**CENSUS DATA ANALYSIS**

**A PROJECT REPORT**

***Submitted by***

**MAGESH M**

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

***in***

**PROFESSIONAL DIPLOMA**

***in***

**BIG DATA WITH HADOOP**

**NIIT: CHENNAI**

**MARCH: 2017**

**Acknowledgement**

I would like to express my special thanks of gratitude to my Tech Mentor **Ms.Amirtha** as well as our batch Instructor **Mr.Annu Sharma** and our SLT faculty **Mr.Sandeep** who gave me the golden opportunity to do this wonderful project on the topic Analysis of Census Data, which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **NIIT (Chennai)** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & member of **NIIT (Chennai)** especially ***Mrs. Koteeswari*** for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

**Abstract**

Big Data analysis has been a very hot and active research during the past few years. It is getting hard to efficiently execute data analysis task with traditional data warehouse solutions.

Parallel processing platforms and parallel dataflow systems running on top of them are increasingly popular. They have greatly improved the throughput of data analysis tasks. The trade-off is the consumption of more computation resources. Tens or hundreds of nodes run together to execute one task.

This project is about to a financial data analysis on census data by using the Hadoop tools inside the Hadoop ecosystem. The project includes the financial calculation of annual income and calculating income tax for each individual citizens by following US Tax rates. Finally all processed data is exported to MySQL(RDBMS) using Hadoop tools.

**Big Data**

Big Data is a phrase used to mean a massive volume of both structured and unstructured data that is so large it is difficult to process using traditional database and software techniques. In most enterprise scenarios the volume of data is too big or it moves too fast or it exceeds current processing capacity. Big Data may also be called enterprise Big Data. An example of Big Data might be petabytes (1,024 terabytes) or exabytes (1,024 petabytes) of data consisting of billions to trillions of records of millions of people—all from different sources (e.g. Web, sales, customer contact center, social media, mobile data and so on). The data is typically loosely structured data that is often incomplete and inaccessible

**Hadoop**

Hadoop, formally called Apache Hadoop, is an Apache Software Foundation project and open source software platform for scalable, distributed computing. Hadoop can provide fast and reliable analysis of both structured data and unstructured data. Given its capabilities to handle large data sets, it's often associated with the phrase big data.

The Apache Hadoop software library is essentially a framework that allows for the distributed processing of large datasets across clusters of computers using a simple programming model. Hadoop can scale up from single servers to thousands of machines, each offering local computation and storage.

IBM defines Big Data Hadoop as –“Big Data Hadoop is a software project that enables distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with very high degree of fault tolerance. Rather than relying on high-end hardware, the resiliency of these clusters comes from the software's ability to detect and handle failures at the application layer.”

**What is Big Data Analytics?**

If you are convinced with the potential and strong power of big data, and still are a bit obscure on what it can really do for you and for your company then Big Data Analytics is something that you must leverage for profitable business decision making.

Why collect and store zettabytes of data if it cannot be leveraged for analysis in full context? Or if one has to wait for years to get outcomes?

The process of analyzing large structured and unstructured data sets to discover indefinite relations, hidden patterns and any other valuable information that can be leveraged for better business decision making. Big Data Analytics tackles even the most challenging business problems through high-performance analytics. Big data analytics drives innovations by helping organizations make best possible decisions through –high performance data mining, predictive analytics, text mining, social sentiment analysis, text mining, forecasting and optimization. To add on to this, organizations are realizing that distinct properties of deep learning and machine learning are well-suited to address their requirements in novel ways through big data analytics.

**Big Deal Companies are striking with Big Data Analytics**

It’s a Big Deal because, using Big Data one can build better products, offer better services and predict the future better. All this means Big Money. So Big Data is a Big Deal!

* Macy’s , the largest retail store in US runs a daily price check analysis on million items based on demand and inventory. Whenever a neighboring competitor between Los Angeles and New York reduces the prices for various products. Macy’s analytics system adjusts pricing of close to 73 million items based on the availability and demand to pace up with the competition. Macy’s analytics algorithms are designed to adjust prices several time in a day to react in a better manner to local competition. If there is no competitor in the neighborhood, the prices remain unchanged.
* The latest semantic search at Walmart depends on machine learning, text analysis and also synonym mining which helps Walmart produce effective search results. Walmart has witnessed a significant increase of 15% in the number of online shoppers completing their purchase which is some billions of dollars.
* Tesco PLC, one of the largest supermarket chain in UK collected unstructured data points from over 70 million refrigerators which were analyzed to leverage performance efficient. Analytics helped Tesco improve the performance and predict when the refrigerators would need to be serviced. Tesco, furthermore analyzed these data points for predictive maintenance to cut down on the energy costs of the refrigerator.

