**C: SCREEN SHOTS OF THE RUNNING SYSTEM.**

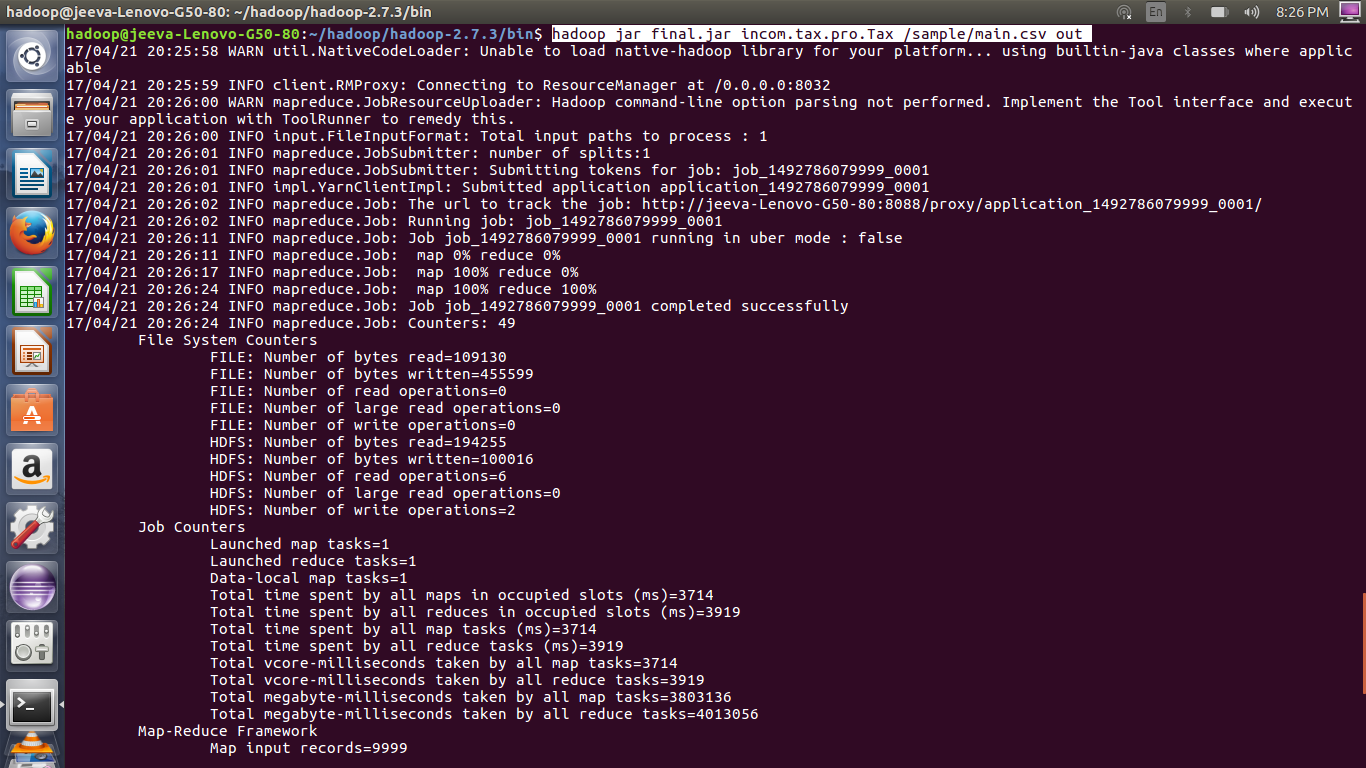
**Starting Hadoop:**



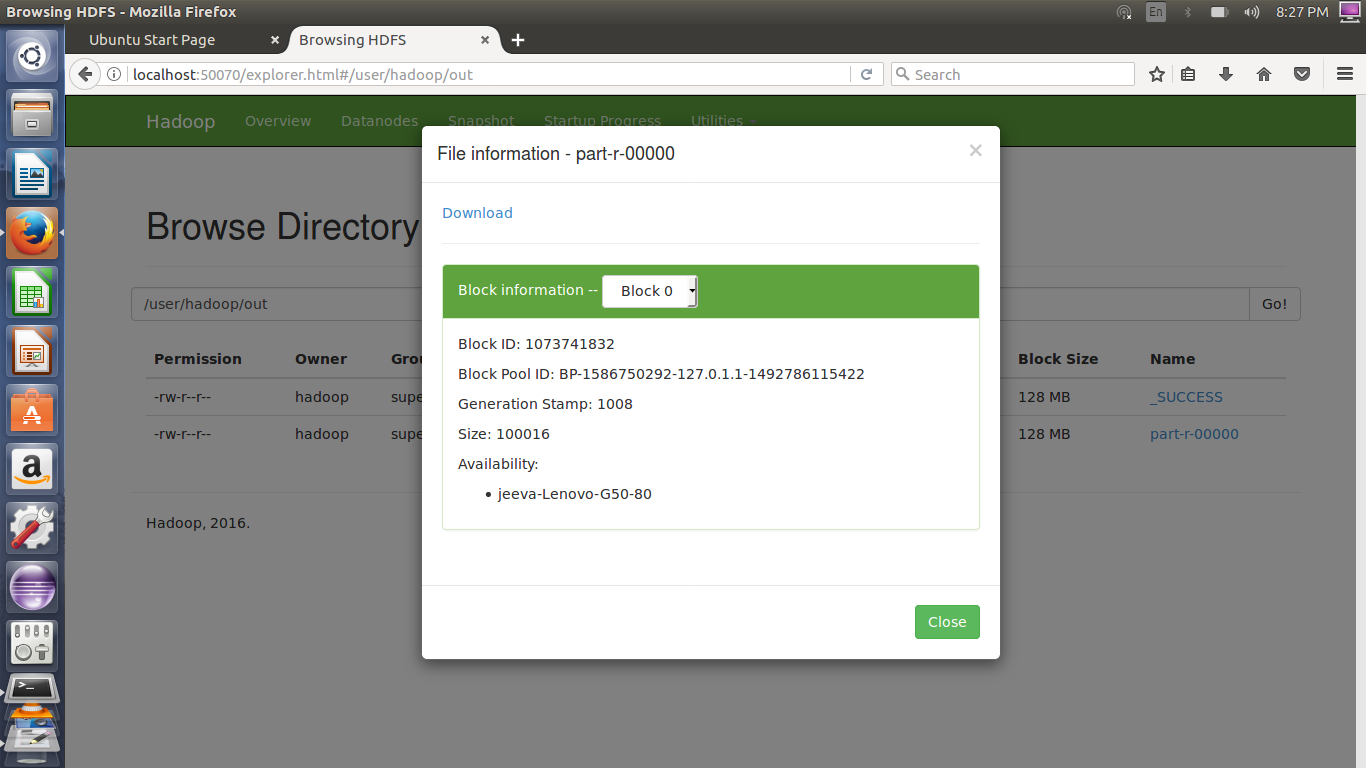
