DATA MINING CLASSIFICATION ALGORITHM BASED ON MAPREDUCE

Msc Computer and Communication Networks

Abdallah SOBEHY
abdallah.sobehy@telecom-sudparis.eu

Neha KAUL
neha.kaul@telecom-sudparis.eu

TELECOM SUD PARIS
EVRY, FRANCE 91000

Table of Contents

Introduction

"The number of transistors incorporated in a chip will approximately double every 24 months."
--Gordon Moore, Intel co-founder

With transistors reaching their physical limit, doubling the processing speed is no longer achieved by increasing the processor's clock frequency, but rather by having multiples cores on chip and performing parallel processing. But this method would suffer from scalability issues when processing large sums of data "Big Data"; because even with powerful processors, their computing power is fixed while the data is increasing. Parallelization then arises as the main countermeasure to the scalability issue. It is shifting gradually from chip level to commodity hardware, which is the use of multiple computers; not necessarily with very high computing power - added together to form a dynamic scalable computing power depending on the size of data. Utilizing multiple computers introduces new challenges to the process of computation such as: heterogeneity, cluster management, possible failures of nodes. Managing such challenges is a complicated task for the programmer, as it creates the need for a programming model that transparently handles those issues, adding extra work on the programmer to write down an algorithm to solve such underlying problems.
In 2004, Google pioneered a programming model known as "MapReduce" [4] which is designed to process large sums of data in a distributed environment. It implicitly deals with the heterogeneity of computers that are used in computations, the increasing volume of data, possible failures of tasks and communication between machines without the interference of the programmer. It is mainly composed of two stages:
a) Map which applies a desired function to all data in the form of key/value pairs to produce a set of key/value pairs.
b) Reduce where the output of the map phase is shuffled and the key/value pairs with the same key are summarized by the reduce function to produce the needed result.

With the increase of internet users around the globe and the introduction of cloud computing; data mining is becoming a very important issue when processing Big Data. Particularly, data mining is the analysis of large sums of data "Big data" in order to extract useful information; its

main purpose is to make efficient future decisions by learning from existing data. Cloud Computing with its numerous, shared resources offers a rich medium for storing Big Data. Initially developed to work smoothly in a distributed environment, like the cloud with parallel processing; MapReduce is an optimal method for the application of Data mining.

In this paper we will apply a data mining technique and an associated algorithm to a real life example, using the MapReduce programming model. Section 1 introduces the data mining concept along with two data-mining techniques: Classification and association rules. For each technique we give two algorithms for implementation. Section 2 describes the MapReduce model and the most known open source framework implementation "hadoop". Section 3 briefs the environment setup that was used in the implementation of the case study. Section 4 details the case study and the implementation of two Classification algorithm: Zero R and One R to show how data mining can be used for future marketing campaign of a bank to target customers who are likely to accept an offered product. The reasons for choosing those two algorithms are that; "Zero R" is used as a benchmark to judge the performance of other algorithms, and "One R" is simple to implement with a fairly good accuracy. Finally, in section 5 we conclude our work with a comparison between the used algorithms.

1. Introduction to Data Mining

The development of the Internet and Information Technology has generated a large amount of data stored in numerous databases in various locations. This data is being used in many different applications and fields. However, we are unable to turn this data into knowledgeable and useful information for managerial decision making processes in business. The data may be in different formats, like documents, audio/video, numbers, text, figures, Hypertext formats, etc. As the data is available in the different types, it should be stored, maintained and analyzed in the correct way. To take full advantage of the available data; data retrieval only is not sufficient. A tool capable of automatic summarization of data, extraction and recovery of the essence of information stored and discovery of possible repeating patterns in raw data is required. With the continuously growing size of data that is being stored in databases, files and other repositories, it is of the utmost importance, to possess a powerful tool for analysis and interpretation of the data that is capable of extracting interesting information from the data that could help in the decision-making processes. The solution to this problem is 'Data Mining'.