**Why Hadoop is important?**

Hadoop is changing the perception of handling Big Data especially the unstructured data. Let’s know how Apache Hadoop software library, which is a framework, plays a vital role in handling Big Data. Apache Hadoop enables surplus data to be streamlined for any distributed processing system across clusters of computers using simple programming models. It truly is made to scale up from single servers to a large number of machines, each and every offering local computation, and storage space. Instead of depending on hardware to provide high-availability, the library itself is built to detect and handle breakdowns at the application layer, so providing an extremely available service along with a cluster of computers, as both versions might be vulnerable to failures.

* Store – Big data need to be collected in a seamless repository, and it is not necessary to store in a single physical database.
* Process – The process becomes more tedious than traditional one in terms of cleansing, enriching, calculating, transforming, and running algorithms.
* Access – There is no business sense of it at all when the data cannot be searched, retrieved easily, and can be virtually showcased along the business lines.

*Assumptions and Goals*

* In case of Hardware Failure: A core architectural goal of HDFS is detection of faults and quick, automatic recovery from them.
* Need Streaming Data Access: To run the application HDFS is designed more for batch processing rather than interactive use by users to streaming their data sets.
* Designed for Large Data Sets: HDFS is designed in such a way that it tuned to support large files and it provides big aggregate data bandwidth and scale to many nodes in a single cluster.
* Simple Coherency Model: HDFS applications need a write-once-read-many access model for files. A Map-Reduce application or a web crawler application fits perfectly with this model.
* Portability Issues: HDFS has been designed to be easily portable from one platform to another Across Heterogeneous Hardware and Software Platforms.

**Hadoop Important Glossary**

***HDFS:*** An acronym for “Hadoop Distributed File System”, which breaks large application workloads into smaller data blocks that are replicated and distributed across a cluster of commodity hardware for faster processing.

***Map-Reduce:*** A software framework for easily writing applications that process vast amounts of data (multi-terabyte data-sets) in parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner. Hadoop acts as a platform for executing Map-Reduce.

***YARN:*** a resource manager for Hadoop 2. YARN is short for “Yet another resource negotiator”.

**Some of the Hadoop Ecosystem Tools:**

**Apache Hadoop:** An open source platform that allows for the distributed processing of large data sets across clusters of computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. The platform particularly suited to large volumes of unstructured data such as Facebook comments and Twitter tweets, email and instant messages, and security and application logs.

**Hadoop Common:** Usually only referred to by programmers, Hadoop Common is a common utilities library that contains code to support some of the other modules within the Hadoop ecosystem. When Hive and HBase want to access HDFS, for example, they do so using JARs (Java archives), which are libraries of Java code stored in Hadoop Common.

**Apache Pig:** Apache Pig is a parallel dataflow system runs on top of Apache Hadoop, which is a parallel processing platform. Pig/Hadoop is one of the most popular combinations used to do large scale data processing.

**Apache Hive:** A data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. It allows you to query data using a SQL-like language called HiveQL (HQL).

**HiveQL (HQL):** A SQL like query language for Hadoop used to execute Map-Reduce jobs on HDFS.

**Apache Sqoop:** A tool designed to transfer data between Hadoop and relational databases.

**Apache Oozie:** A workflow engine for Hadoop.

**Apache Flume:** A service for collecting, aggregating, and moving large amounts of log and event data into Hadoop.

**Project Outline**

|  |  |
| --- | --- |
| *Title* | Census Data Analysis |
| *Inputs* | Census Data |
| *Datasets* | a) Age, Education, Marital Status, Gender, Tax Filer Status, Income, Parents, Country of Birth, Citizenship, Weeks Worked.  b) Age, Age Group. |
| *Analysis Relevance* | Employment analysis on given census datasets. |
| *Purpose* | To provide Analyzed Results to Government to help it plan politics to improve economic life of citizens. |
| *Methodology* | Agile. |
| *Technology used* | Hadoop Framework. |

**Project Implementation**

**Assumptions:**

1. Hadoop Cluster is running.
2. Hadoop Ecosystem tools are installed.
3. Census data available in HDFS in JSON format.

**Tools and Framework Used:**

1. Java.
2. MySQL
3. Hadoop Distributed File Systems (HDFS).
4. Yet Another Recourse Negotiator (YARN) architecture.
5. Apache Pig.
6. Apache Hive.
7. Apache Sqoop.