**Upload Dataset into HDFS:**



**Run the Map-Reduce Program:**



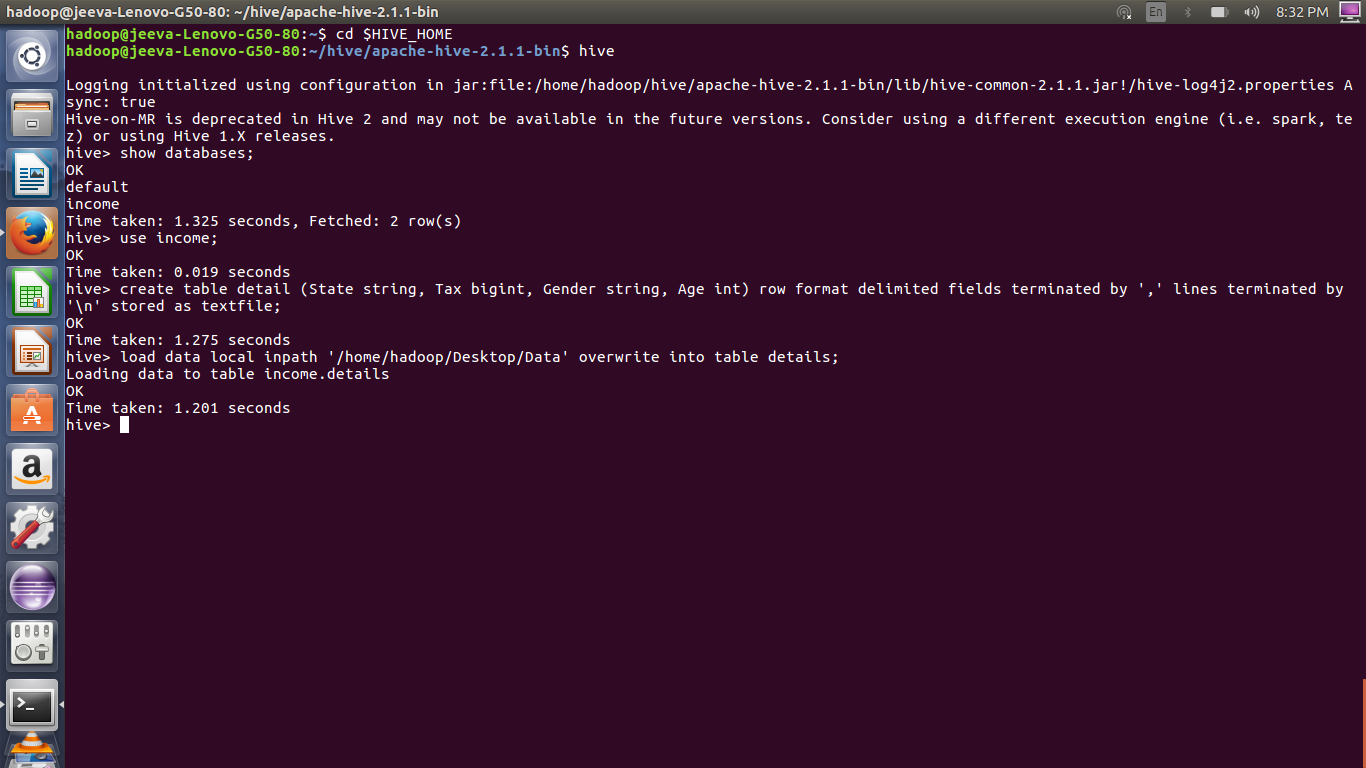
**Browse and Download the Resultant Data of Map-Reduce:**



