

CHAPTER-1

INTRODUCTION

1.1. OVERVIEW

In this digital world, everything becomes data and it is highly challenging to store them and make use of it. We produce 2.5 quintillion bytes of data everyday through the staggering array of digital connections that link people, objects and devices. Every email, text, post, online search, app interaction, card transaction, and doctor's visit contribute to these huge volume of data, in greater amounts than ever before. It's estimated that 90 percent of data in existence was generated in the last two years. All these data are collectively called by a term called 'BIG DATA'. Big Data is data that contains greater variety arriving in increasing volumes and with ever-higher velocity. These massive volumes of data can be used to address various business problems you wouldn't have been able to tackle before. So it is particularly important to store these huge volume of data and to process them. Big Data is a buzzword that describes voluminous data having structured and unstructured formats. The entirety of data is difficult to process using traditional methods of relational database . Big Data is a term used to define huge amount of data that is captured and stored over distributed file systems to reduce the overhead.

After processing, the data is managed for analysis of productive and useful information. Therefore the term Big Data is computationally managing the voluminous data and its processing using several tools. It has been found that the total volume of data that was present in the past has been doubled in the last few years. Petabyte and Zettabytes have become common terms now a days. Big data ensures to maintain the accuracy and performance efficiency over the variety of

data provided from several organizations, researchers, institutions and consumers by ensuring properties of Volume, Value, Velocity, Variety and Veracity.

The term BIG DATA in the literature comprises of the 5V's of Big Data which include Volume, Variety, Velocity, Value and Veracity. Big Data can be understood better based on its characteristics.

Volume: The name Big Data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, 'Volume' is one characteristic which needs to be considered while dealing with Big Data.

Variety: Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications. Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

Velocity: The term 'velocity' refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data.

Value: It is the most important V of the Big Data. The term Value referring to the worth of the data being extracted. Having endless amounts of data is one thing, but unless it can be turned into value it is useless.

Veracity: Veracity is the quality or trustworthiness of the data. It refers how accurate is all this data. Accurate data provides high accuracy results. With all these advantages, Big Data is used to find solution for all business problems. Now it exists in almost all the departments.

Banks have a problem to identify the behavior/nature of their customers. Based on the current salary and assets owned by the customers may not help the banks to classify their customers accurately. Customers with valuable assets and good salary may show reluctance to payback their loans and the customers with not much valuable and average salary may payback their loans properly. This can be solved by handling past transactions and details of the customers which automatically leads to Big Data handling. By understanding the importance of Big Data in banking sector, the banks started to invest in Big Data to find the solutions for their business problems.

1.2. LITERATURE SURVEY

- [1] **Ivanilton Polato a,b,n , Reginaldo Ré b , Alfredo Goldman a , Fabio Kon: A comprehensive view of Hadoop research—A systematic literature review, Journal of Network and Computer Applications 46(2014)**

In recent years, the valuable knowledge that can be retrieved from petabyte scale dataset known as Big Data led to the development of solutions to process information based on parallel and distributed computing. Lately Apache Hadoop has attracted strong attention due to its applicability to Big Data processing.

The support of Hadoop by the research community has provided the development of new features to the framework. Recently the number of publications in journals and conferences about hadoop has increased consistently, which makes it difficult for researchers to comprehend the full body of research and areas that require further investigation.

A systematic literature review to assess research contributions to Apache Hadoop. The objective was to identify gaps, providing motivation for new

research and outline collaborations to Apache Hadoop and its ecosystem, classifying and quantifying the main topics addressed in the literature.

The analysis led to some relevant conclusions such as, many interesting solutions developed in the studies were never incorporated into the framework, most publications lack sufficient formal documentation of the experiments conducted by authors, hindering their reproducibility.

Finally the systematic review presented in the paper demonstrates that Hadoop has evolved into a solid platform to process large datasets but it is able to spot promising areas and suggest topics for future research within the framework.

[2] PrathyushaRani Merla, Yiheng Liang: Data Anlaysis Using Hadoop Map Reduce Environment, IEEE International Conference on Big Data (BIGDATA), 2017.

The project deals with analysis of YouTube data using Hadoop Map Reduce framework on a cloud platform AWS. YouTube API provides the necessary interface/methods to download the data from YouTube data center. The reporting API supports applications that can retrieve and store bulk reports, then provide tools to filter, sort and mine the data. The Analytics API supports targeted, real time queries to generate custom reports in response to user interaction.

The data from YouTube are collected from client and are stored in the Hadoop Distributed File System in CSV format. HDFS allows applications to access data from it with the help of YARN. The name node in HDFS monitors access to the files stored in it. The data nodes allows to do read/write activities of the file and contains the data and metadata of the films.

The MapReduce program obtains the data for processing from the HDFS. The key significance of using the MapReduce framework is that it offers scalability and a cost effective solution to the problem. The result of the analysis are shown in graphical format using pie charts and bar graphs. The top five trending video categories are listed in a pie chart with different colors for each different category. All these statistics helps in understanding the data analysis of YouTube in simpler format.

The project primarily intends in showing how large datasets like YouTube can be analyzed using Hadoop Ecosystem. The results of the project can be transformed into decisions which has good impact.

[3] SFeng Li, Beng Chin Ooi, M. Tamer Ozsu, and Sai Wu: Distributed data management using MapReduce, ACM Comput. Surv. 46, 3, Article 31 (January 2014)

With the increasing amount of data and the availability of high-performance and relatively low-cost hardware, database systems have been extended and parallelized to run on multiple hardware platforms to manage scalability. Recently a new distributed data processing framework called MapReduce was proposed whose fundamental idea is to simplify the parallel processing using a distributed computing platform that offers only two interfaces: map and reduce. Programmers implement their own map and reduce functions, while the system is responsible for scheduling and synchronizing the map and reduce tasks.

MapReduce is a simplified parallel data processing approach for execution on a computer cluster. Its programming model consists of two user defined functions, map and reduce.

The inputs of map function are a set of key/value pairs. When a MapReduce job is submitted to the system, the map tasks are started on the compute nodes and each map task applies the map function to every key/value pair $(k1, v1)$ that is allocated to it. Zero or more intermediate key/value pairs $(list(k2, v2))$ can be generated for the same input key/value pair. These intermediate results are stored in the local file system and sorted by the keys. After all the map tasks complete, the MapReduce engine notifies the reduce tasks to start their processing. The reducers will pull the output files from the map tasks in parallel and merge-sort the files obtained from the map tasks to combine the key/value pairs into a set of new key/value pairs $(k2, list(v2))$, where all values with the same key $k2$ are grouped into a list and used as the input for the reduce function. The reduce function applies the user-defined processing logic to process the data. The results, normally a list of values, are written back to the storage system.

Hadoop is currently the most popular open-source MapReduce implementation. It is written in Java and has been tested in Yahoo's cluster. Although Hadoop can be deployed on different storage systems, the released Hadoop package includes HDFS as the default storage system. The two modules of Hadoop, namely, HDFS and the processing engine, are loosely connected. They can either share the same set of compute nodes or be deployed on different nodes. In HDFS, two types of nodes are created: the *name node* and *data node*. The name node records how data are partitioned and monitors the status of data nodes in HDFS. Data imported into HDFS are split into equal-size chunks and the name node distributes the data chunks to different data nodes, which store and manage the chunks assigned to them. The name node also acts as the dictionary server, providing partitioning information to applications that search for a specific chunk of data.

As a massively parallel processing framework, MapReduce is well recognized for its scalability, flexibility, fault tolerance, and a number of other

attractive features. In particular, it facilitates parallelization of a class of applications, commonly referred to as embarrassingly parallelizable. However, as has been commonly acknowledged, MapReduce has not been designed for large-scale complex data management tasks. For example, the original framework does not provide high-level language support that is familiar to and expected by database users; consequently, users have to individually develop various processing logics and programs. It also does not have built-in indexing and query optimization support required for database queries. This has naturally led to a long stream of research that attempts to address the lack of database functionality.

[4] Arushi Jaina, Vishal Bhatnagara Ambedkar: Crime Data Analysis Using Pig with Hadoop, International Conference on Information Security & Privacy, 2015

Hadoop is a framework for the analysis and transformation of very large data sets using the Map Reduce paradigm. An important characteristic of Hadoop is the splitting of data and computation across thousands of hosts and running applications in parallel close to their data. Hadoop accomplish this by HDFS and Map Reduce. Pig is an apache open source project. It runs on Hadoop by making use of both HDFS and Map Reduce. There are two main components for Pig. First component Pig Latin is the parallel dataflow language which is designed in such a way to fit between the SQL and the Map Reduce. Pig Latin enables the use to define the reading, processing, storing the data in parallel. Pig Latin script explicates a directed acyclic graph, where data flows are represented as edges and operators are represented as nodes. The second component is the run time environment in which Pig Latin programs are executed.

The paper illustrates the application of Pig Latin over Hadoop framework provides easy implementation of mapreduce over Hadoop framework. Data flow language helps in converting query language into map reduce algorithm.

1.3. OBJECTIVE

The main objective of this project is to develop a system to classify the customers based on their previous transactions, not only on their current salary and assets. This system will categorize the customers into low, medium and high risk customers based on their behaviour. It will help the banks to decide if the holder can be approved of the credit card and loans and enables them to do marketing based on their customers. The main aim of this project is to identify their customers and schedule their plans based on them.

CHAPTER -2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

According to the RBI statistics, a bank handles about 8,00,00,000 transactions in a month. Moreover the number keeps increasing every day. Therefore it is crucial to provide a method to handle the complexity of the data used in banks. In order to analyze such large data, the traditional client-server architecture is not helpful. Therefore banks are trying to move towards a Horizontal scaling method that can process and store the big data.

Several storage methods such as SQL and relational databases have been used in banks for storage and retrieval. However, due to the lack of efficient response time and processing speed, the traditional storage methods have been upgraded and now a days, several storages such as cloud, NoSQL, HDFS, etc. are used. The processing of data in a bank requires an efficient classification and clustering mechanisms so as to extract information from data and hence provide a relevant aspect in decision making and analysis. However, the banks strict policies do not reveal the absolute mechanism that is used in handling the voluminous data and its information extraction.

A storage and processing unit along with several algorithms to extract patterns over the data is being expected of any bank. But when we consider the efficiency of the analysis, the bank must be open to new technologies and better approaches to improvise the accuracy of the information. Every bank that functions today has a major problem to identify the customer behavior. Today most banks consider the present salary and the assets of the customer to analyze the behavioral pattern of a customer. But with Big Data concepts and Hadoop framework, we can do a better procedure for improvising the efficiency.

2.1.1 Drawbacks

The drawbacks of the system are,

1. The traditional client server architecture is not forward compatible.
2. The storage and processing of such architecture is less efficient comparing to big data approach.
3. The banks lack the knowledge about customer behavior and hence provide less service to the customers.
4. The reports and analysis accuracy can be improved manifold when the banks are open to big data frameworks.

2.2 PROBLEM DEFINITION

Banks have a necessity to understand the behavioral pattern of customers to improve the services provided by the bank. In order to implement new schemes for target customers. But there is a problem where banks consider only the assets and current salary of the customers to analyze their behavior. Some customers who have good salary and assets may show reluctance to pay back loans but some customers with average salary may pay back the loans and credit balance properly. Hence in order to solve this discrepancy, we are using the transactions and previous payment transactions, to understand the behavior of the customers. Since transactions of an individual are enormous amount of data, the analysis includes the usage of Hadoop framework and R programming language. By using R, we can provide a visual and graphical representation of data.