**JOB 1: Conversion of JSON file format to CSV file format.**

*Tool used: Apache Pig.*

**Step 1:** Loading JSON file into Pig Grunt Shell in MapReduce mode.

loadjson = load '/project/sample.dat' using JsonLoader('Age:INT, Education:chararray, MaritalStatus:chararray, Gender:chararray, TaxFilerStatus:chararray, Income:DOUBLE, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:INT');

**Step 2:** Storing Converted csv file into HDFS

store loadjson into '/project/pigdata1' using PigStorage (',');

**RAW INPUT:**

{"Age": 73,"Education": " High school graduate","MaritalStatus": " Widowed","Gender": " Female","TaxFilerStatus": " Nonfiler","Income": 1700.09,"Parents": " Not in universe","CountryOfBirth": " United-States","Citizenship": " Native- Born in the United States","WeeksWorked": 0}

**SAMPLE OUTPUT:**

73, High school graduate, Widowed, Female, Nonfiler,1700.09, Not in universe, United-States, Native- Born in the United States,0

**JOB 2: Finding number of adults, middle aged and senior citizen to find ages between 14 to 64, where government can create employment.**

*Tool used: Apache Pig.*

**Step 1:** Loading the data from HDFS in mapreduce mode.

employment = load '/project/pigdata1/part-m-00000' using PigStorage(',') as (Age:int, Education:chararray, MaritalStatus:chararray, Gender:chararray, TaxFilerStatus:chararray, Income:double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:int);

**Step 2:** Creating partition bags based on ages between 14 and 65 from employment bag.

split employment into employment1 if Age<14, employment2 if (14<=Age and Age<=65), employment3 if Age>65;

**Step 3:** Storing the processed data into HDFS.

STORE employment1 INTO '*/project/pigdata1/*' USING PigStorage(',');

STORE employment2 INTO '*//project/pigdata1/*' USING PigStorage(',');

STORE employment3 INTO '*/project/pigdata1/*' USING PigStorage(',');

**SAMPLE OUTPUT:**

**dump employment1** :

(2, Children, Never married, Male, Nonfiler,476.21, Both parents present, United-States, Native- Born in the United States,0)

(1, Children, Never married, Female, Nonfiler,3288.88, Mother only present, United-States, Native- Born in the United States,0)

(12, Children, Never married, Male, Nonfiler,766.28, Both parents present, United-States, Native- Born in the United States,0)

(13, Children, Never married, Male, Nonfiler,61.23, Mother only present, United-States, Native- Born in the United States,0)

(0, Children, Never married, Male, Nonfiler,1936.23, Both parents present, United-States, Native- Born in the United States,0)

**dump employment2** :

(18, High school graduate, Never married, Female, Nonfiler,2876.96, Not in universe, United-States, Native- Born in the United States,0)

(37, High school graduate, Married-civilian spouse present, Female, Joint both under 65,756.74, Not in universe, United-States, Native- Born in the United States,52)

(29, Associates degree-academic program, Married-civilian spouse present, Male, Joint both under 65,1120.32, Not in universe, United-States, Native- Born in the United States,52)

(43, Some college but no degree, Divorced, Male, Single,1691.39, Not in universe, United-States, Native- Born in the United States,52)

(41, High school graduate, Married-civilian spouse present, Female, Joint both under 65,565.73, Not in universe, United-States, Native- Born in the United States,26)

(38, 11th grade, Married-civilian spouse present, Female, Joint both under 65,1397.09, Not in universe, United-States, Native- Born in the United States,52)

**dump employment3** :

(70, High school graduate, Married-civilian spouse present, Female, Nonfiler,1519.16, Not in universe, United-States, Native- Born in the United States,0)

(74, Some college but no degree, Widowed, Female, Single,236.42, Not in universe, United-States, Native- Born in the United States,0)

(69, 7th and 8th grade, Married-civilian spouse present, Male, Nonfiler,832.43, Not in universe, United-States, Native- Born in the United States,0)

(77, Bachelors degree(BA AB BS), Married-civilian spouse present, Female, Joint both 65+,989.82, Not in universe, United-States, Native- Born in the United States,0)

(79, High school graduate, Married-civilian spouse present, Female, Joint both 65+,868.88, Not in universe, United-States, Native- Born in the United States,0)

(80, High school graduate, Widowed, Male, Nonfiler,490.29, Not in universe, United-States, Native- Born in the United States,0)

**Job 3: Finding Number of Unemployed Graduates.**

*Tool Used: Apache Pig.*

**Step 1:** Loading the Data into Pig Grunt Shell in mapreduce mode.

employment = load '/project/pigdata1/part-m-00000' using PigStorage(',') as (Age:int, Education:chararray, MaritalStatus:chararray, Gender:chararray, TaxFilerStatus:chararray, Income:double, Parents:chararray, CountryOfBirth:chararray, Citizenship:chararray, WeeksWorked:int);

