

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…………………………………………………………
* Project Objective……………………………………………..................
* Source of data……………………………………………………………….
* Data Description…………………………………………………………….
* Missing Values……………………...............................................
* Data processing………………………………………………………………
* Future Scope of Improvements……………………………………….
* Code……………………………………………………………………………….

**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](https://en.wikipedia.org/wiki/Programming_model" \o "Programming model) and an associated implementation for processing and generating [big data](https://en.wikipedia.org/wiki/Big_data" \o "Big data) sets with a [parallel](https://en.wikipedia.org/wiki/Parallel_computing" \o "Parallel computing), [distributed](https://en.wikipedia.org/wiki/Distributed_computing" \o "Distributed computing) algorithm on a [cluster](https://en.wikipedia.org/wiki/Cluster_(computing)" \o "Cluster (computing)).

A MapReduce program is composed of a [map](https://en.wikipedia.org/wiki/Map_(parallel_pattern)" \o "Map (parallel pattern)) [procedure](https://en.wikipedia.org/wiki/Procedure_(computing)" \o "Procedure (computing)) (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](https://en.wikipedia.org/wiki/Marshalling_(computer_science)" \o "Marshalling (computer science)) 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](https://en.wikipedia.org/wiki/Redundancy_(engineering)" \o "Redundancy (engineering)) and [fault tolerance](https://en.wikipedia.org/wiki/Fault-tolerant_computer_system" \o "Fault-tolerant computer system).

The model is a specialization of the *split-apply-combine* strategy for data analysis. It is inspired by the [map](https://en.wikipedia.org/wiki/Map_(higher-order_function)" \o "Map (higher-order function)) and [reduce](https://en.wikipedia.org/wiki/Fold_(higher-order_function)" \o "Fold (higher-order function)) functions commonly used in [functional programming](https://en.wikipedia.org/wiki/Functional_programming" \o "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](https://en.wikipedia.org/wiki/Message_Passing_Interface" \o "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](https://en.wikipedia.org/wiki/Single-threaded" \o "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](https://en.wikipedia.org/wiki/Multi-threaded" \o "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](https://en.wikipedia.org/wiki/Library_(software)" \o "Library (software)) have been written in many programming languages, with different levels of optimization. A popular [open-source](https://en.wikipedia.org/wiki/Open-source_software" \o "Open-source software) implementation that has support for distributed shuffles is part of [Apache Hadoop](https://en.wikipedia.org/wiki/Apache_Hadoop" \o "Apache Hadoop). The name MapReduce originally referred to the proprietary [Google](https://en.wikipedia.org/wiki/Google" \o "Google) technology, but has since been [genericized](https://en.wikipedia.org/wiki/Generic_trademark" \o "Generic trademark). By 2014, Google was no longer using MapReduce as their primary [big data](https://en.wikipedia.org/wiki/Big_data" \o "Big data) processing model, and development on [Apache Mahout](https://en.wikipedia.org/wiki/Apache_Mahout" \o "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](https://en.wikipedia.org/w/index.php?title=Worker_node&action=edit&redlink=1" \o "Worker node (page does not exist)) applies the map function to the local data, and writes the output to a temporary storage. A [master node](https://en.wikipedia.org/w/index.php?title=Master_node&action=edit&redlink=1" \o "Master node (page does not exist)) 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

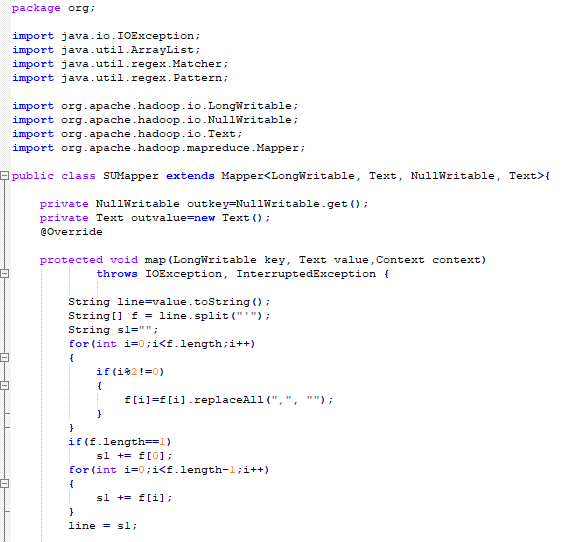
MapReduce is useful in a wide range of applications, including distributed pattern-based searching, distributed sorting, web link-graph reversal, Singular Value Decomposition,web access log stats, [inverted index](https://en.wikipedia.org/wiki/Inverted_index" \o "Inverted index) construction, [document clustering](https://en.wikipedia.org/wiki/Document_clustering" \o "Document clustering), [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning), and [statistical machine translation](https://en.wikipedia.org/wiki/Statistical_machine_translation" \o "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