2.3 PROPOSED SYSTEM

The proposed system comprises of the combination of Hadoop, Apache pig and R to classify and cluster the bank data to identify the target customers for a scheme such as a loan or credit card issuance or setting up a market over a place. The system comprises of two stages namely classification and clustering. The

primary stage comprises of using the Hadoop framework to process and store data in the HDFS. Later the data is classified into three categories namely, VIP customers, good customers and risky customers based on some of the parameters such as transaction amount, loan status, credit card status, average salary, age, unemployment rate, disposition status and entrepreneurship rate.

Initially the data is preprocessed and converted into csv files which are further stored into the HDFS to be used for further for processing. The processing tool used in the Hadoop framework is the Apache Pig which is capable of handling enormous amount of data. Firstly the pig scripts are used to process the tables, and establish relationships over the tables. Then several parameters are used to classify the customers into three categories namely VIP customer, good customer and risky customer.

After the classification of customers, clustering is performed using R programming language. Since R is best suited for statistical approach, the RStudio IDE is used to perform clustering. The data is imported into the RStudio and K-Means Clustering algorithm is used for the clustering the customer data. Later the results are represented using graphs and plots. Using such results, the bank can decide the target customers and other decisions that are important for providing schemes and beneficiaries for the customers.

2.3.1 ADVANTAGES

The advantages of the proposed system are listed as follows:

1. Understanding the customer behavior for a bank.
2. Analyzing the bank data to provide useful results based on the behavioral pattern of customer.
3. Improves the decision making of banks regarding the schemes and loan/credit cards.

4. Provides graphical and analytical approach towards understanding the customer behavior.
5. Identification of target customers.
6. Providing knowledge to understand the market and set up variety of schemes and marketing/sales strategies.

2.4. FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

2.4.1 Economic Feasibility

Economic feasibility is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. The proposed Bank customer classification system has minimal hardware and software requirements. Minimum requirements of the software specifications are less due to the open source tools such as Hadoop, Apache Pig and RStudio.

By analyzing the parameters such as Software/hardware cost, Total estimated cost of the project, Financing of the project by equal sharing between our team members, Cost-benefit analysis. Projected cash flow and profitability, we conclude that our project is economically feasible.

2.4.2 Technical Feasibility

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out. The proposed system is technically feasible by meeting the required resources and it satisfies the required technologies.

With respect to technical feasibility, we analyzed various technical considerations that include Software/IDE's available, Programming languages available, Automatic software testing tools available, Platforms available, Protocols available for network communication, Interoperability factors and also our knowledge level and expertization with the tools available to implement our project.

2.4.3 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during the project scope definition and how it satisfies the requirements identified in the requirement analysis phase of system development. The proposed banking system is operationally feasible. Once the bank data is obtained, the customers are classified and clustered based on the parameters such as transactional data, average salary, age, entrepreneurship rate, unemployment rate, type of the cards

they possess and the status of the loan. As a result we conclude that our project is operationally feasible.

2.4.4 Legal Feasibility

Legal Feasibility determines whether the proposed system conflicts with legal requirements, e.g., the proposed system must comply with the local regulations and if the proposed venture is acceptable in accordance to the laws of the bank.

The main objectives of the legal feasibility analysis are as follows.

1. To ensure that the project is legally doable.
2. To facilitate risk management, indicating the risks and obstacles that need to be addressed within the technical analyses, the financial model and/or the Value for Money analysis.

In the proposed system, the legal issue is to obtain the permission for getting the private information from a bank or its governing body. Once the project is approved, the implementation can be done. Thus our project is legally feasible.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 SYSTEM REQUIREMENTS

3.1.1 VM ware workstation



Fig 3.1 VM ware workstation icon

VMware Workstation is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems (an x86 version of earlier releases was available); it enables users to set up virtual machines (VMs) on a single physical machine, and use them simultaneously along with the actual machine. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS. VMware Workstation is developed and sold by VMware, Inc., a division of Dell Technologies. There is a free-of-charge version, VMware Workstation Player, for non-commercial use. An operating systems license is needed to use proprietary ones such as Windows. Ready-made Linux VMs set up for different purposes are available from several sources.

VMware Workstation supports bridging existing host network adapters and sharing physical disk drives and USB devices with a virtual machine. It can simulate disk drives; an ISO image file can be mounted as a virtual optical disc

drive, and virtual hard disk drives are implemented as .vmdk files. VMware Workstation Pro can save the state of a virtual machine (a "snapshot") at any instant. These snapshots can later be restored, effectively returning the virtual machine to the saved state,[5] as it was and free from any post-snapshot damage to the VM.

VMware Workstation includes the ability to group multiple virtual machines in an inventory folder. The machines in such a folder can then be powered on and powered off as a single object, useful for testing complex client-server environments.

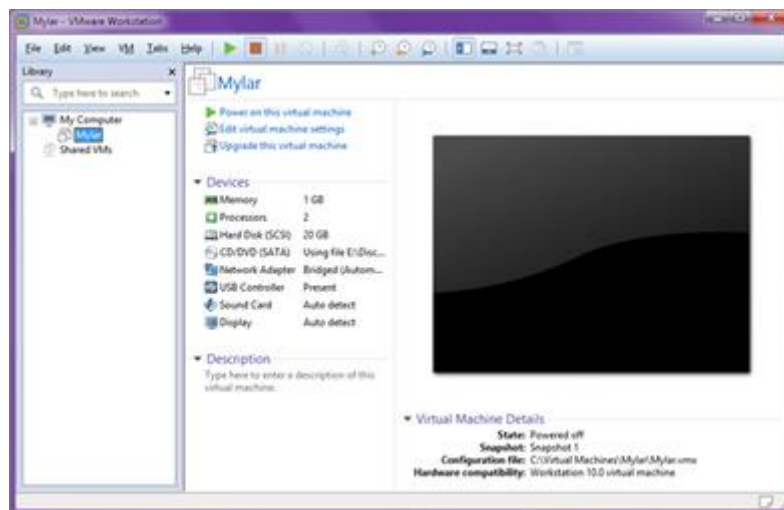


Fig 3.2 VM ware workstation platform

3.1.2 Cloudera OS



Fig 3.3 Cloudera Icon

Cloudera, Inc. is a US-based software company that provides a software platform for data engineering, data warehousing, machine learning and analytics that runs in the cloud or on premises. Cloudera started as a hybrid open-source Apache Hadoop distribution, CDH (Cloudera Distribution Including Apache Hadoop), that targeted enterprise-class deployments of that technology. Cloudera states that more than 50% of its engineering output is donated upstream to the various Apache-licensed open source projects (Apache Spark, Apache Hive, Apache Avro, Apache HBase, and so on) that combine to form the Apache Hadoop platform.

Cloudera is also a sponsor of the Apache Software Foundation. Cloudera distribution including Apache Hadoop provides an analytics platform and the latest open source technologies to store, process, discover, model and serve large amounts of data. CDH, the Cloudera Hadoop distribution, includes several related open source projects, such as Impala and Search. CDH (Cloudera's Distribution Including Apache Hadoop) is the most complete, tested, and widely deployed distribution of Apache Hadoop. CDH is 100% open source and is the only Hadoop solution to offer batch processing, interactive SQL and interactive search as well as enterprise-grade continuous availability.

3.1.3 Apache Hadoop



Fig 3.4 Apache Hadoop Icon

Apache Hadoop is an open source platform designed for distributed storage and distributed processing of Big data. Therefore it can be said as Hadoop is a framework used to access, store and process huge amount of structured, semi-structured and unstructured data. Good news is that it is not a license product, it means that anyone can download Hadoop for free and use it, that is why it is called as open source platform. We will understand the term “distributed” later, but before that let us understand the term “Framework”. Hadoop ecosystem is called as a framework because it provides a base structure for processing data. Hadoop comprises of many components that can work together in handling enormous data and processing it. These components are together referred as Hadoop Eco-Systems. Hadoop has three major components namely Core, HDFS and MapReduce.

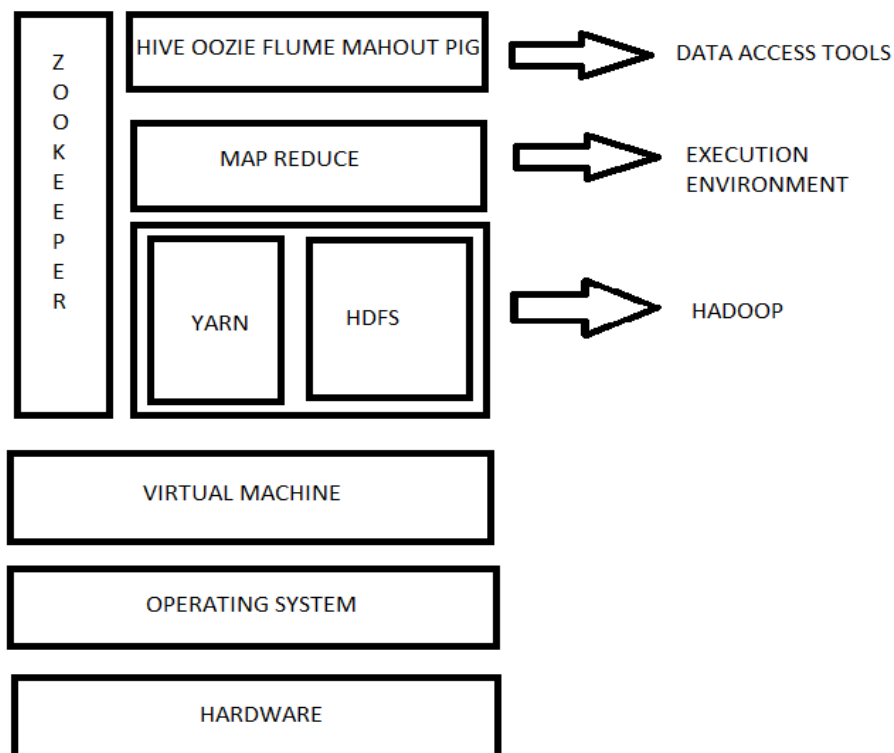


Fig 3.5 Hadoop Architecture

3.1.4 Apache Pig



Fig 3.6 Apache Pig Icon

Yahoo! Started its project contributed to Pig and its implementation. The necessity of Pig was felt by Yahoo! At the initial stage of processing huge amount of log data, which became very tedious using MapReduce as it is very rigid processing model. After the initial development, Yahoo! Hadoop users started to adopt Pig. So, a team of development engineers were asked to build the Pig photo type into a production quality product. In September 2008, Pig came into practice and the same year it became a part of Hadoop ecosystem. Pig uses sql-like scripting language Pig Latin which is a Data Flow Language. Pig latin is a simple language that executes a set of statements Pig Latin depicts a Direct Acyclic Graph (DAG). In Direct Acyclic Graph (DAG), the edges are data flows and the nodes are operators that process the data. A program which takes about 4 hours to write in java will take only 15 minutes to write in Latin.