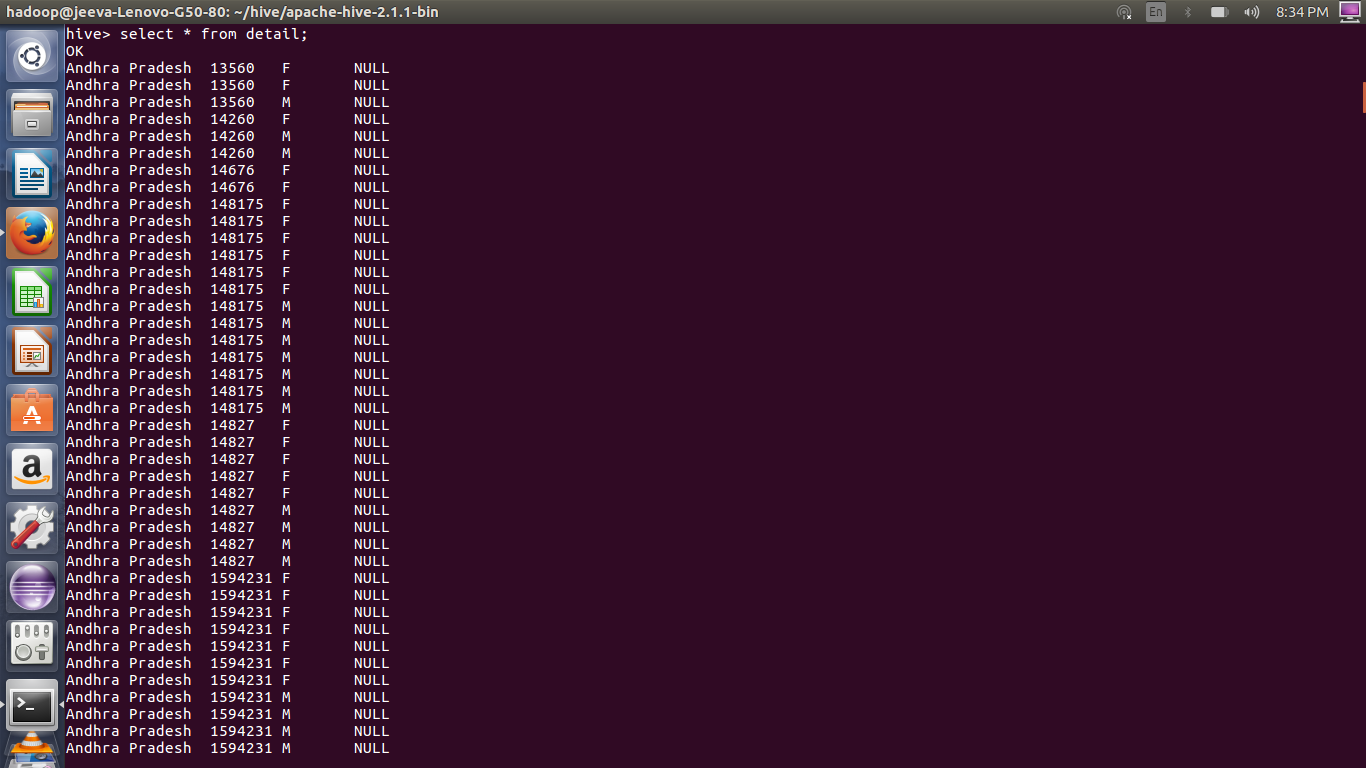
**The Pre-processed dataset:**



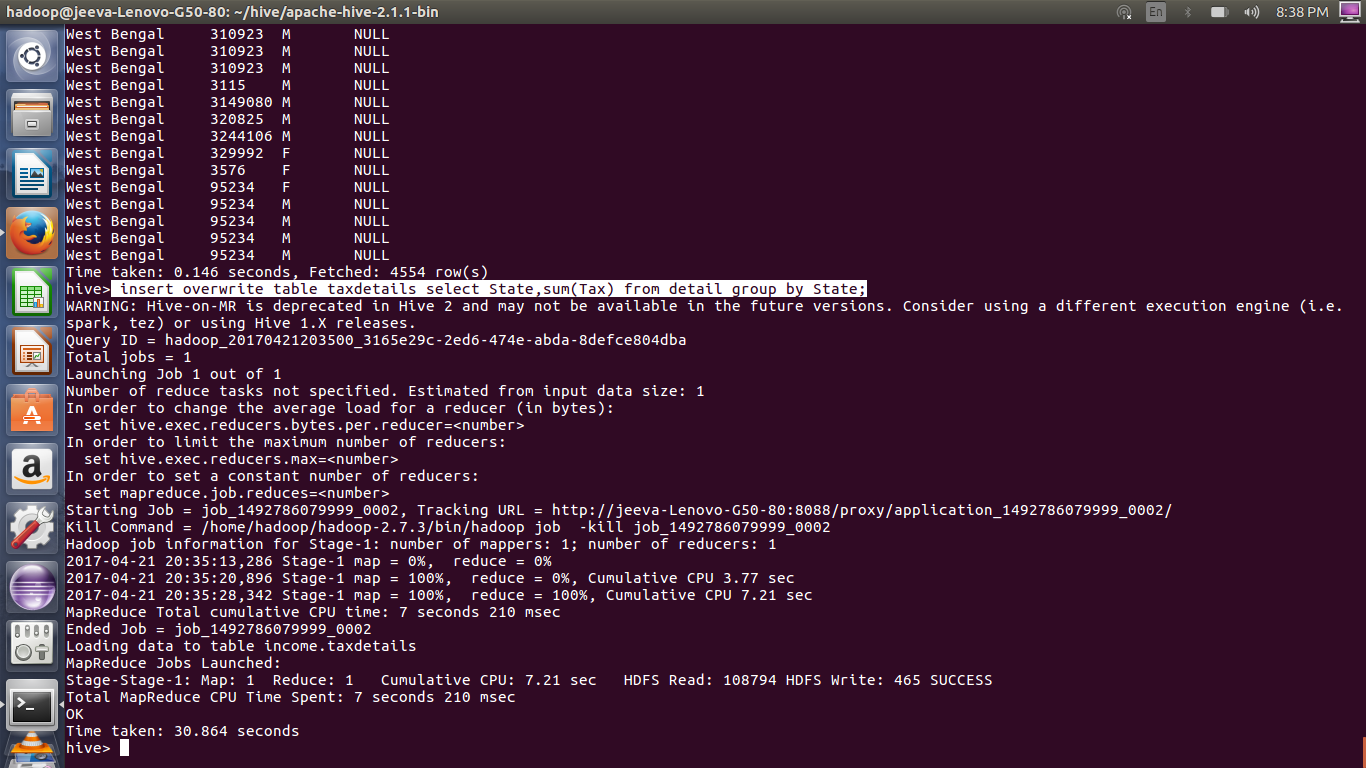
**Store the Pre-processed data into Hive Table:**