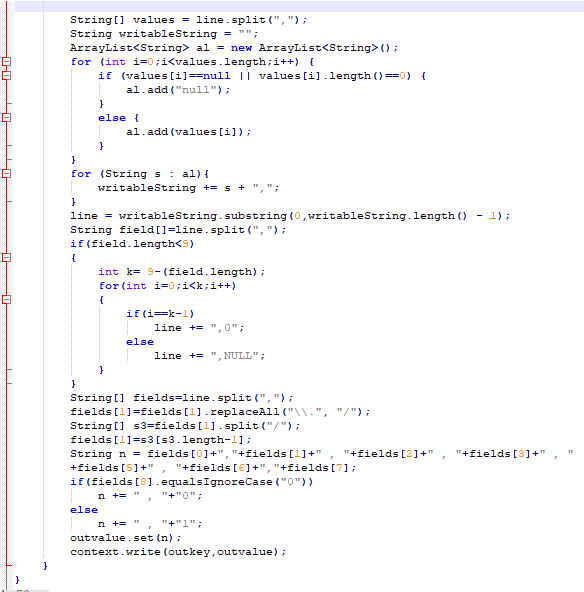
**FUTURE SCOPE OF INPROVEMENTS:-**

* 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.

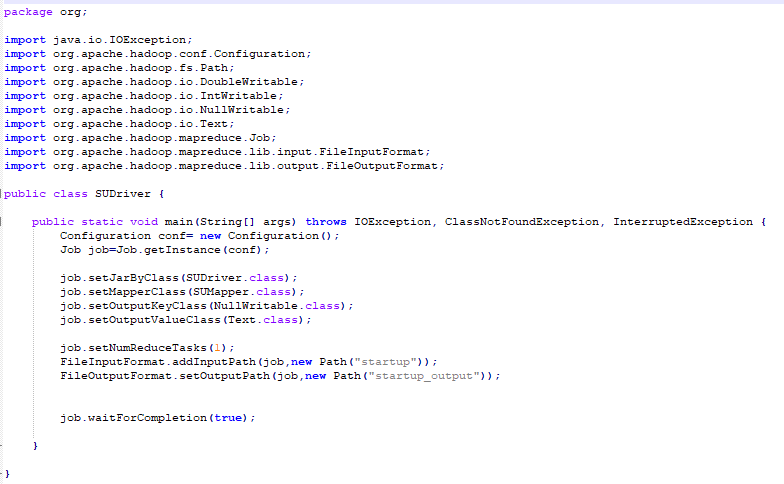