1.1 Data Mining Definition

Data mining is the extraction of hidden predictive information from large databases; it is a powerful technology with great potential to help organizations focus on the most important information in their data warehouses [14]. Data mining tools help in predicting future trends and behaviors and also help organizations to make practical and knowledge driven decisions. Data mining is popularly known as Knowledge Discovery in Databases (KDD). It is a method or a process of extraction of inherent, previously unknown and potentially useful information from the data. Data mining aims at finding hidden information or patterns in the data.

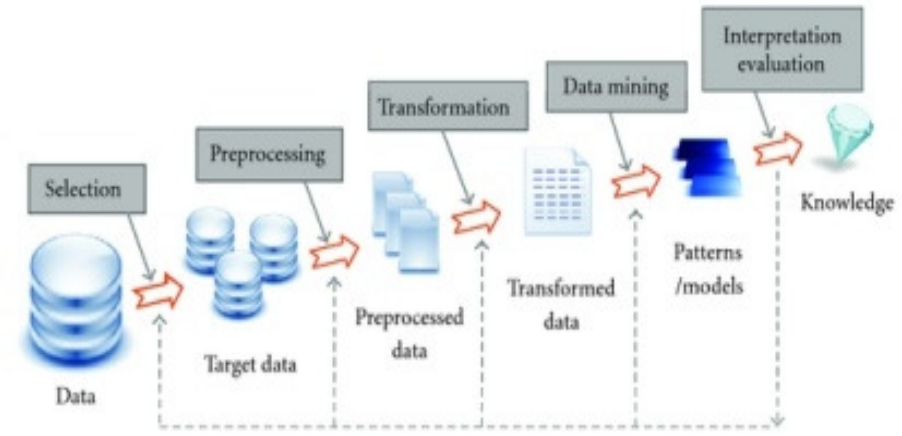


Figure 1: Knowledge Discovery Process [13]

1.2 Data Mining Applications

Data Mining has been successfully applied in different types of fields ranging from military to business to science to sports. Some of the common applications of data mining are as follows:

1.2.1 Business

A lot of organizations make use of data mining techniques to maintain a competitive edge over their rival organizations, for retail data analysis, database-based marketing and stock selection. By implementing mining on customer databases, patterns and trends are extracted, on the basis of which customer profiles are built. These profiles are then used for marketing purposes. For example, the shopping patterns of a customer can be found out using data mining and this information can be used in a sales campaign to promote the products most purchased by the customers. Stock selection is done with the help of data mining by building models and finding trends which is then used to predict the performance of stocks.

1.2.2 Medical Science and Healthcare

Data mining has a wide scope in the medical sciences domain, it can help in proper diagnosis of diseases, profiling of patients, health care and patient history generation. It has also been effectively used in the diagnosis of lung abnormality, and tumors that may be cancerous or benign, and helped in retrieving data to improve the current healthcare systems and their policies. The data mining algorithms significantly reduce the cost of diagnosis and the risk involved for the patient.

1.2.3 Education Systems

Many universities worldwide offer varying degrees of education. The number of students enrolled is growing on a daily basis. Data mining technology can help in predicting student trends and help in reducing the knowledge gap in higher educational systems. The hidden trends, patterns, associations, and anomalies that are discovered by data mining implemented on educational data can improve the decision making processes in such systems. This improvement can provide advantages such as maximizing the system efficiency, decreasing the rate of drop-out of students, increasing retention rate of students, increasing the learning outcome of students and reducing the cost of overall system processes. The extracted knowledge from educational data can be used to enhance and improve the quality of education.

1.3 Data Mining techniques

The importance of data mining in various fields has led to the emergence of many implementation techniques and algorithms. The following table shows some of the existing techniques and the associated algorithms.