**Inserted Data:**



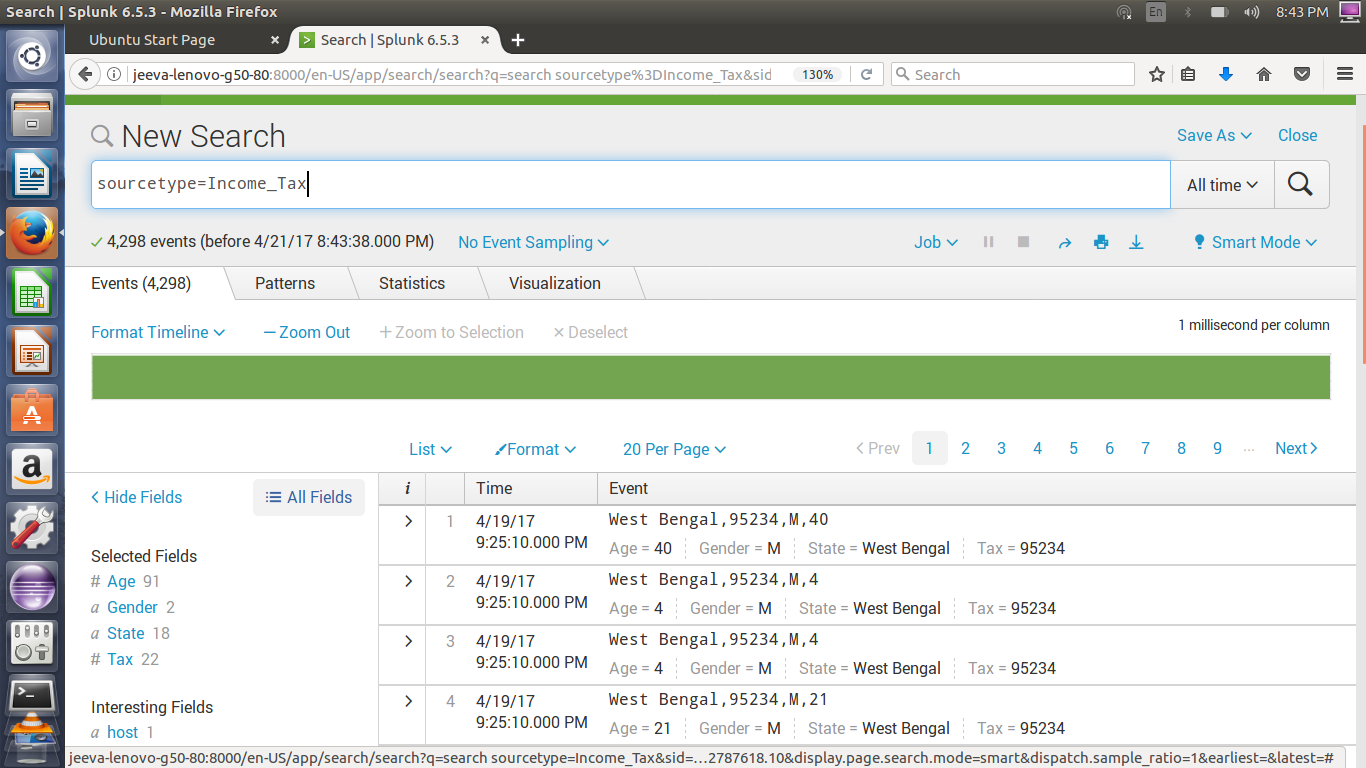
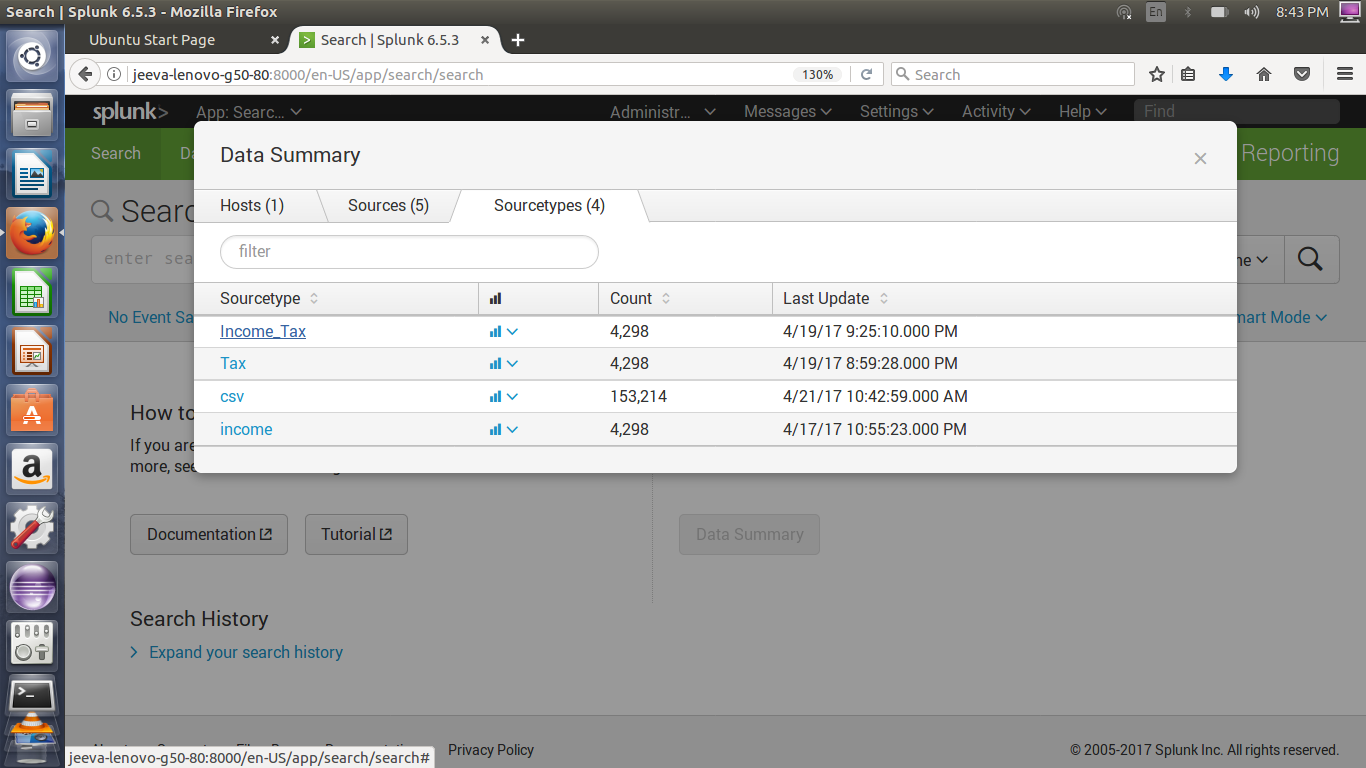
**Calculate the Income tax by Individual state using Hive query:**