**CODE**

**\*Mapper:**

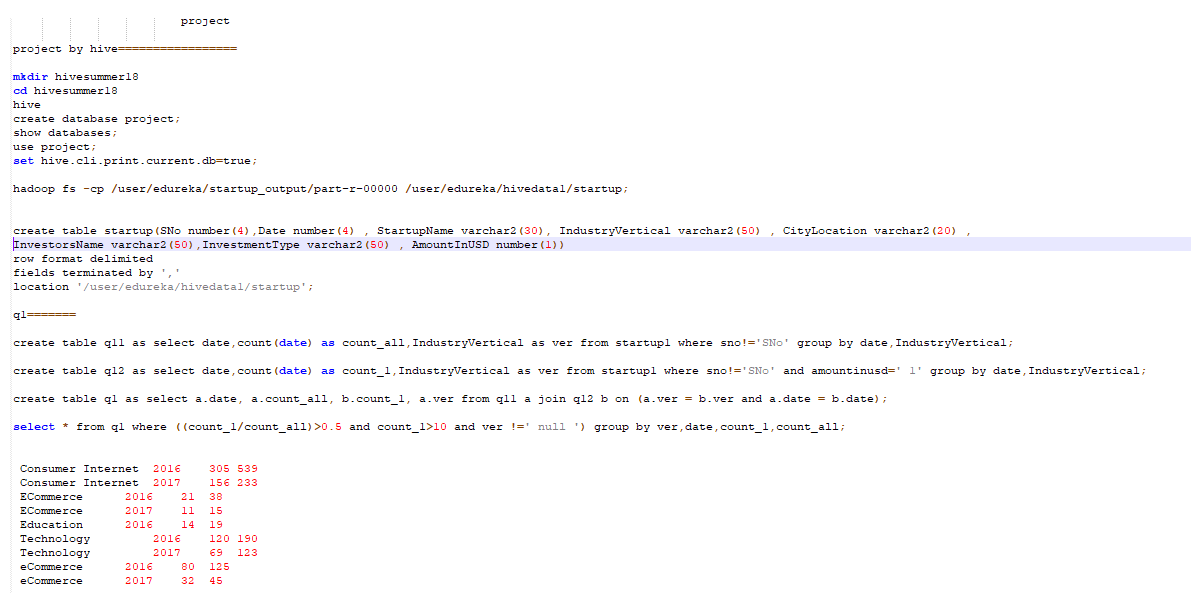


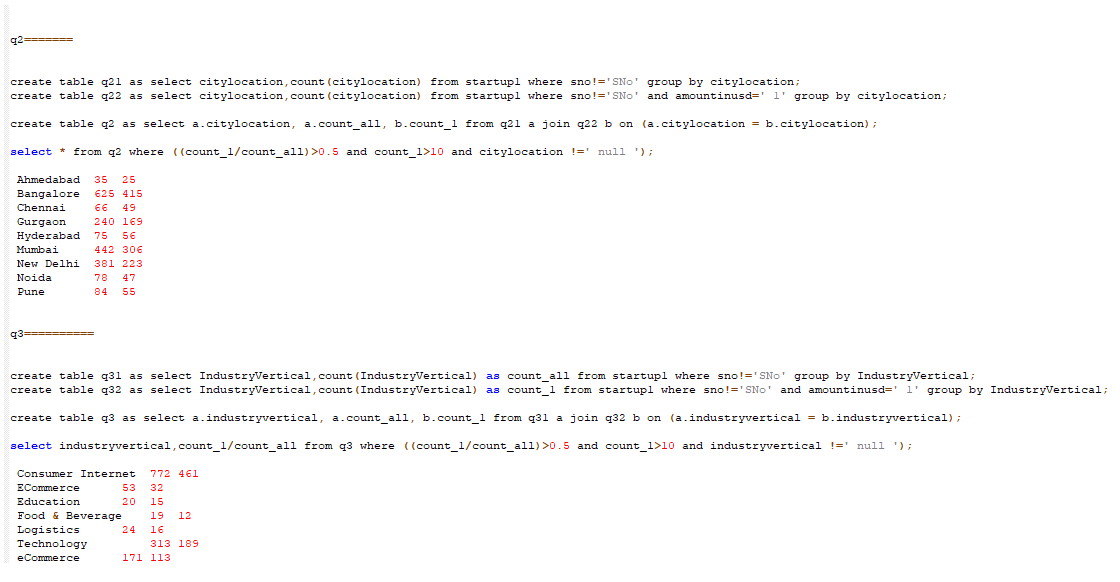


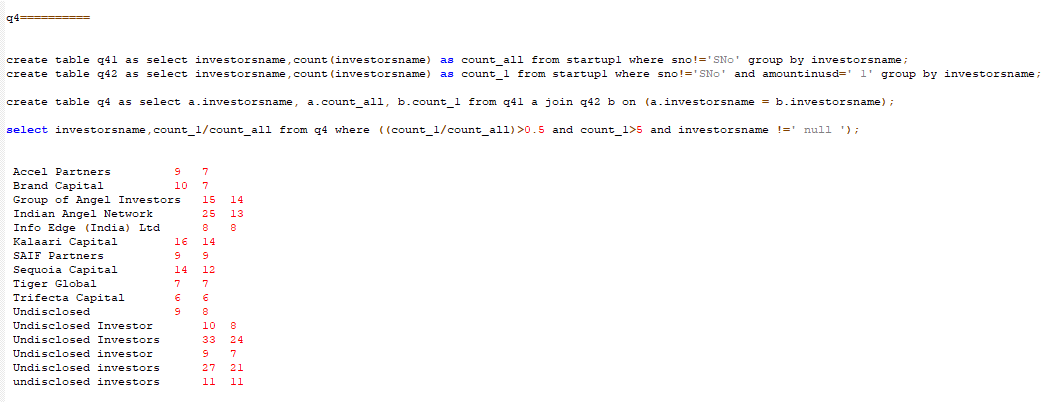
**\*Driver:**



**\*HIVE Programming:**

****

****

****