globsyn
Taking People To The Next Level

BIG DATA ANALYTICS PROJECT

ON

STARTUP FUNDING USING HADOOP ECOSYSTEM

Group Members

JEET SAHA, Academy Of Technology,

ALLOCATED FORM NO-: 38947

SPANDAN SEN SARMA, Academy Of Technology,

ALLOCATED FORM NO-: 38948

RAHUL ROY, Academy Of Technology,

ALLOCATED FORM NO-:38951

Table of Contents

Acknowledgment

We would like to take this opportunity to express our profound gratitude and deep regards to our faculty **Sir. Titas Roy Chowdhury** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

Each of us project members have worked to our utmost best as to make this project a success in our respective fields. We are grateful to one another for the cooperation during the course of this assignment.

JEET SAHA

SPANDAN SEN SARMA

RAHUL ROY

Project objective

Now a days maximum number of people wants to do business or to become an entrepreneur.The most important part of this is the funding that is the main objective of our project various company has given their details for the funding we need to analysis the data using hadoop ecosystem so the it stands the data are of various fields and we can find the funding project

Main objective is to predict the that is the company is eligible for the start-up funding or not. Using data analytics using HIVE,MAP-REDUCE(HADOOP ECOSYSTEM)

Source Of Data-:

This Data is provided us by our respected Sir Mr Titas Roy Chowdhury(GLOBSYN).

Data Description

- **Main objective** is to predict the that is the company is eligible for the start-up funding or not. Using data analytics using HIVE,MAP-REDUCE(HADOOP ECOSYSTEM).
- **Date:**

This attribute holds the dated heads with all the day,month,year as required to satisfying the start-up

- **Start-up Name:**

This attributes has the records of start-up company's name

- **Industry Vertical:**
  It holds the vertical or the type required in the industry as(Technology,e-commerce) etc.
- **Sub vertical:** It holds the sub vertical of the company as(online platform,cloud solution) etc.
- **City Location:** this data holds the city in which is the start-up.
- **Investors Name:**
  It holds the investors name.
- **Investment type:** the type of investors as (seed funding,Private Equality).
- **AmountInUsd:** it holds the amount in USD($).
- **remarks:** after continue the start-up remarks is the review from the market. Leveled them as a Series A/Series B etc.

MISSING VALUES

The Dataset have many null attribute. Specially the project which is unable to get a funding is full with null value.

sno,date and start-up_name have no null value. But the others have some null. The column 'Remarks' is full of null value. Not

only this but also 'AmountInUSD' column have huge null values.

We simply considered the null valued row as a information which is not getting any funding. In that case the

funding amount is replaced with the value "0". And the investors information is replaced with the "NULL" string.

But sometimes it is also seen that some start-up get funding but although their information is null. We considered them at

the time of Map-reduced, But at the time of Data Analysis we considered as a row which have insufficient information.

Data Processing:-

From the above mentioned 10 columns some columns are not important that's why we simply drop them. The unimportant

columns are 'sno','subvertical','investmenttype','remarks'. 'subvertical' is important if we go for a deep analysis for a specific industryvertical.

Now in these 6 important columns if anyone have a null value then we replace it by 'null' string. And if there 'AmountInUSD' present then we replace it by "1" which means true. And the vacant fields are replaced with "0" that means no funding granted.

So, by getting we can understand that it is a information of a granted project. In case for "0", it represents the information of a rejected project or plan.

We done these all jobs by Map-Reduced program in Java. By running one reducer we get our required output.

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

The importance of big data doesn't revolve around how much data you have, but what you do with it. You can take data from any source and analyze it to find answers that enable 1) cost reductions, 2) time reductions, 3) new product development and optimized offerings, and 4) smart decision making. When you combine big data with high-powered analytics, you can accomplish business-related tasks such as:

- Determining root causes of failures, issues and defects in near-real time.
- Generating coupons at the point of sale based on the customer's buying habits.
- Recalculating entire risk portfolios in minutes.
- Detecting fraudulent behavior before it affects your organization.

Hadoop

Hadoop is an open-source framework to store and process Big Data in a distributed environment. It contains two modules, one is Map-Reduce and another is Hadoop Distributed File System (HDFS).

- Map-Reduce: It is a parallel programming model for processing large amounts of structured, semi-structured, and unstructured data on large clusters of commodity hardware.
- HDFS:Hadoop Distributed File System is a part of Hadoop framework, used to store and process the data-sets. It provides a fault-tolerant file system to run on commodity hardware.

The Hadoop ecosystem contains different sub-projects (tools) such as Sqoop, Pig, and Hive that are used to help Hadoop modules.

- Sqoop: It is used to import and export data to and from between HDFS and RDBMS.
- Pig: It is a procedural language platform used to develop a script for Map-Reduce operations.
- Hive: It is a platform used to develop SQL type scripts to do Map-Reduce operations.