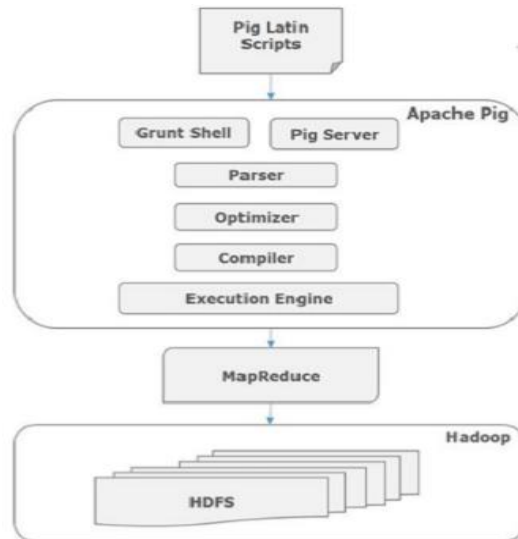


Fig 3.7 Apache Pig Architecture

3.1.5 RStudio



Fig 3.8 RStudio Icon

RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. RStudio was founded by JJ Allaire, creator of the programming language ColdFusion. Hadley Wickham is the Chief Scientist at RStudio. RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server. Prepackaged distributions of RStudio Desktop are available for Windows, macOS, and Linux. RStudio is available in open source and commercial editions and runs on the desktop (Windows, macOS, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian, Ubuntu, Red Hat Linux,

CentOS, openSUSE and SLES). RStudio is partly written in the C++ programming language and uses the Qt framework for its graphical user interface. The bigger percentage of the code is written in Java. JavaScript is also amongst the languages used.

Work on RStudio started around December 2010, and the first public beta version (v0.92) was officially announced in February 2011. Version 1.0 was released on 1 November 2016. Version 1.1 was released on 9 October 2017. In April 2018 it was announced RStudio will be providing operational and infrastructure support for Ursa Labs. Ursa Labs will focus on building a new data science runtime powered by Apache Arrow.

3.1.6 Eclipse IDE



Fig 3.9 Eclipse Icon

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGi implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles. Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License. It was one of the first IDEs to run under GNU Class path and it runs without problems under Iced Tea.

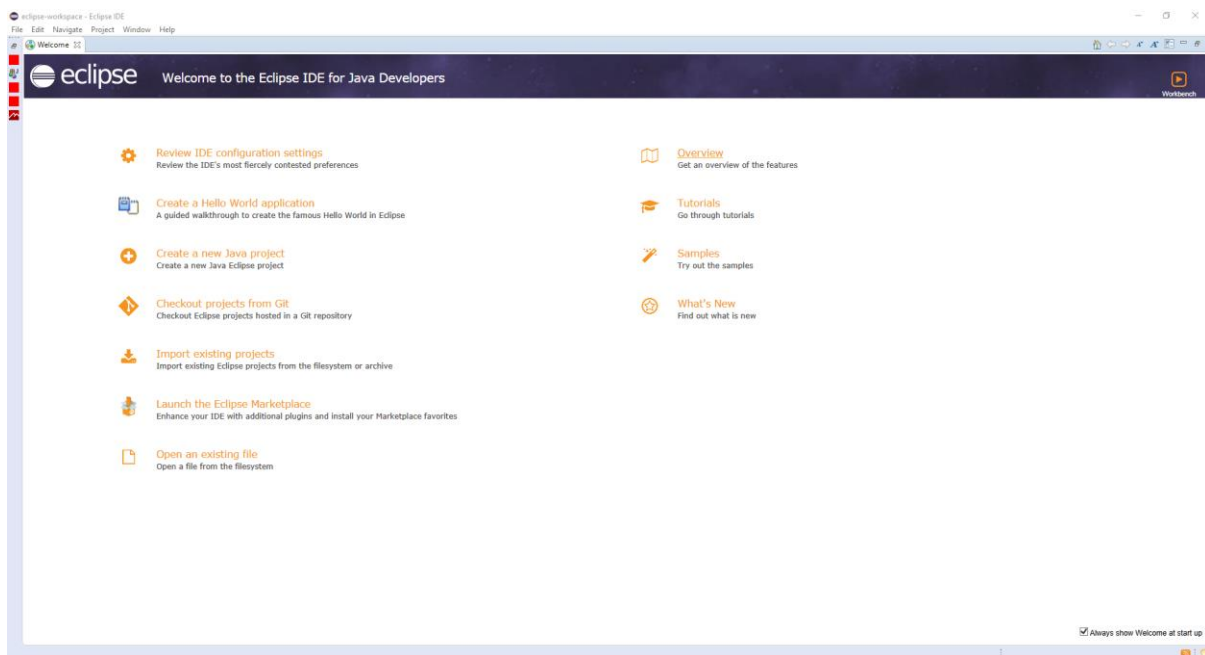


Fig 3.10 Eclipse Platform

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The project has come up with the following proposed architecture for the implementation of CLASSIFICATION AND CLUSTERING OF BANK DATA USING HADOOP, PIG AND R.

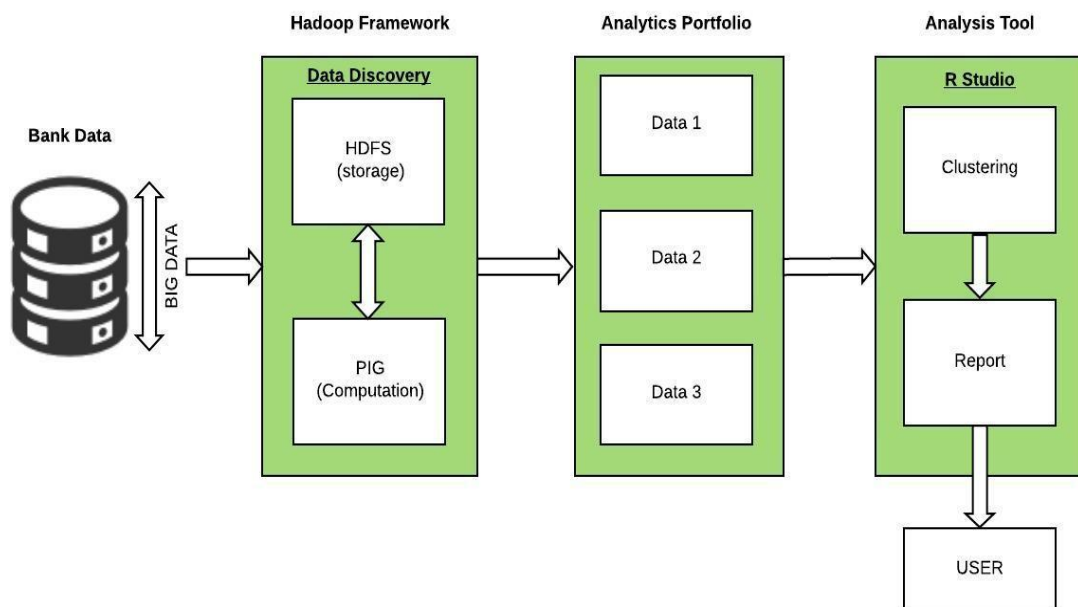


Fig 4.1 System Architecture

The large volume of structured data from the bank are collected and preprocessed from being ASCII format to CSV files in order to remove the improper data and then fed into the Hadoop Distributed File System (HDFS). The data in HDFS is retrieved and processed to classify the data using PIG. Then the classified data is further analyzed to get the accurate results. Using tools like R

Studio to perform clustering to get accurate results and based on the clustering, the report will be generated which can be viewed by the user. Therefore a number of graphical representations showing the clusters and the details of the customers are given as the end result. The report is generated by Hadoop using the queries whereas the clusters are created using R Studio.

4.2 DATA FLOW DIAGRAM

The first step involved is collecting the large volume of Bank data. All those data are preprocessed in order to clean the data and those data are load into the Hadoop Distributed File System (HDFS). HDFS is the place where the data are stored. This is followed by the process in which the relationship between the data are obtained and the tables are created based on the relationships found. From the tables created, the data are extracted to get the more accurate results by performing clustering. Then the final process is clustering the different types of data to get the results of high accuracy and the final report is generated based on the result and provided to the user.

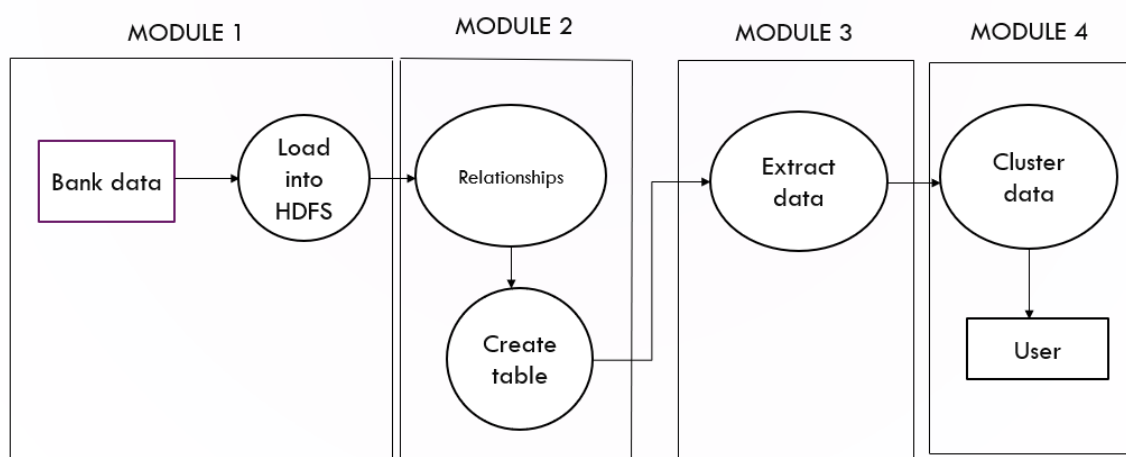


Fig 4.2 Data Flow Diagram

4.3 MODULE DESCRIPTION

The process of Classification and Clustering of Bank Data using Hadoop, PIG and R is done through various modules. Such modules involved in this system are as follows:

- Pre Processing Data
- Establishing Relationships and Creating Tables
- Extracting Data
- Clustering Data

4.3.1 Pre Processing Data:

The raw data collected from the bank may contains some improper, wrong and unnecessary data, in order to remove all those data, the data is preprocessed. Here the input data will be the raw data in Ascii format which undergone preprocessing steps to give the output data in .csv format. The processing steps including the data loaded into HDFS and then accessed using Apache PIG commands to perform cleaning and preprocessing data.



Fig 4.3 Pre Processing Data

4.3.2 Establishing Relationships and Creating Tables:

The processed data are again stored into the HDFS and then they are accessed using PIG Latin queries to create a relationship between them and then based on the relationships found the tables are created. Here the input data will be in the .csv format and the output will be the formatted tables.



Fig 4.4 Establishing Relationships and Creating Tables

4.3.3 Extracting Data:

The data in the format of tables are now applied with the constraints to classify them into high, medium, low risk customers. Here the input is table data and query statements using Latin are established to classify data into three sets and those classified data are stored in HDFS. The various parameters considered for the classification of the data are Transaction amount, Loan Status, Credit card status, Average Salary, Age, Unemployment rate, Entrepreneurship rate, Disposition etc.