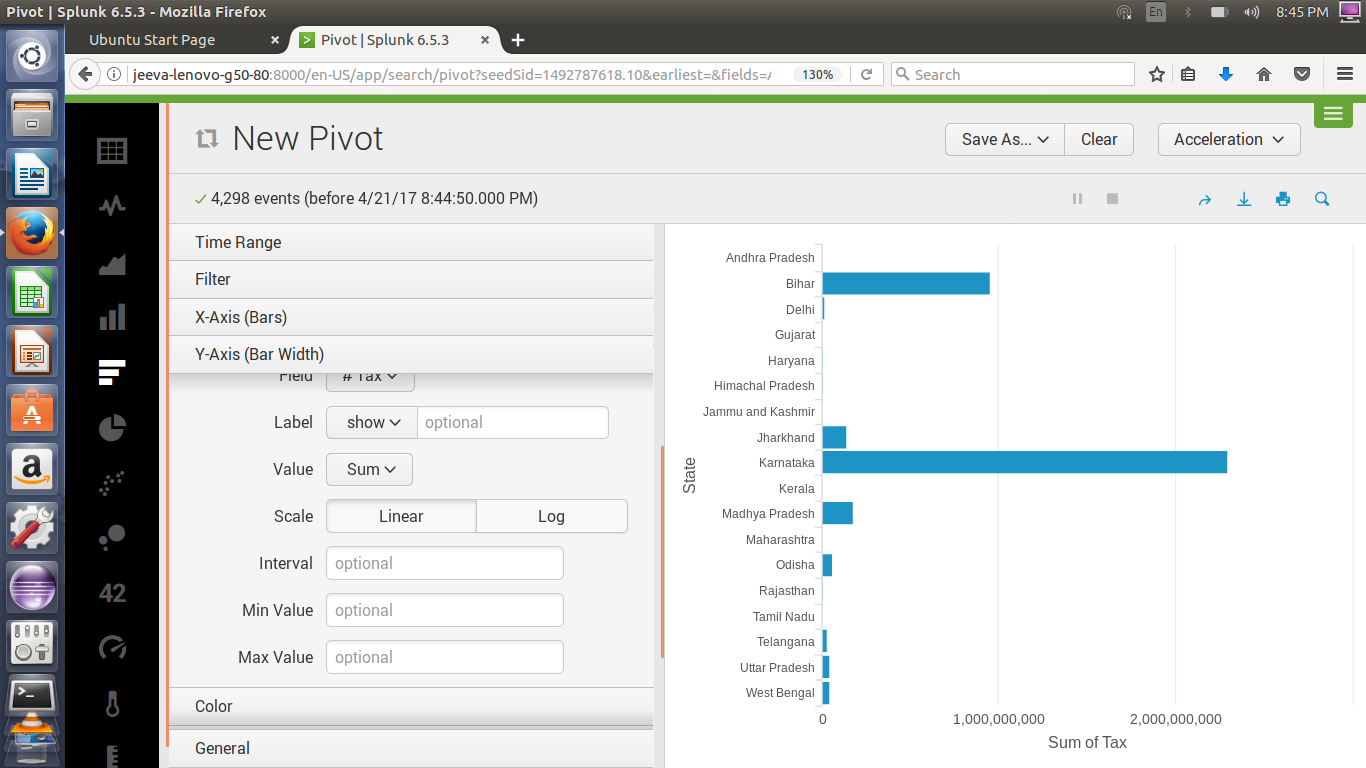
**Resultant Table:**



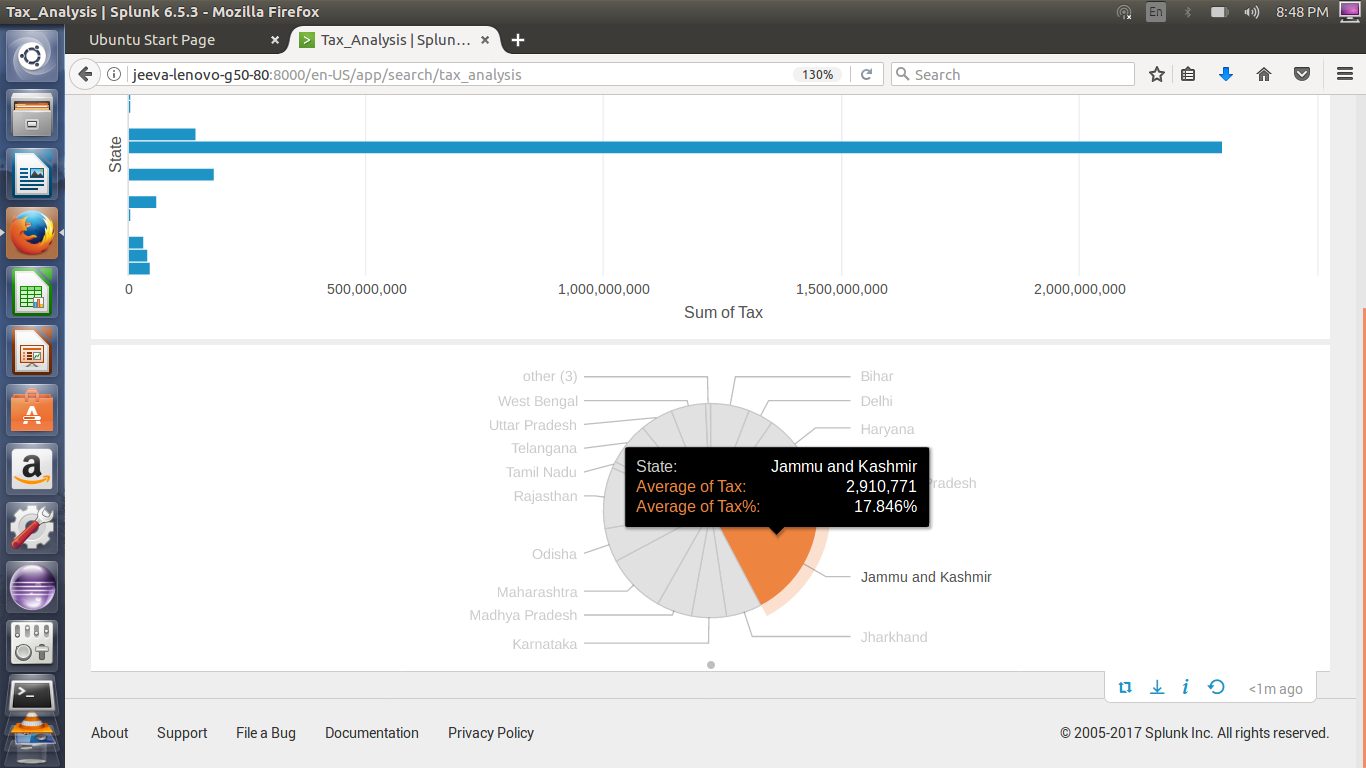
**Upload the pre-processed dataset into Splunk File system:**