Note: There are various ways to execute Map-Reduce operations:

- The traditional approach using Java Map-Reduce program for structured, semi-structured, and unstructured data.
- The scripting approach for Map-Reduce to process structured and semi structured data using Pig.
- The Hive Query Language (HiveQL or HQL) for Map-Reduce to process structured data using Hive.

Hive

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic Map-Reduce.

Hive is not

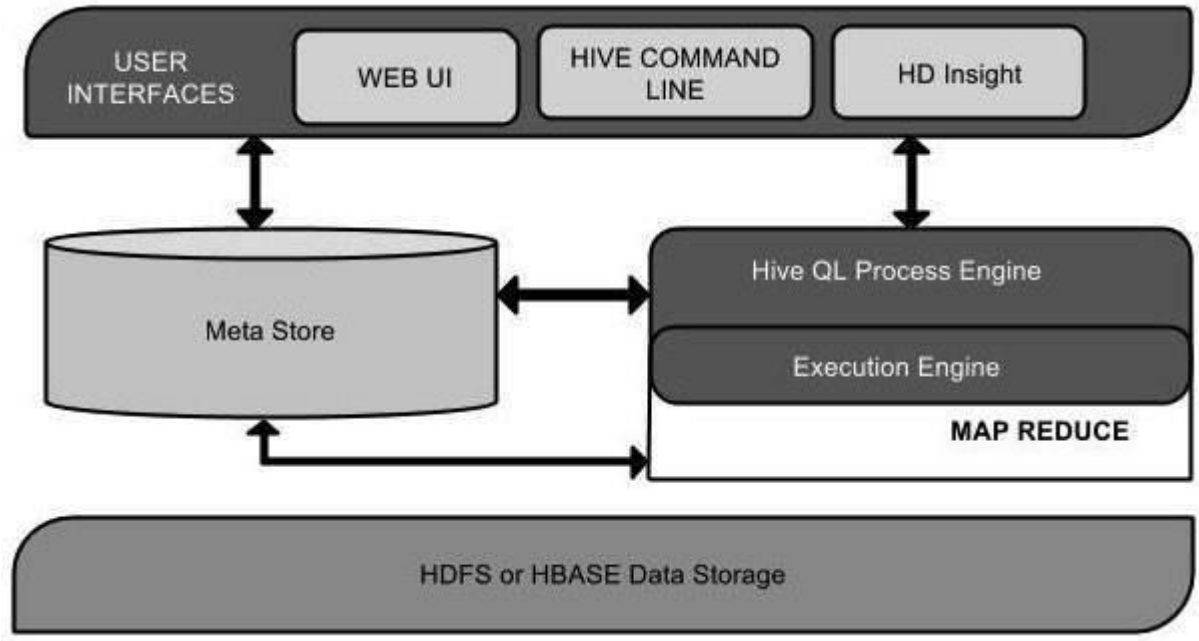
- A relational database
- A design for Online Transaction Processing (OLTP)
- A language for real-time queries and row-level updates

Features of Hive

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Architecture of Hive

he following component diagram depicts the architecture of Hive:



This component diagram contains different units. The following table describes each unit:

Unit Name	Operation
User Interface	Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight (In Windows server).

Meta Store	Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.
HiveQL Process Engine	HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.
Execution Engine	The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.
HDFS or HBASE	Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

Pig

Apache Pig is a high level data flow platform for execution Map Reduce programs of Hadoop. The language for Pig is pig Latin.

The Pig scripts get internally converted to Map Reduce jobs and get executed on data stored in HDFS. Every task which can be achieved using PIG can also be achieved using java used in Map reduce.

Usage of Pig

Let's see the 3 usage of Pig technology.

1) Ease of programming

Writing complex java programs for map reduce is quiet tough for non programmers. Pig makes this process easy. In pig, the queries are converted to map reduce internally.

2) Optimization opportunities

The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.

3) Flexibility

User defined function are written in which the user can write their own logic to execute over the data set.

Pig Data Types

Apache Pig supports many data types. A list of Apache Pig Data Types with description and examples are given below.

Type	Description	Example
Int	Signed 32 bit integer	2
Long	Signed 64 bit integer	15L or 15l
Float	32 bit floating point	2.5f or 2.5F
Double	32 bit floating point	1.5 or 1.5e2 or 1.5E2
charArray	Character array	hello javatpoint
byteArray	BLOB(Byte array)	
tuple	Ordered set of fields	(12,43)
bag	Collection f tuples	{(12,43),(54,28)}
map	collection of tuples	[open#apache]

MapReduce

MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.