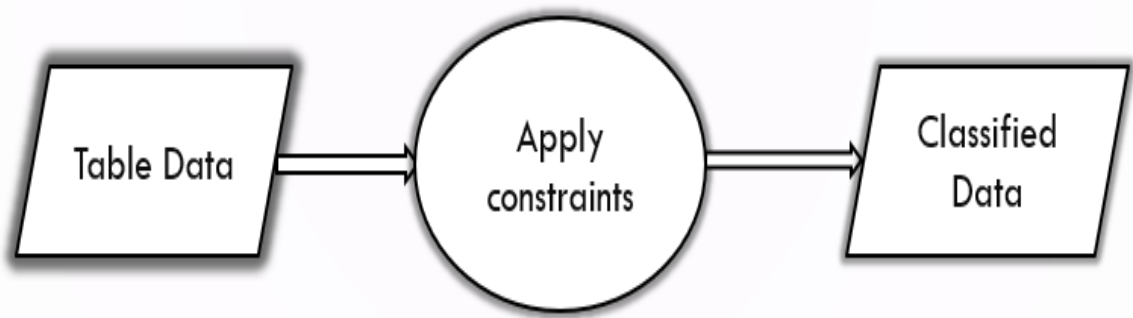


Fig 4.5 Extracting Data

4.3.4 Clustering Data:

The classified data is further loaded into the R studio to perform clustering in order to get the more accurate results. Here the input data is classified data which are loaded into the R studio and clustered using K-Means clustering algorithm and the graphs are plotted for interface. The K-Means clustering formula used is

$$J(V) = \sum_{i=1}^C \sum_{j=1}^{C_i} (\|x_i - v_j\|)^2$$

Where $\|x_i - v_j\|$ is Euclidian Distance, C_i is number of data points in 'I' cluster, C is cluster center. The K-Means Algorithm can be given as follows

1. Clusters the data into k groups where k is predefined.
2. Select k points at random as cluster centers.
3. Assign objects to their closest cluster center according to the Euclidean distance function.
4. Calculate the centroid or mean of all objects in each cluster.

5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.



Fig 4.6 Module Web-Server

CHAPTER 5

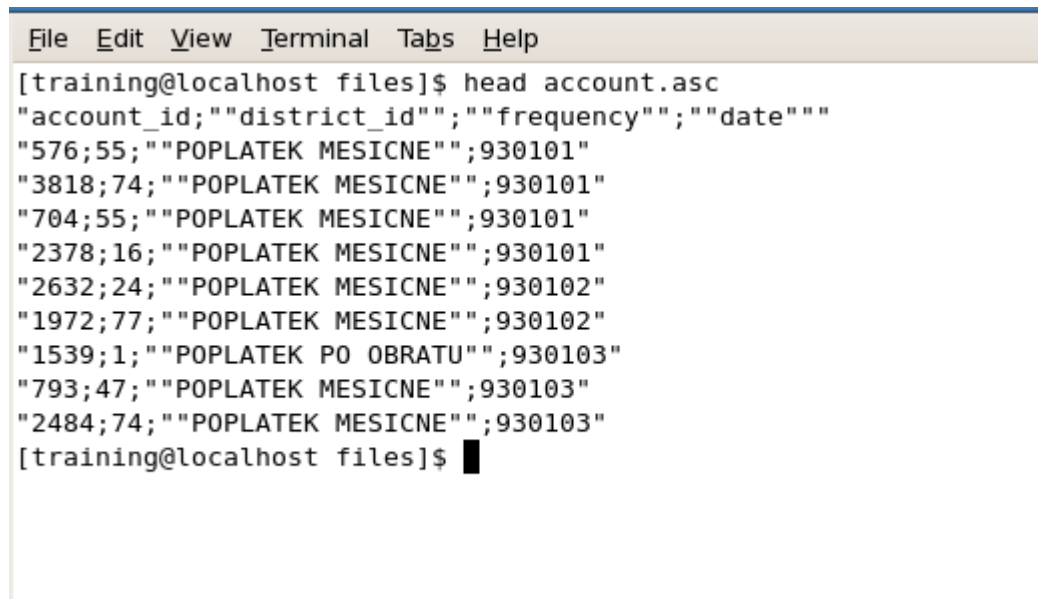
EXPERIMENTAL RESULTS

5.1 MODULE 1

5.1.1 PREPROCESSING OF DATA

The preprocessing of data includes changing data from ASCII format to CSV format so that the data can be used for further processing and storage. Unless the data is compatible to the formats used in Pig the data cannot be used or will lead to improper processing of data.

Converting ASCII File to .CSV File



```
File Edit View Terminal Tabs Help
[training@localhost files]$ head account.asc
"account_id";"district_id";"frequency";"date"
"576;55;"POPLATEK MESICNE";930101"
"3818;74;"POPLATEK MESICNE";930101"
"704;55;"POPLATEK MESICNE";930101"
"2378;16;"POPLATEK MESICNE";930101"
"2632;24;"POPLATEK MESICNE";930102"
"1972;77;"POPLATEK MESICNE";930102"
"1539;1;"POPLATEK PO OBRATU";930103"
"793;47;"POPLATEK MESICNE";930103"
"2484;74;"POPLATEK MESICNE";930103"
[training@localhost files]$
```

Fig 5.1 ASCII File

```
File Edit View Terminal Tabs Help
[training@localhost project]$ head account.csv
576,55,POPLATEK MESICNE,930101
3818,74,POPLATEK MESICNE,930101
704,55,POPLATEK MESICNE,930101
2378,16,POPLATEK MESICNE,930101
2632,24,POPLATEK MESICNE,930102
1972,77,POPLATEK MESICNE,930102
1539,1,POPLATEK PO OBRATU,930103
793,47,POPLATEK MESICNE,930103
2484,74,POPLATEK MESICNE,930103
1695,76,POPLATEK MESICNE,930103
[training@localhost project]$
```

Fig 5.2 .csv File

5.1.2 LOADING DATA INTO HDFS

After the data is being converted into csv format, it is loaded into the HDFS (HADOOP DISTRIBUTED FILE SYSTEM) so that it can be used by hadoop components such as pig, hive, zookeeper etc.,

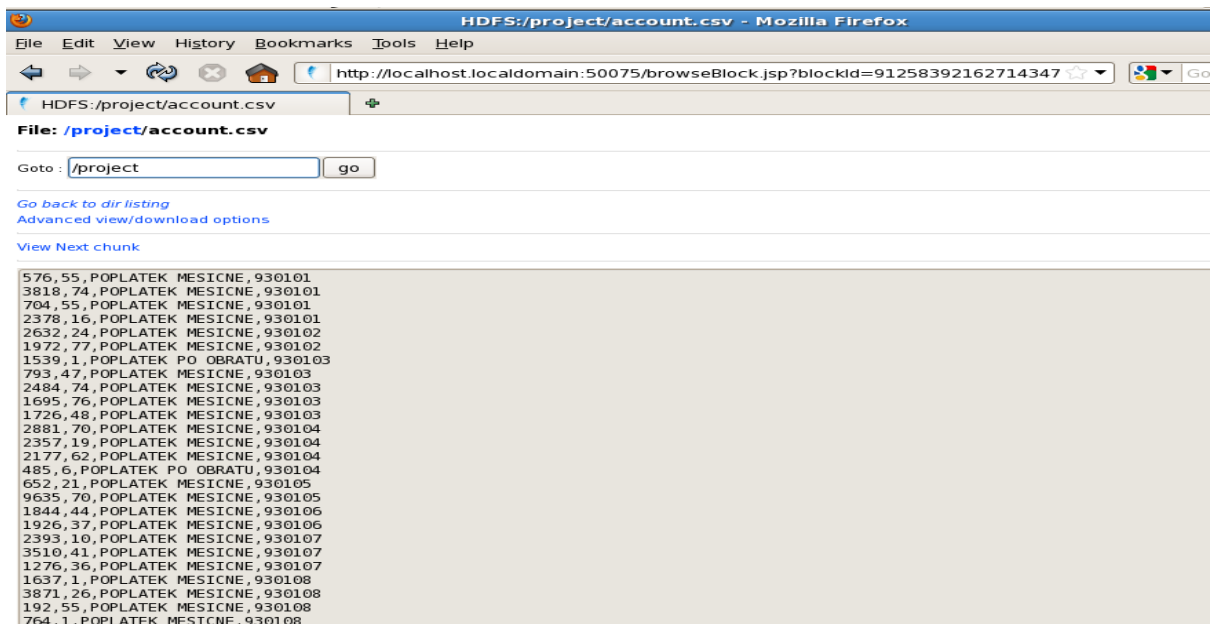


Fig 5.3 Loading data into HDFS

5.2 MODULE 2

5.2.1 PROCESSING OF DATA

Initially the data is processed by inserting the fields forming tables. Later the fields are formed into tuples using unique keys. The tables are further joined to provide the relevant information regarding the parameters used to classify the data into three category.

```
(13803,1,471114)
(13820,33,485731)
(13845,15,730216)
(13852,5,720126)
(13886,44,545412)
(13912,50,751120)
(13915,77,741123)
(13921,62,706220)
(13923,7,795222)
(13924,54,525909)
(13931,8,420101)
(13955,1,456030)
(13956,1,430406)
(13968,61,680413)
```

Fig 5.4 Inserting Fields

```
(13803,{(13803,1,471114)})
(13820,{(13820,33,485731)})
(13845,{(13845,15,730216)})
(13852,{(13852,5,720126)})
(13886,{(13886,44,545412)})
(13912,{(13912,50,751120)})
(13915,{(13915,77,741123)})
(13921,{(13921,62,706220)})
(13923,{(13923,7,795222)})
(13924,{(13924,54,525909)})
(13931,{(13931,8,420101)})
(13955,{(13955,1,456030)})
-----
```

Fig 5.5 Forming Tuples

```
(13803,1,471114)
(13820,33,485731)
(13845,15,730216)
(13852,5,720126)
(13886,44,545412)
(13912,50,751120)
(13915,77,741123)
(13921,62,706220)
(13923,7,795222)
(13924,54,525909)
(13931,8,420101)
```

Fig 5.6 Flattening Tuples

```
(13820,33,70)
(13845,15,45)
(13852,5,46)
(13886,44,64)
(13912,50,43)
(13915,77,44)
(13921,62,48)
(13923,7,39)
(13924,54,66)
(13931,8,76)
```

Fig 5.7 Using UDFs

5.3 MODULE 3

5.3.1 Extracting Data

Finally all the data are joined together to form a complete table from which the required data can be extracted. The resultant tables is stored into the HDFS so that it can be used to classify the data used the pig queries.