**Step 2:** Filtering employent data with weeksworked as 0 and education as ' Bachelors degree(BA AB BS)';

unemployed = filter employment by WeeksWorked == 0 and Education == ' Bachelors degree(BA AB BS)';

**Step 3:** Grouping the data with the education column.

graduates = group unemployed by $1;

**Step 4:** Counting number of unemployed graduates.

count = foreach countofgrad generate group, COUNT(unemployed.$1);

**OUTPUT:**

**dump count :**

( Bachelors degree(BA AB BS),25)

**Job 4:** Finding number of citizens getting retired in next 3 years in gender wise.

*Tool Used: Apache Hive.*

**Step 1:** Creating database in Hive and using it.

create database project;

use project;

**Step 2:** Adding jar file for Hive process.

add jar /usr/local/hive/hcatalog/share/hcatalog/hive-hcatalog-core-1.2.1.jar;

**Step 3:** Creating tables inside the Project database.

create external table employment (Age int, Education string, MaritalStatus string, Gender string, TaxFilerStatus string, Income double, Parents string, CountryofBirth string, Citizenship string, WeeksWorked int) row format serde 'org.apache.hive.hcatalog.data.JsonSerDe';

**Step 4:** Loading data into the created hive table.

load data local inpath '/home/magesh/Documents/Mainproject/sample.dat' overwrite into table employment;

**Step 5:** Writing hive query to find the citizens retiring after three years.

select Gender, count(\*) from employment WHERE weeksworked != 0 and Age == 62 group by Gender;

**OUTPUT:**

Female 5

Male 7

**Job 5: Calculating PF for employed citizens.**

**Step1: Writing Mapreduce code for calculating PF of all citizen.**

**import java.io.IOException;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.DoubleWritable;**

**import org.apache.hadoop.io.Text;**

**import org.apache.hadoop.mapreduce.Job;**

**import org.apache.hadoop.io.LongWritable;**

**import org.apache.hadoop.mapreduce.Mapper;**

**import org.apache.hadoop.conf.Configuration;**

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;**

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;**

**//73, High school graduate, Widowed, Female, Nonfiler,1700.09, Not in universe,**

**//United-States, Native- Born in the United States,0**

**public class pfcalculation**

**{**

**public static class pfMapper extends Mapper<LongWritable,Text,Text,DoubleWritable>**

**{**

**private DoubleWritable pf1 = new DoubleWritable();**

**long mykey = 0;**

**public void map(LongWritable key, Text value, Context context)throws IOException, InterruptedException**

**{**

**String[] line = value.toString().split(",");**

**double income = (Double.parseDouble(line[5]));**

**double PIA = (income\*0.5);**

**double PF = ((PIA\*12.4)/100);**

**String filer = line[3];**

**Text gen = new Text();**

**gen.set(filer);**

**pf1.set(PF);**

**context.write(gen, pf1);**

**}**

**}**

**public static void main(String[] args) throws Exception**

**{**

**Configuration conf = new Configuration();**

**conf.set("mapred.textoutputformat.separator", ",");**

**Job job = Job.getInstance(conf);**

**job.setJarByClass(pfcalculation.class);**

**job.setJobName("PF Calculation");**

**job.setMapperClass(pfMapper.class);**

**job.setNumReduceTasks(0);**

**job.setOutputKeyClass(Text.class);**

**job.setOutputValueClass(DoubleWritable.class);**

**FileInputFormat.addInputPath(job, new Path(args[0]));**

**FileOutputFormat.setOutputPath(job, new Path(args[1]));**

**System.exit(job.waitForCompletion(true) ? 0 : 1);**

**}**

**}**

**Step2 : Running jar file in hadoop using jar command.**

**Output :**

**Male,47.509359999999994**

**Male,103.59146000000001**

**Male,114.33172**

**Female,67.89186**

**Male,22.5742**

**Female,108.39832**

**Male,129.3878**

**Male,3.7962599999999997**

**Male,159.3617**

**Male,88.70402**

**Male,111.45306000000001**

**Female,101.30242**

**Male,138.89984**

**Female,178.37152000000003**

**Female,94.23442000000001**

**Female,46.917880000000004**

**Project Summary :**

* **Splitting the ages and taking count of people for whom, the government can offer jobs.**
* **Making count of unemployed graduates who are eligible for working, so that the government can create job opportunities.**
* **Making count of people who will get retired in next 3 years, so that government can allocate jobs for future graduates.**
* **Making report on PF amount for all citizens based on their salary.**