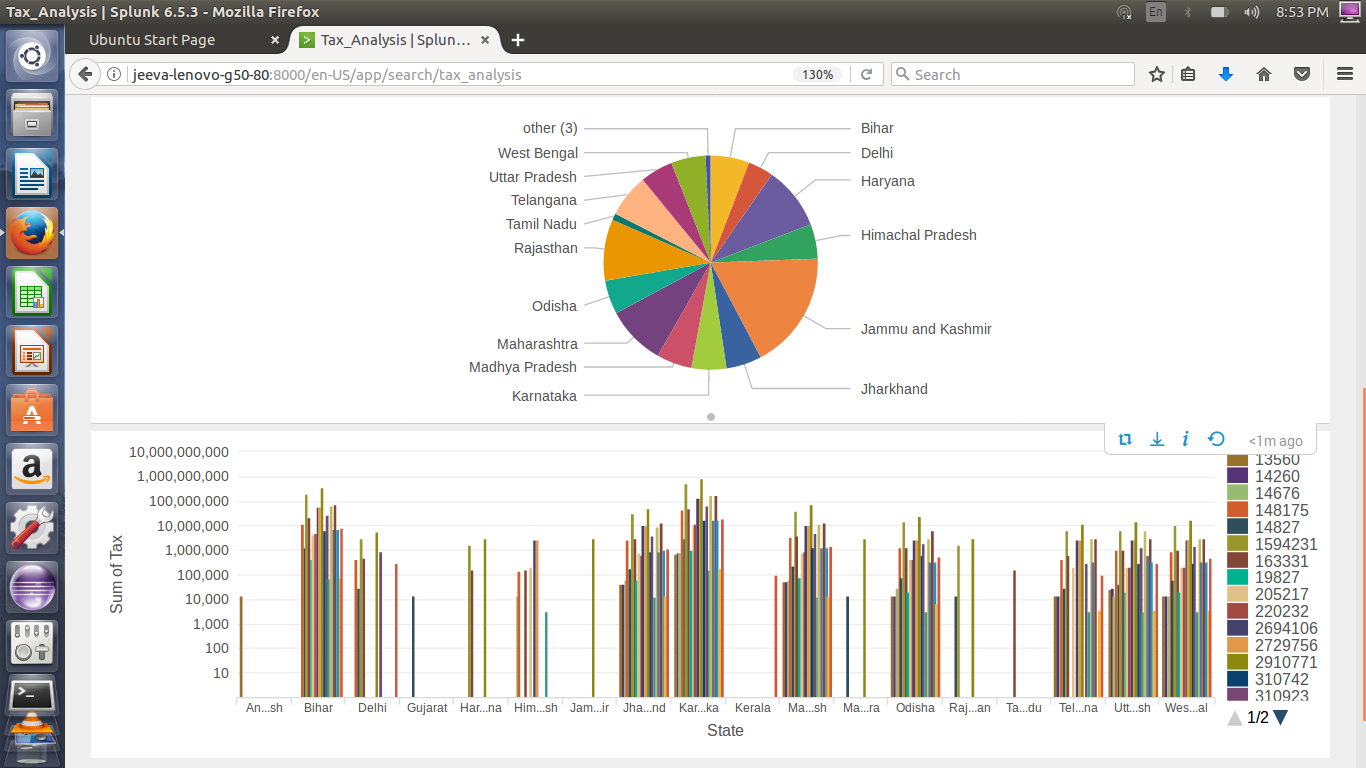


**Visualize the Tax by Individual state:**

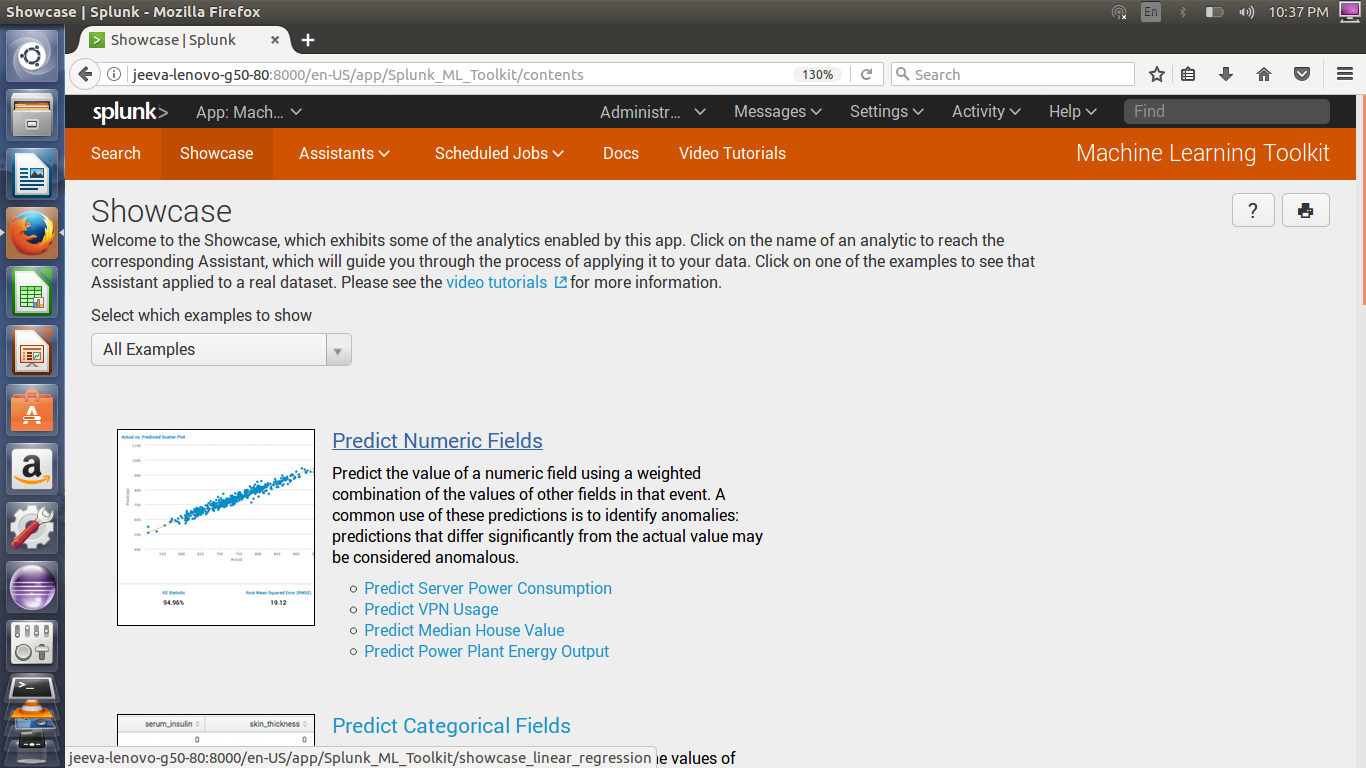
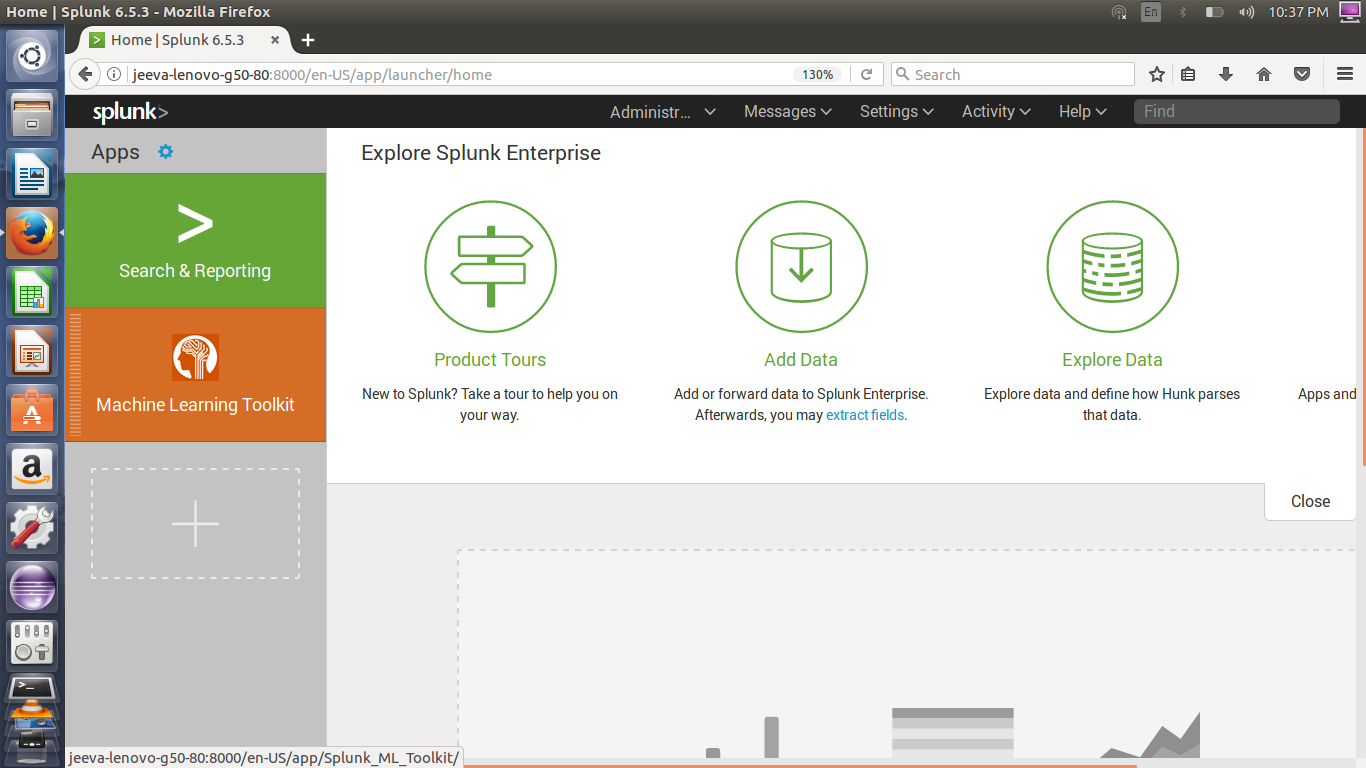