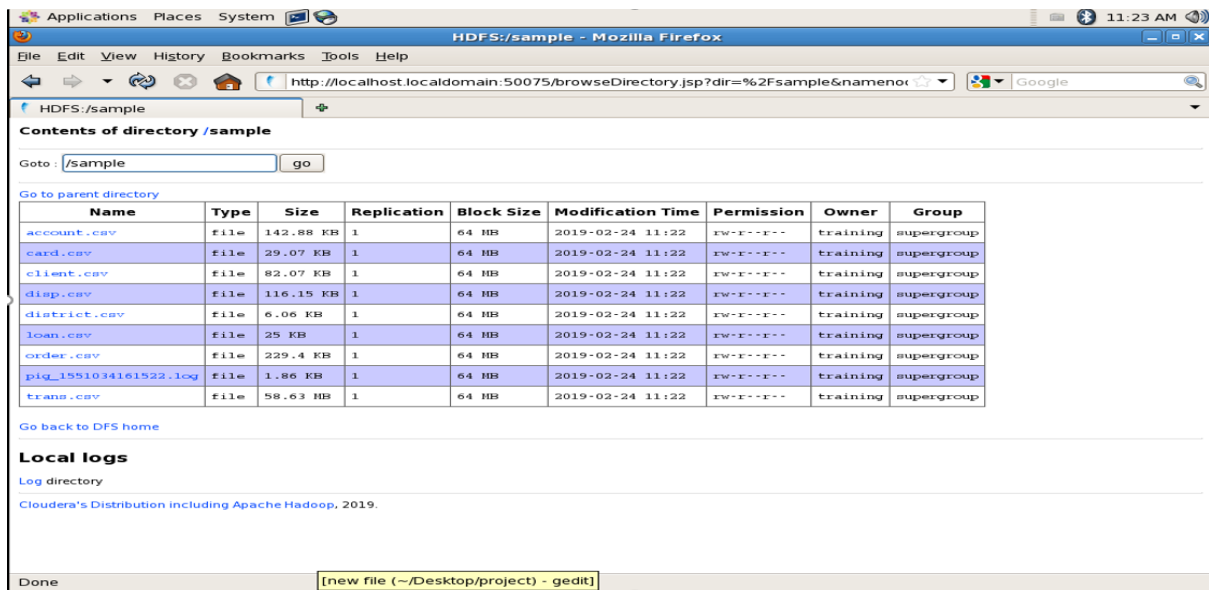


Fig 5.8 Input files in HDFS

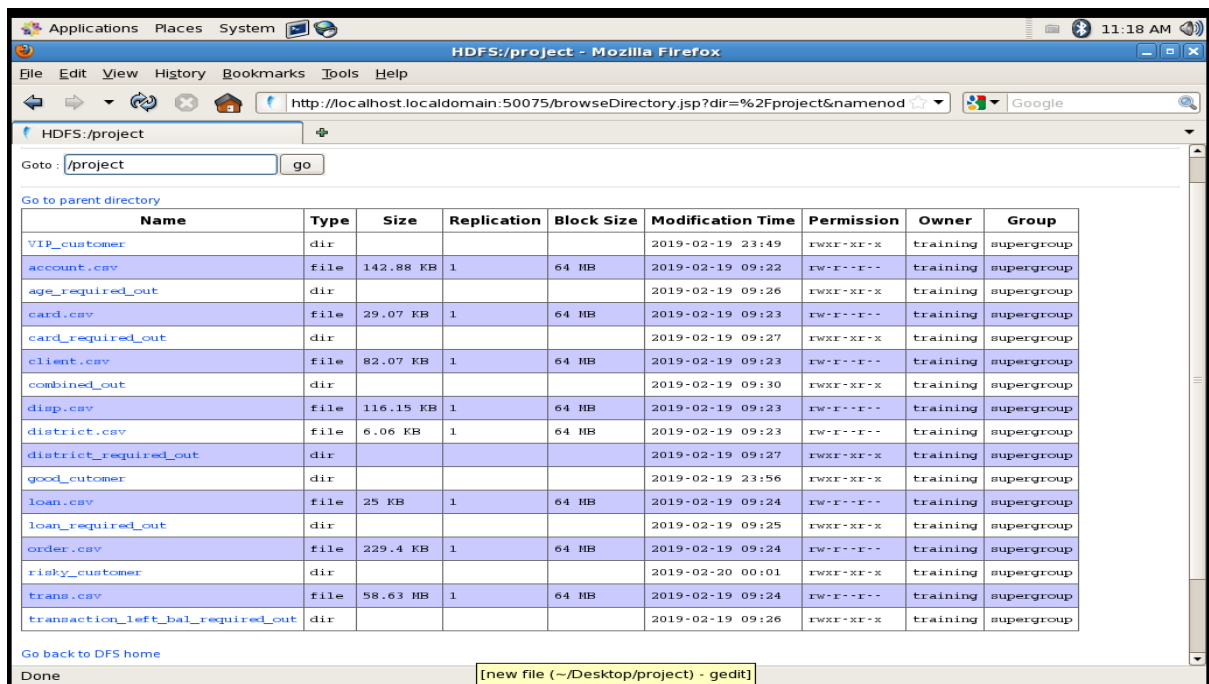


Fig 5.9 Customized data in HDFS

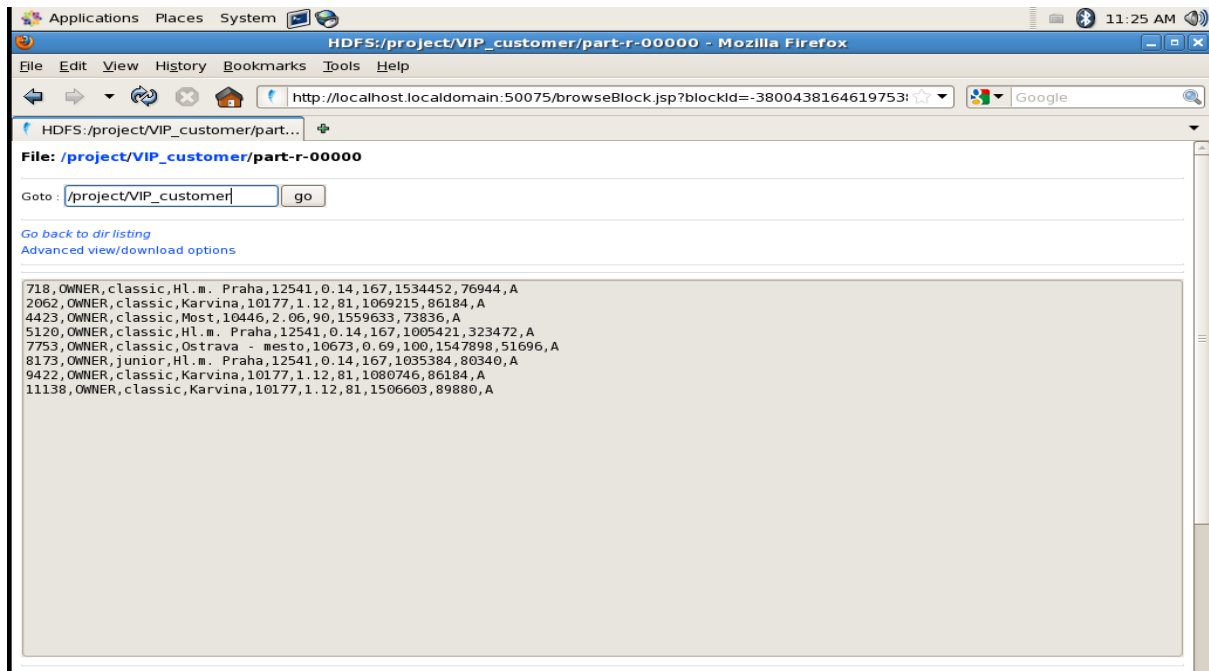


Fig 5.10 Extracted Files

5.4 MODULE 4

5.4.1 Clustering of Data

Comparing data that uses transaction and which does not use the transaction details of a customer as a parameter for analysis of behavioural pattern of customers. The Fig 5.11 and 5.12 represent the boxplot diagram of data that has transaction and transaction less data.

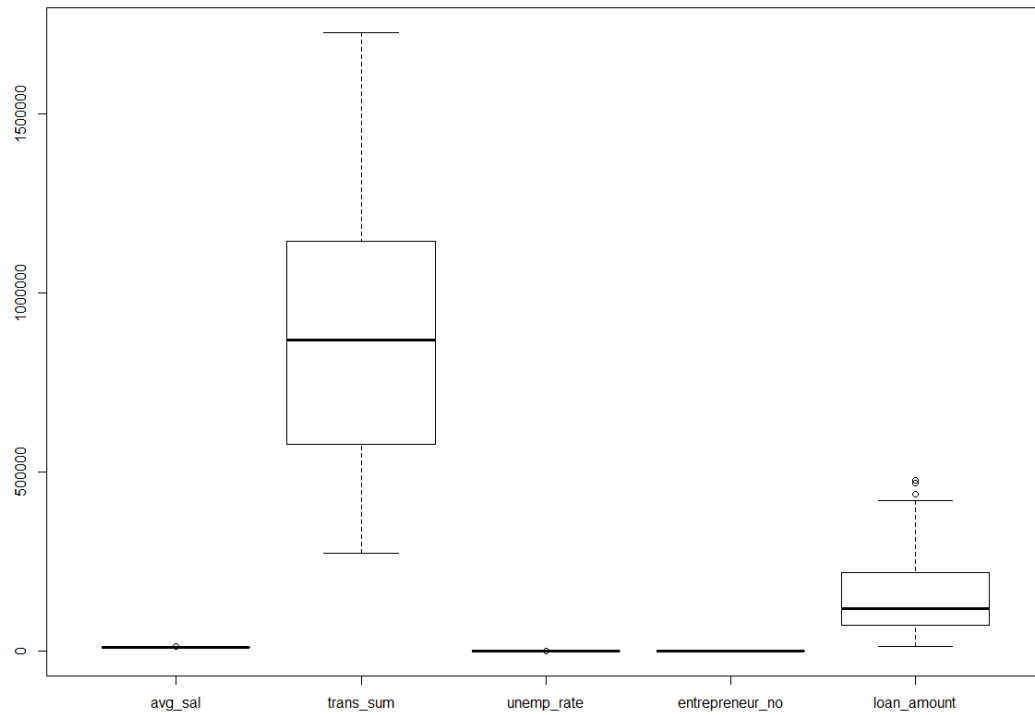


Fig 5.11 Boxplot of numerical data that uses transaction

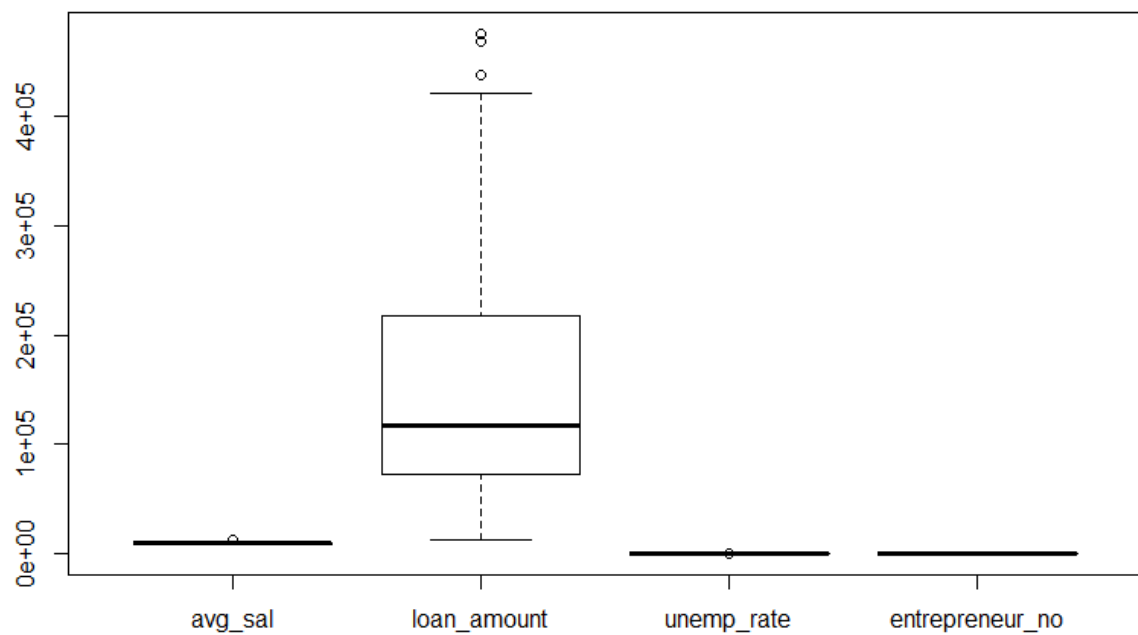


Fig 5.12 Boxplot of numerical data that does not use transaction

The Fig 5.13 and 5.14 represents cluster plot where the data uses transaction parameter and without transaction parameter. There can be seen a slight difference in plot against cluster points and its sum of squares.