A MapReduce program is composed of a map procedure (or method), which performs filtering and sorting (such as sorting students by first name into queues, one queue for each name), and a *reduce* method, which performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The "MapReduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

The model is a specialization of the *split-apply-combine* strategy for data analysis. It is inspired by the map and reduce functions commonly used in functional programming,although their purpose in the MapReduce framework is not the same as in their original forms. The key contributions of the MapReduce framework are not the actual map and reduce functions (which, for example, resemble the 1995 Message Passing Interface standard's *reduce* and *scatter* operations), but the scalability and fault-tolerance achieved for a variety of applications by optimizing the execution engine. As such, a single-threaded implementation of MapReduce will usually not be faster than a traditional (non-MapReduce) implementation; any gains are usually only seen with multi-threaded implementations.The use of this model is beneficial only when the optimized distributed shuffle operation (which reduces network communication cost) and fault

tolerance features of the MapReduce framework come into play. Optimizing the communication cost is essential to a good MapReduce algorithm.

MapReduce libraries have been written in many programming languages, with different levels of optimization. A popular open-source implementation that has support for distributed shuffles is part of Apache Hadoop. The name MapReduce originally referred to the proprietary Google technology, but has since been genericized. By 2014, Google was no longer using MapReduce as their primary big data processing model, and development on Apache Mahout had moved on to more capable and less disk-oriented mechanisms that incorporated full map and reduce capabilities.

A MapReduce framework (or system) is usually composed of three operations (or steps):

1. Map: each worker node applies the `map` function to the local data, and writes the output to a temporary storage. A master node ensures that only one copy of redundant input data is processed.
2. Shuffle: worker nodes redistribute data based on the output keys (produced by the `map` function), such that all data belonging to one key is located on the same worker node.
3. Reduce: worker nodes now process each group of output data, per key, in parallel.

Uses

MapReduce is useful in a wide range of applications, including distributed pattern-based searching, distributed sorting, web link-graph reversal, Singular Value Decomposition,web access log stats, inverted index construction, document clustering, machine learning, and statistical machine translation. Moreover, the MapReduce model has been adapted to several computing environments like multi-core and many-core systems, desktop grids, multi-cluster, volunteer computing environments, dynamic

FUTURE SCOPE OF INPROVEMENTS:-

- we can see which verticals or fields are appropriate for a start up. According to this we can invest our money.

- As here we see the changing of funding in no. of projects in every year. It help us understand what should be the future funding.

- We can also use predicting methods on it by Naive Bays or other. By this predicting model we think or make diction on a particular project that is should be funded or not. And what is chance that it survives in the market.

- By the city location we observe that we city are accepted more new plans or project. By this we also plan for a new start-up in other city.

- By the investor's information we can also know that which company in this market are the most interested in bringing the new plan or project or product in the market.

- It is less important but we also estimate the amount need for a start-up in a specific industry vertical.

CODE

*Mapper:

```
package org;

import java.io.IOException;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class SUMapper extends Mapper<LongWritable, Text, NullWritable, Text>{

    private NullWritable outkey=NullWritable.get();
    private Text outvalue=new Text();
    @Override

    protected void map(LongWritable key, Text value,Context context)
            throws IOException, InterruptedException {

        String line=value.toString();
        String[] f = line.split("'");
        String sl="";
        for(int i=0;i<f.length;i++)
        {
            if(i%2!=0)
            {
                f[i]=f[i].replaceAll(",", "");
            }
        }
        if(f.length==1)
            sl += f[0];
        for(int i=0;i<f.length-1;i++)
        {
            sl += f[i];
        }
        line = sl;
```

```
        String[] values = line.split(",");
        String writableString = "";
        ArrayList<String> al = new ArrayList<String>();
        for (int i=0;i<values.length;i++) {
            if (values[i]==null || values[i].length()==0) {
                al.add("null");
            }
            else {
                al.add(values[i]);
            }
        }
        for (String s : al){
            writableString += s + ",";
        }
        line = writableString.substring(0,writableString.length() - 1);
        String field[]=line.split(",");
        if(field.length<9)
        {
            int k= 9-(field.length);
            for(int i=0;i<k;i++)
            {
                if(i==k-1)
                    line += ",0";
                else
                    line += ",NULL";
            }
        }
        String[] fields=line.split(",");
        fields[1]=fields[1].replaceAll("\\.", "/");
        String[] s3=fields[1].split("/");
        fields[1]=s3[s3.length-1];
        String n = fields[0]+","+fields[1]+" , "+fields[2]+" , "+fields[3]+" , "
        +fields[5]+" , "+fields[6]+","+fields[7];
        if(fields[8].equalsIgnoreCase("0"))
            n += " , "+"0";
        else
            n += " , "+"1";
        outvalue.set(n);
        context.write(outkey,outvalue);
    }
}
```