**Save the visualization result in DashBoard:**

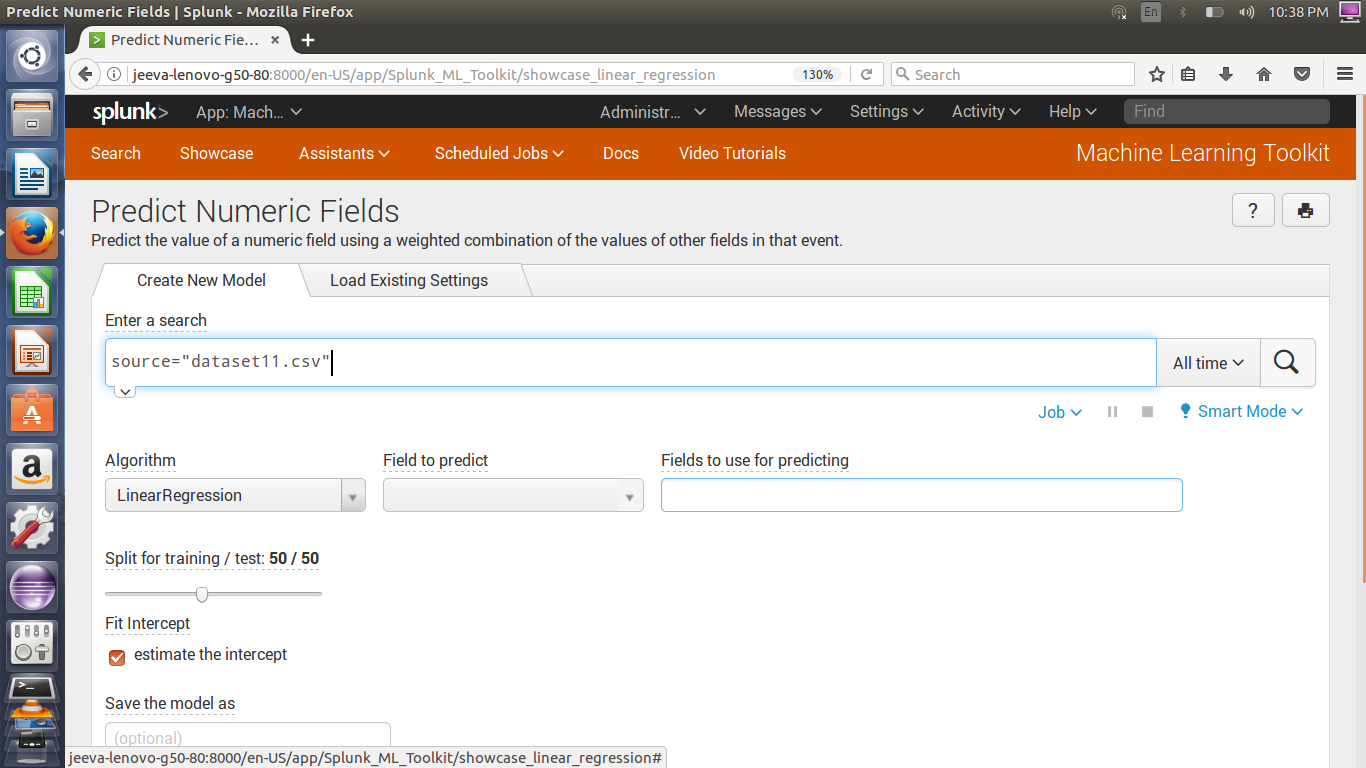




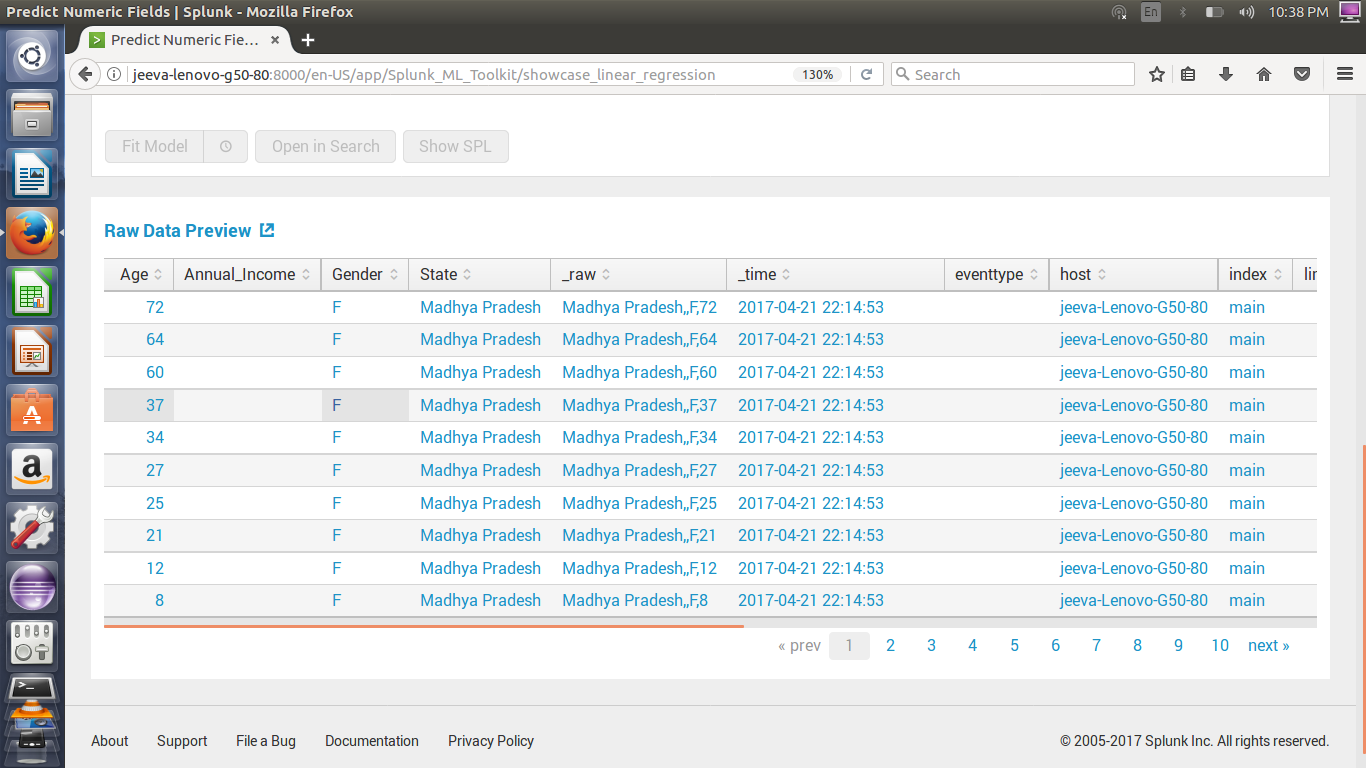
**Choose machine learning toolkit:**



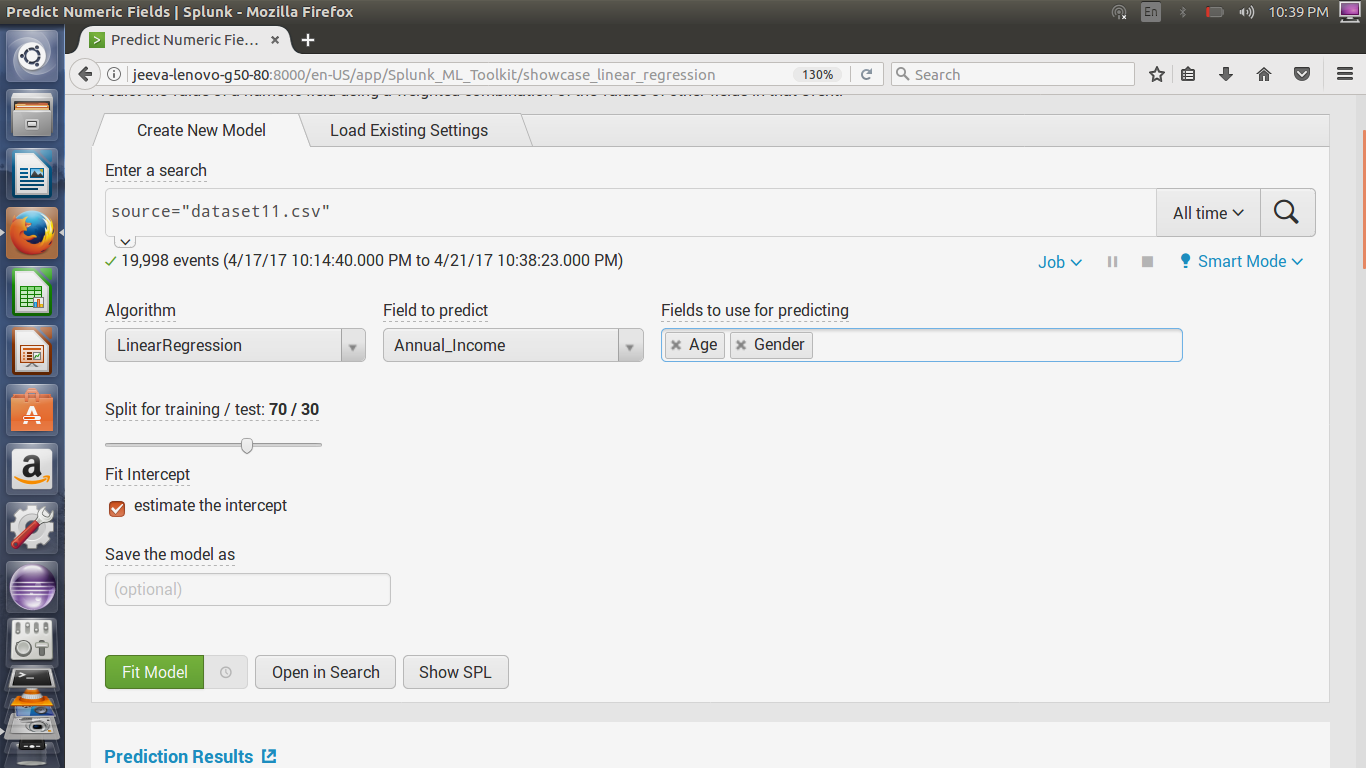