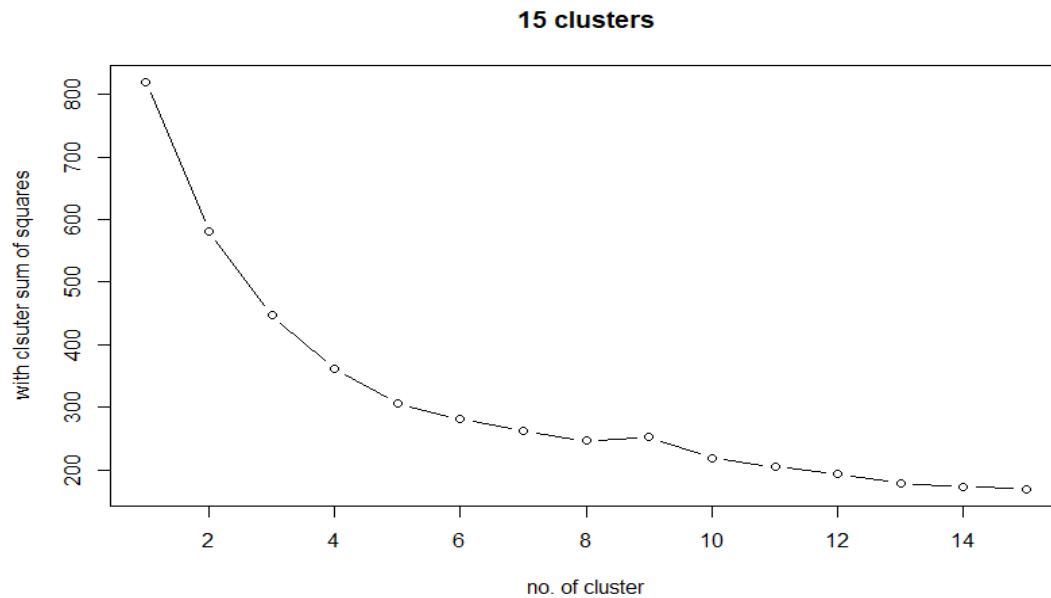


Fig 5.13 Plot of Transaction used data

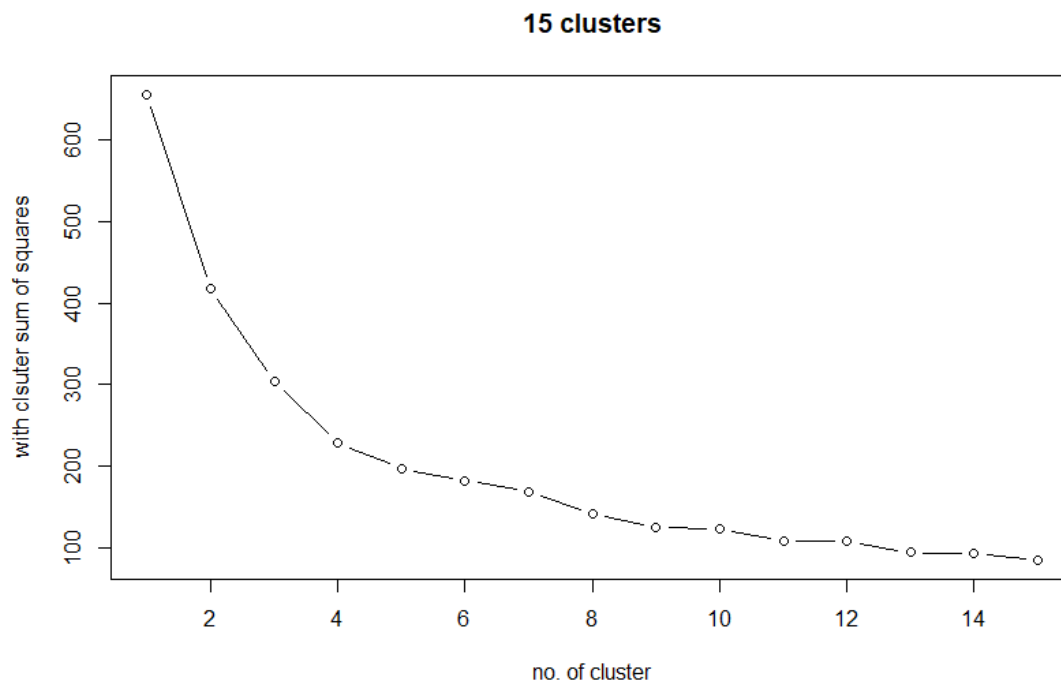


Fig 5.14 Plot of Transaction less data

The cluster points of Average salary and Transaction data with Average salary and transaction less data i.e., using the loan amount can be seen in the Fig 5.15 and 5.16. The numerical cluster plot using the cluster points can be seen in the Fig 5.17 and 5.18

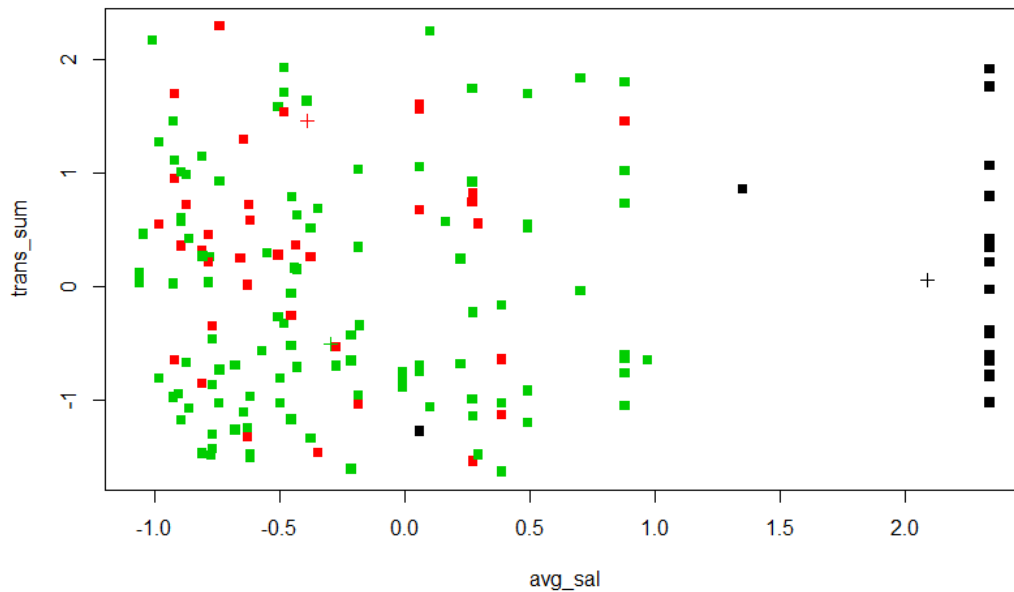


Fig 5.15 Plot of Average salary and Transaction data

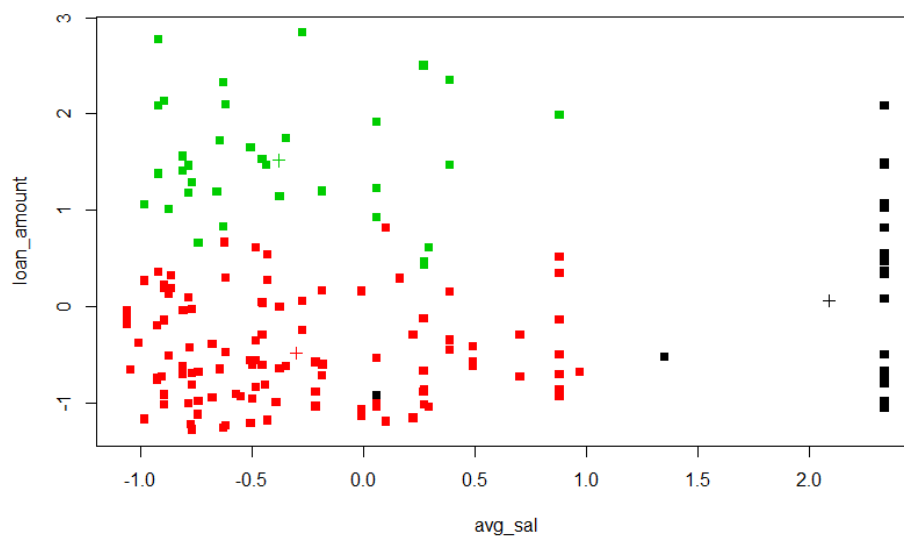


Fig 5.16 Plot of Average salary and Loan Amount

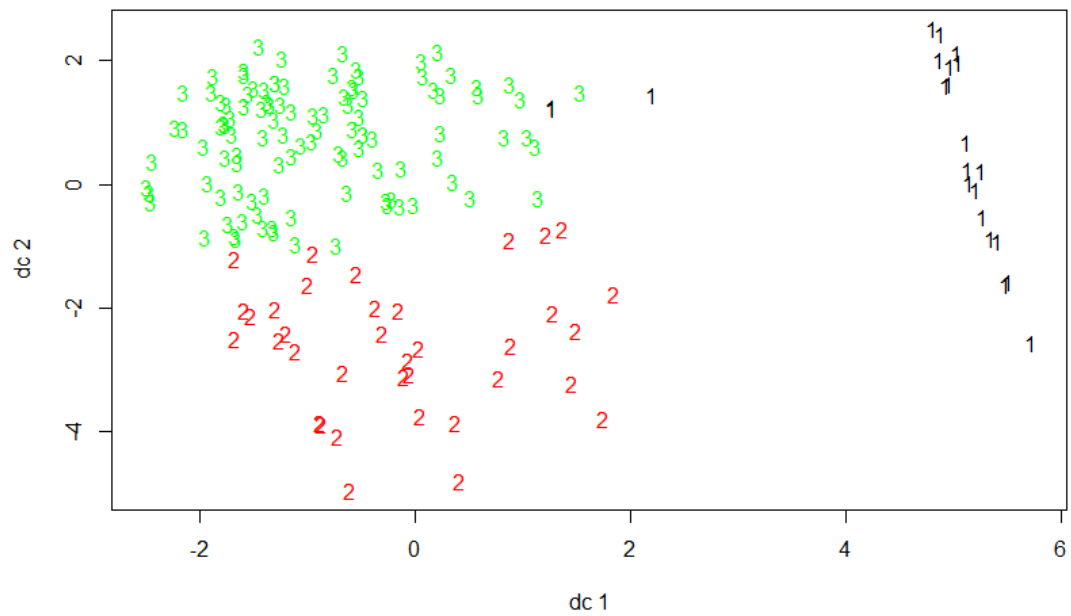


Fig 5.17 Cluster points of three clusters(transaction)