Technique	Algorithms
Classification	ZeroR, OneR, Naive Bayesian, Decision Tree
Regression	Multiple Linear Regression, K Nearest Neighbors, Artificial Neural Network, Support Vector Machine.
Clustering	Agglomerative, Divisive, K Means, Self-Organizing Map
Association Rules	Apriori, SETM, AIS, AprioriTid, AprioriHybrid

Table 1 [16]

We elaborate the Classification techniques and two algorithms: OneR and ZeroR in this section, then we apply both algorithms in the last section and contrast the obtained results.

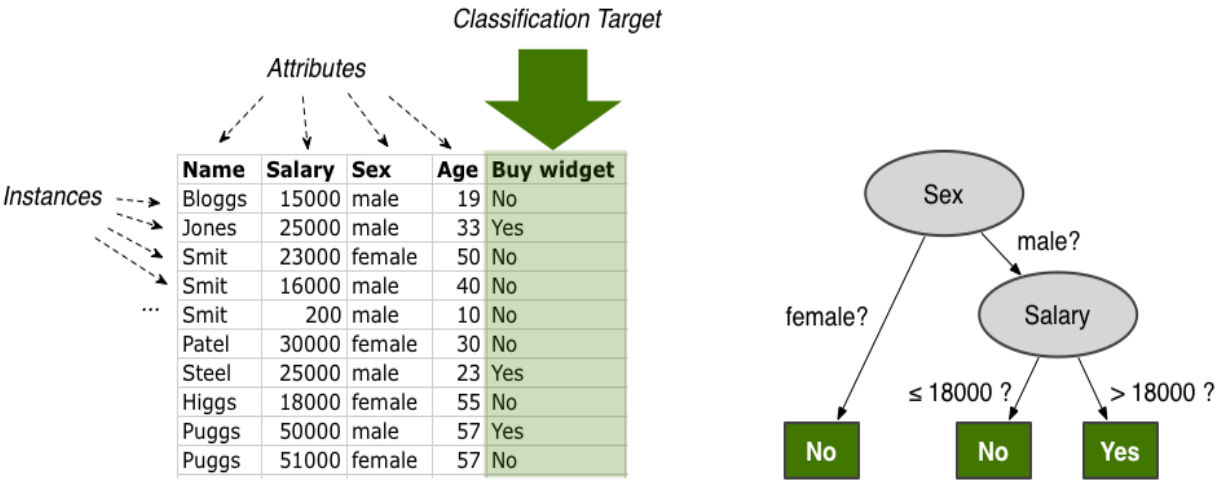
1.3.1 Classification

Classification is a data mining technique which has the objective to assign an object into one of several categories or classes depending on its other known classes. The class that is to be predicted is called the target class. Classification in data mining is similar to normal classification process that we do in our daily lives. An example of classification is when teachers classify their students' educational level according to their attendance, concentration in class, assignments results...etc. We can consider that those attributes are the explanatory attributes that help the teacher predict the educational level, which is the target attribute [6].

More formally, the classification predicts the value of a target class which is of categorical type from other categorical or numerical classes using a classifier. A categorical class is one which has a number of fixed discrete values (i.e.: yes or no, adult or young, weekly or monthly or daily...etc.) while the numerical value is one that can take any value in a finite or infinite range (i.e.: age, height, volume...etc.) [5]

Classifier

The first step is the learning step, where a portion of the available data consisting of objects with known values for all classes is treated to build the classifier; this set of data is called the training set [7]. The following figure explains how the classifier is built and a possible outcome of rules



that would be used to predict the target class.

Fig. 2: [15]

In figure 2, the target class is the "Buy widget" column (e.g.: that it is of our interest to predict). By examining the other classes, the rule on the right of the figure is derived. The rule indicates that the person is expected to buy the widget if he is a male with salary greater than 18,000.