**Search the Uploaded Dataset:**



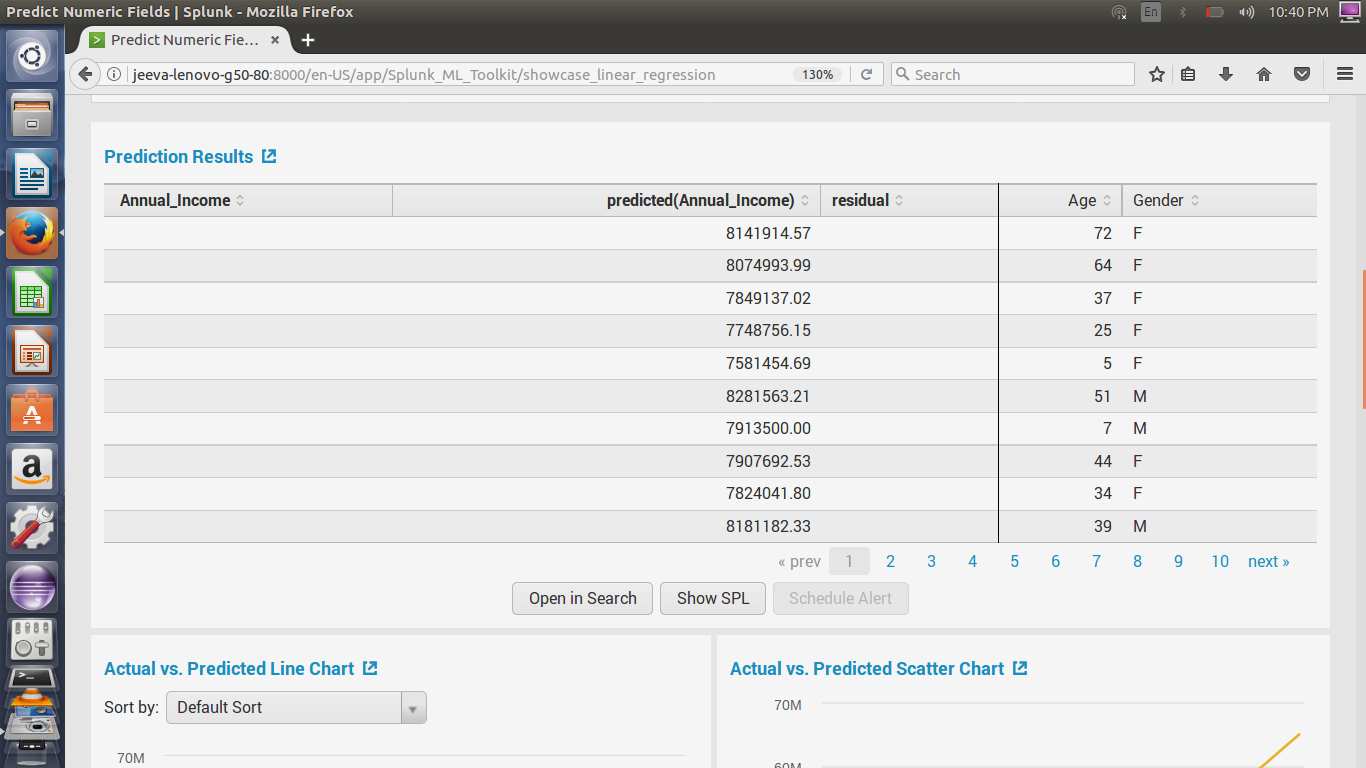
**Preview of uploaded dataset:**



**Choose Algorithm , Prediction field & fields used for prediction:**



**Final predicted result:**



**Comparison of Actual vs. Predicted data:**