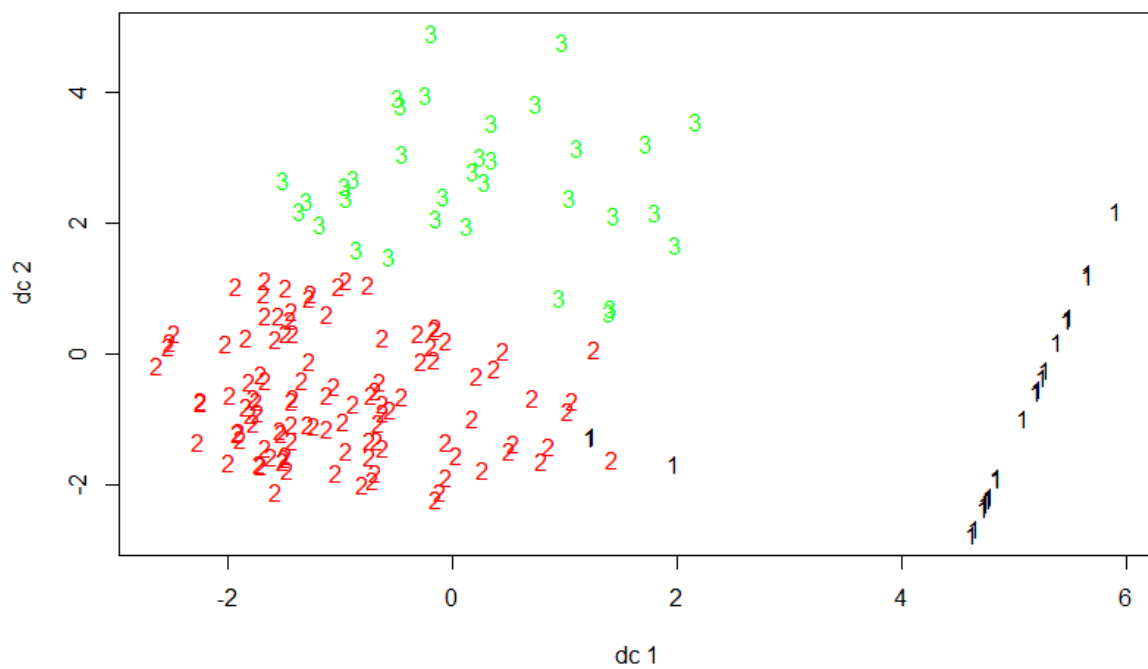


Fig 5.18 Cluster points on transaction less data

The Fig 5.19 and 5.20 represent the cluster plot of the bank data and Fig 5.21 and 5.22 represents the mean cluster plot of the data of the transaction and transaction less data. Therefore from these clusters we can see that the point variability of the data when used transaction is 66.26% while using loan amount is 76.32%.