*Driver:

```
package org;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class SUDriver {

    public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException {
        Configuration conf= new Configuration();
        Job job=Job.getInstance(conf);

        job.setJarByClass(SUDriver.class);
        job.setMapperClass(SUMapper.class);
        job.setOutputKeyClass(NullWritable.class);
        job.setOutputValueClass(Text.class);

        job.setNumReduceTasks(1);
        FileInputFormat.addInputPath(job,new Path("startup"));
        FileOutputFormat.setOutputPath(job,new Path("startup_output"));


        job.waitForCompletion(true);

    }

}
```

*HIVE Programming:

```
                    project

project by hive=================

mkdir hivesummer18
cd hivesummer18
hive
create database project;
show databases;
use project;
set hive.cli.print.current.db=true;

hadoop fs -cp /user/edureka/startup_output/part-r-00000 /user/edureka/hivedata1/startup;


create table startup(SNo number(4),Date number(4) , StartupName varchar2(30), IndustryVertical varchar2(50) , CityLocation varchar2(20) ,
InvestorsName varchar2(50),InvestmentType varchar2(50) , AmountInUSD number(1))
row format delimited
fields terminated by ','
location '/user/edureka/hivedata1/startup';

q1=======

create table q11 as select date,count(date) as count_all,IndustryVertical as ver from startup1 where sno!='SNo' group by date,IndustryVertical;

create table q12 as select date,count(date) as count_1,IndustryVertical as ver from startup1 where sno!='SNo' and amountinusd=' 1' group by date,IndustryVertical;

create table q1 as select a.date, a.count_all, b.count_1, a.ver from q11 a join q12 b on (a.ver = b.ver and a.date = b.date);

select * from q1 where ((count_1/count_all)>0.5 and count_1>10 and ver !=' null ') group by ver,date,count_1,count_all;


 Consumer Internet  2016    305 539
 Consumer Internet  2017    156 233
 ECommerce      2016    21  38
 ECommerce      2017    11  15
 Education      2016    14  19
 Technology         2016    120 190
 Technology         2017    69  123
 eCommerce      2016    80  125
 eCommerce      2017    32  45
```

```
q2=======


create table q21 as select citylocation,count(citylocation) from startup1 where sno!='SNo' group by citylocation;
create table q22 as select citylocation,count(citylocation) from startup1 where sno!='SNo' and amountinusd=' 1' group by citylocation;

create table q2 as select a.citylocation, a.count_all, b.count_1 from q21 a join q22 b on (a.citylocation = b.citylocation);

select * from q2 where ((count_1/count_all)>0.5 and count_1>10 and citylocation !=' null ');

 Ahmedabad  35  25
 Bangalore  625 415
 Chennai    66  49
 Gurgaon    240 169
 Hyderabad  75  56
 Mumbai     442 306
 New Delhi  381 223
 Noida      78  47
 Pune       84  55


q3==========


create table q31 as select IndustryVertical,count(IndustryVertical) as count_all from startup1 where sno!='SNo' group by IndustryVertical;
create table q32 as select IndustryVertical,count(IndustryVertical) as count_1 from startup1 where sno!='SNo' and amountinusd=' 1' group by IndustryVertical;

create table q3 as select a.industryvertical, a.count_all, b.count_1 from q31 a join q32 b on (a.industryvertical = b.industryvertical);

select industryvertical,count_1/count_all from q3 where ((count_1/count_all)>0.5 and count_1>10 and industryvertical !=' null ');

 Consumer Internet  772 461
 ECommerce      53  32
 Education      20  15
 Food & Beverage    19  12
 Logistics      24  16
 Technology         313 189
 eCommerce      171 113


q4==========


create table q41 as select investorsname,count(investorsname) as count_all from startup1 where sno!='SNo' group by investorsname;
create table q42 as select investorsname,count(investorsname) as count_1 from startup1 where sno!='SNo' and amountinusd=' 1' group by investorsname;

create table q4 as select a.investorsname, a.count_all, b.count_1 from q41 a join q42 b on (a.investorsname = b.investorsname);

select investorsname,count_1/count_all from q4 where ((count_1/count_all)>0.5 and count_1>5 and investorsname !=' null ');


 Accel Partners         9   7
 Brand Capital          10  7
 Group of Angel Investors   15  14
 Indian Angel Network       25  13
 Info Edge (India) Ltd      8   8
 Kalaari Capital        16  14
 SAIF Partners          9   9
 Sequoia Capital        14  12
 Tiger Global           7   7
 Trifecta Capital       6   6
 Undisclosed            9   8
 Undisclosed Investor       10  8
 Undisclosed Investors      33  24
 Undisclosed investor       9   7
 Undisclosed investors      27  21
 undisclosed investors      11  11
```