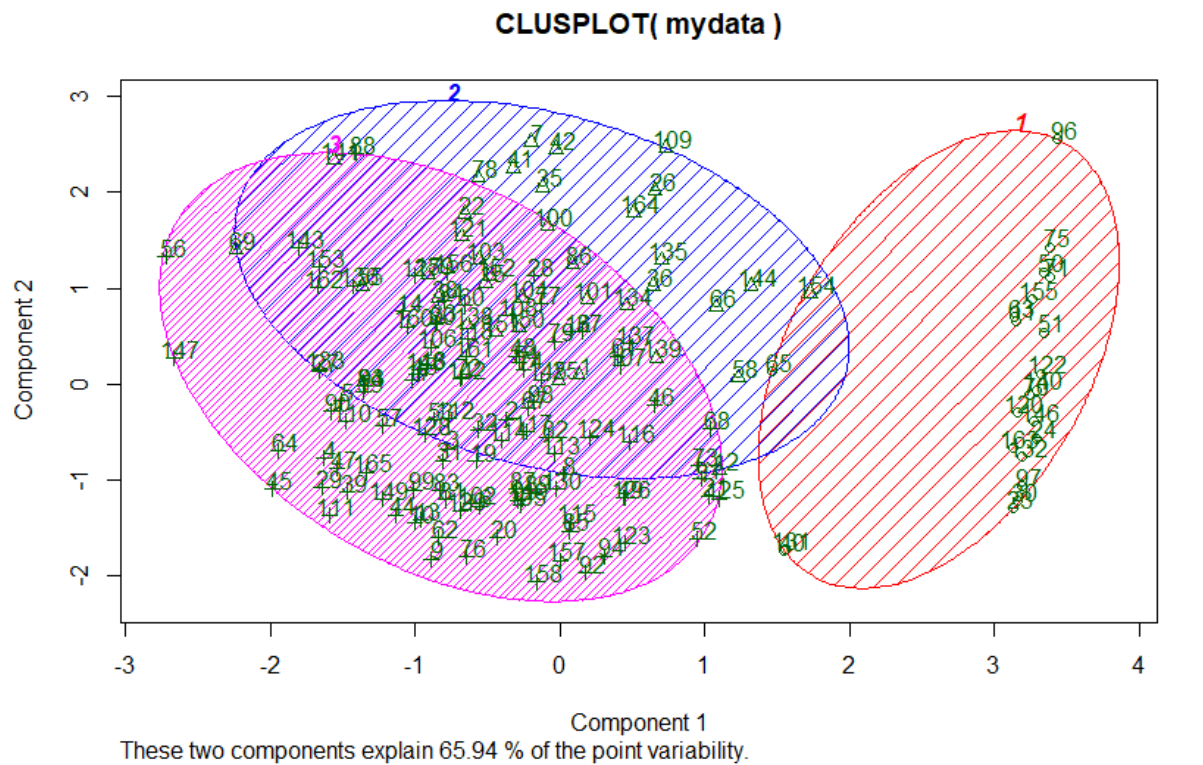


Fig 5.19 Cluster plot of transaction based classification

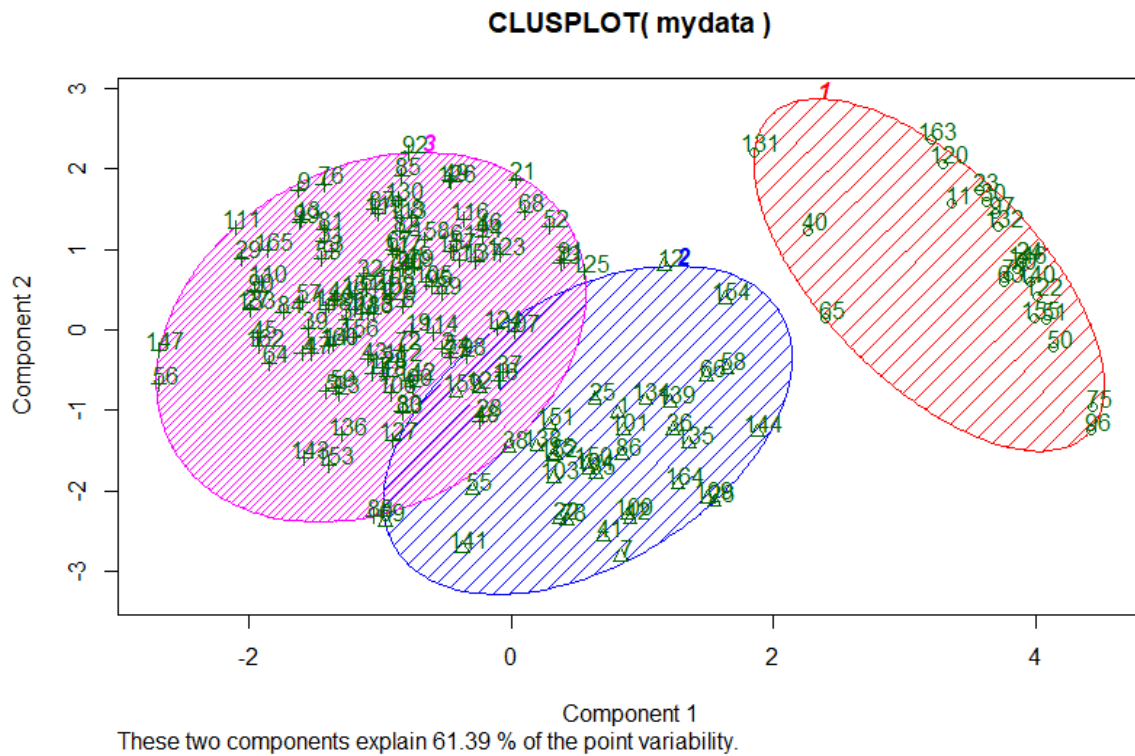


Fig 5.20 Mean Cluster plot of transaction based classification

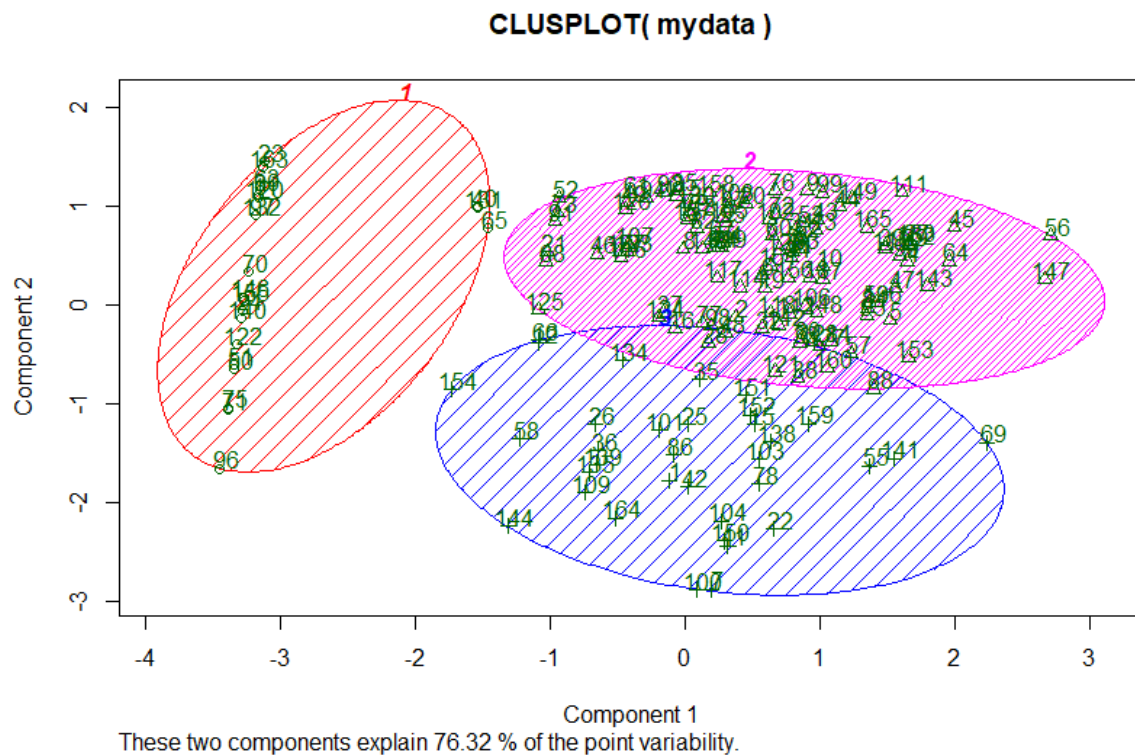


Fig 5.21 Cluster plot of transaction based classification

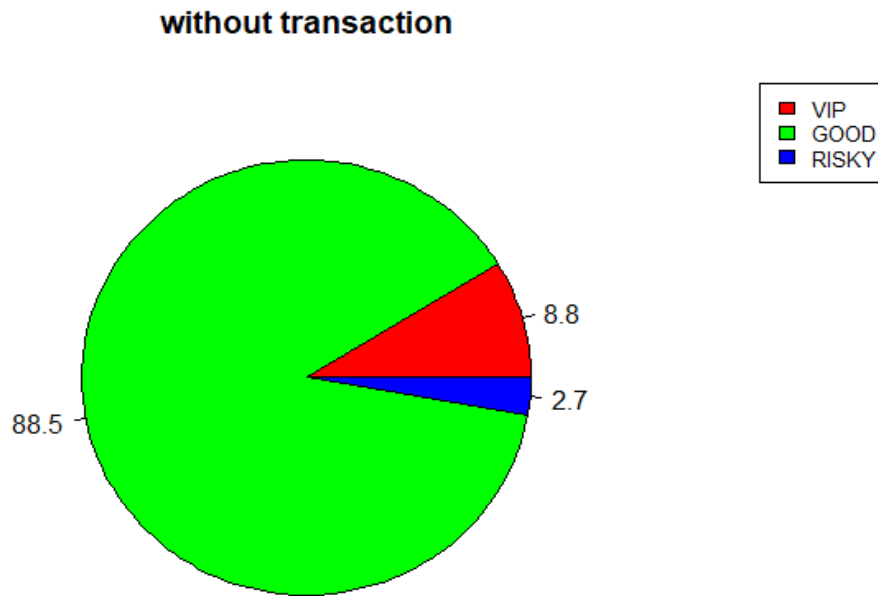


Fig 5.23 PieChart of Transaction less Data

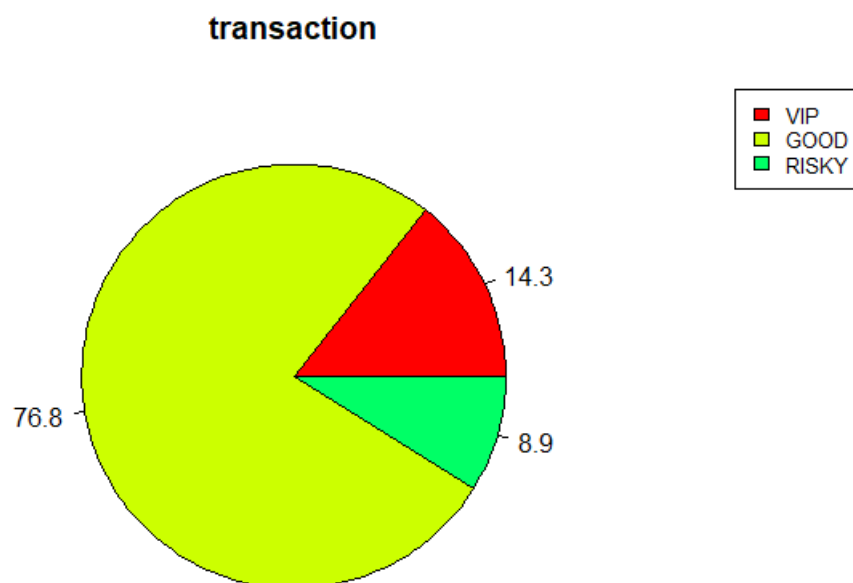


Fig 5.24 PieChart for Transactional Data

CHAPTER 6

TESTING

6.1 TEST PLAN

System testing is the process of validating and verifying that a system

- meets requirements that guided its design and development
- works as expected
- Can be implemented with the same characteristics.

So, testing has been carried out to check whether the proposed system has met the requirements and has derived the expected result.

6.2 TEST DESIGN

System testing is a process of checking whether the developed application is working according to the original objectives and requirements. The system should be tested experimentally with test data so as to ensure that the system works according to the required specification. When the system is found working, test it with actual data and check performance.

All tests should be traceable to customer requirements. The focus of testing will shift progressively from programs. Exhaustive testing is not possible. To be more effective, testing should be one, which has probability of finding errors.

The following are the attributes of good test

- A good test has a high probability of finding an error.
- A good test is not redundant.
- A good test should be “Best of Breed”.

- A good test should neither too simple nor too complex.

6.2.1 TYPES OF TESTING

The details of the software functionality tests are given below. The testing procedure that has been used is as follows:

- Unit/Component Testing
- System Testing

6.3 TEST EXECUTION

6.3.1 UNIT/COMPONENT TESTING

Project Name:	Classification and Clustering of Bank Data using Hadoop, PIG and R.	Test Designed by:	Agilan S
Module Name:	Pre Processing Data	Test Designed date:	15/01/2019
Pre-condition:	Availability of collected Bank raw data	Test Executed by:	Faheen Fathima B N
Description:	To test whether the collected raw data is converted into required format	Test Execution date:	27/01/2019

Test Case#	Test Title	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Post Condition
------------	------------	------------	-----------	-----------------	---------------	-------------------	----------------

M1_1	Preprocessing and Data conversion.	1. Collect the Data in whatever format they are (mostly Ascii format) 2. Load those data into HDFS. 3. Use Apache PIG commands	Large volume of Bank data in Ascii format.	Processed and cleaned data which is in .csv format.	The data is converted into the .csv format and get stored in HDFS.	pass	-
------	------------------------------------	--	--	---	--	-------------	---

Table 6.1 Pre Processing Data

Project Name:	Classification and Clustering of Bank Data using Hadoop, PIG and R.	Test Designed by:	FaheenFathima B N
Module Name:	Establishing Relationships and Creating Tables.	Test Designed date:	02/02/2019

Pre-condition:	Availability of processed data which is in .csv format.	Test Executed by:	Agilan S
Description:	To test if the module is able to find the relationships between the data and to create tables.	Test Execution date:	10/02/2019

Test Case#	Test Title	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Post Condition
M2_1	Relationship establishment and Tables Creation.	1. Get the processed data in a required format. 2. Develop the PIG Latin queries to develop relationship and tables.	Processed Data in .csv format.	Different tables with the established relationship between them.	Tables with relationships established.	Pass	

Table 6.2 Establishing Relationships and Creating Tables

Project Name:	Classification and Clustering of Bank Data using Hadoop, PIG and R.	Test Designed by:	Arun Kumar A.K
Module Name:	Extracting data	Test Designed date:	16/02/2019
Pre-condition:	Availability of data in the table format.	Test Executed by:	Vinitha B.
Description:	To test if the tabular data is classified into three datasets as required.	Test Execution date:	25/02/2019

Test Case#	Test Title	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Post Condition
M3_1	Extracting Data from the tables.	1. Analyse and confirm the different parameters which are	Tabular Data with relationships among them.	Three different datasets which are named as high, low and medium	Three datasets named as high, low and medium risk customers respectively.	Pass	After classification of data, all the three sets has to be stored in HDFS.

		necessar y for classifica tion.		risk customers.			
		2.Establi sh query statemen ts using PIG Latin to classify data					

Table 6.3 Extracting Data

Project Name:	Classification and Clustering of Bank Data using Hadoop, PIG and R.	Test Designed by:	Vinitha B.
Module Name:	Clustering Data	Test Designed date:	02/03/2019
Pre-condition:	Three sets of classified Data	Test Executed by:	Arun Kumar A.K
Description:	To test if the data can be connected to the server	Test Execution date:	05/03/2019

Test Case#	Test Title	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Post Condition
M4_1	Clustering the Data	1. Get and load the classified data into R studio. 2. Apply K-means clustering formula	Three sets of classified data	Clustered Data with graphical representation.	Clustered Data with graphical representation.	Pass	

Table 6.4 Clustering Data

6.3.2 SYSTEM TESTING

Project Name:	Classification and Clustering of Bank Data using Hadoop, PIG and R.	Test Designed by:	Faheen Fathima B.N Vinitha B.
----------------------	---	--------------------------	--------------------------------------

System Name:	Classification and Clustering of Bank Data using Hadoop, PIG and R.	Test Designed date:	06/03/2019
Pre-condition:	Hadoop Framework setup along with R Studio.	Test Executed by:	Agilan S. Arun Kumar A.K
Description:	To test if the whole system gets the input properly and process it and provides the required output or not.	Test Execution date:	08/03/2019

Test Case#	Test Title	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Post Condition
TC_1	Verify whether the given raw data is classified into three datasets.	1.Load Data in HDFS 2.Preprocessing Commands 3.Classification Commands.	Unprocessed Bank data.	Three sets of classified data.	Three sets(HIGH, LOW, MEDIUM) of classified data	Pass	

TC_2	Verify whether the data is clustered and provided in the form of graph.	1.Load the data into RStudio. 2.Apply K-means algorithm on the data.	Different sets of classified data.	Graphical representation of clustered data	Graphical representation of clustered data	Pass	
------	---	---	------------------------------------	--	--	-------------	--

Table 6.5 System Testing

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The enormous rate of growing data leads to store and process them to get the accurate results in almost all the fields. Hence all the business sectors started using big data handling to make them sustainable in this competitive world. Thus this system provides an approach to handle the large amount of data in the banking sector and process them to help the banks to do their business effectively. It becomes an integrated system where the data is stored, classified, clustered and represent the result in graphical format which helps the banks to understand their customers and to identify their targeted customers easily. Since it takes large amount of data for process, the accuracy of the result provided by the system is far better than earlier systems which makes the banks to take more accurate decisions.

7.2 FUTURE WORK

There is some future work which can be done in this report. They are

- Applying machine learning algorithms to the applied statistics to predict the behavior of customers.
- It can be used to predict the effectiveness of a strategy or a policy of bank.
- Several parameters can be improvised to provide more specialized calculations and analysis with respect to the bank.
- Multi node Hadoop cluster can be set up to facilitate horizontal scaling methods used in big data.

REFERENCES

- [1] Ivanilton Polato a,b,n , Reginaldo Ré b , Alfredo Goldman a , Fabio Kon: A comprehensive view of Hadoop research—A systematic literature review, Journal of Network and Computer Applications 46(2014)
- [2] PrathyushaRani Merla, Yiheng Liang: Data Anlaysis Using Hadoop Map Reduce Environment, IEEE International Conference on Big Data (BIGDATA), 2017.
- [3] SFeng Li, Beng Chin Ooi, M. Tamer Ozsü, and Sai Wu: Distributed data management using MapReduce, ACM Comput. Surv. 46, 3, Article 31 (January 2014)
- [4] Arushi Jaina, Vishal Bhatnagara Ambedkar: Crime Data Analysis Using Pig with Hadoop, International Conference on Information Security & Privacy, 2015
- [5] Shruti Verma, Vinod Maan: Comparative Analysis of Hive and Pig, International Journal of Research in Advent Technology, Vol.6, No.5, May 2018
- [6] <https://www.researchgate.net/publication/329402555>
- [7] https://www.tutorialspoint.com/apache_pig/apache_pig_user_defined_functions.html
- [8] https://www.tutorialspoint.com/apache_pig
- [9] <https://www.edureka.co/big-data-and-hadoop>
- [10] <https://www.statmethods.net/r-tutorial/index.html>

- [11] <https://www.dezyre.com/article/difference-between-pig-and-hive-the-two-key-components-of-hadoop-ecosystem/79>
- [12] <https://www.hadoop360.datasciencecentral.com/blog/pig-vs-hive-vs-sql-difference-between-the-big-data-tools>
- [13] Dirk deRoos, Chris Eaton, George Lapis, PauZikopoulos and Tom Deutsch, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data.McGraw Hill Osborne Media;1 edition(October19,2011).
- [14] A B M Shawkat Ali, Saleh A. Wasimi (2009), “*Data Mining: Methods and Techniques*”, CENGAGE Learning, India
- [15] Tekiner F. and Keane J.A., Systems, Man and Cybernetics (SMC), “Big Data Framework” 2013 IEEE International Conference on 13–16 Oct. 2013, 1494–1499.