Testing the classifier

Another set of data known as the test set is used to test the accuracy of the classifier. This data also provides values for all classes including the target class, but here the classifier uses the explanatory attributes to predict the target class, then the predicted value is compared to the actual value. According to the percentage of correct predictions the classifier can be accepted as satisfactory or rejected [7]. To formally express the prediction power of a classifier, a confusion matrix [16] is built from the true and false predictions. The size of the matrix is MxM where M is the number of values of the target class. The following figure shows a confusion matrix for a target class of two values: positive or negative.

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Fig.3 [16]

The model is the classifier and the Target is the target class. a is the number of positive values predicted to be positive, b is the number of negative values predicted to be positive, c is the negative values predicted to be positive and d is the negative values predicted to be negative. We are interested in the Accuracy measure shown in the table to determine the classifier's prediction power.

Challenging issues

a) Data Cleaning Due to the large sums of data, values can have some errors and invalid values known as noise or missing values. The noise can be treated using smoothing techniques and the missing values can be replaced by most occurred value [7].

b) Choosing Training/Test sets available data used to build the classifier and test it might not be sufficient. A rule of thumb is to use two thirds of the data for training and one third for testing. In the case study, we will show that a good classifier can be built by less proportion of data and can have an acceptable representation of the whole data.

1.3.1.1 ZeroR

ZeroR or also known as Zero Rule is an algorithm of the classification technique, it is the simplest and most naïve algorithm as it does not use any of the explanatory classes to predict the target class. Simply, it predicts the majority value of the target class for all objects.

Algorithm

A frequency table of the target class is constructed from the available data. A frequency table is simply a count of each value of the class. The following figure shows a simple example to clarify the idea.

Fig.4 [16]

Figure 4 shows a table with the explanatory or predictor classes are describing the climate status. The target class is the decision of playing golf. For the ZeroR algorithm, the frequency table on the right is constructed with the number of yes decisions (9) and the number of no decisions (5). And the rule of the classifier is to predict yes for all records without taking into consideration the predictor classes. Though it is a very simple algorithm with no predictive power, it is useful to provide a benchmark for other classifiers so as to know what is the least acceptable accuracy for any classifier [16] [17].

1.3.1.2 OneR

OneR or also known as One Rule, as the name suggests it uses only one of the explanatory classes to predict the target class. The classifier is built by choosing the class which best describes the target class; it generates rules such as the one in figure 2 for each of the explanatory classes then chooses the class that has the least error when applying its rule to predict the target class.

Algorithm Similarly to the ZeroR we construct a frequency table for each of the predictors against the target class. Using the same example in figure 4, the frequency tables for the predictor classes would be as shown in figure 5.

Frequency Tables

★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Fig.5 [16]

To choose the best predictor class, for each value of a predictor we sum up the frequencies of the dominant target value; for Humidity predictor for instance we add: 4 (frequency of “no” when the value is “high”) and 6 (frequency of “yes” when the value is “normal”). We repeat this for all predictors and the best predictor is the one with largest sum. In this example the best predictor is Outlook with a sum of 10. And the rule for this predictor is: If sunny or overcast predict yes, if rainy predict no. In the case study discussed later in the report we will see how to choose the explanatory class that best describes the target class from a practical programming point of view [16] [17].

2. MapReduce

2.1 Definition and Conceptual view

Google MapReduce [21] is an emerging parallel programming technique designed to process big data, present in clusters in parallel using commodity hardware. It is a programming model and has a framework that attempts to introduce a level of abstraction in the computation of large scale data, by hiding the system levels details from the developers. MapReduce libraries are available in many programming languages: java, Ruby, Python, C++ [1].

MapReduce programming is similar to functional programming languages like LISP and ML, it has 2 stages: Map and Reduce.

Map: Data is read, filtered and processed in small discrete 'chunks' in parallel. The input data is split and converted into another sorted dataset with the individual elements as tuples or key/value pairs.

Reduce: Output from the Map stage is taken as the input, to perform a summary/aggregation operation. The input data is combined into a smaller set of tuples.

2.2 Hadoop

Hadoop [22] is an open source software framework of Apache that enables distributed processing of big data using simple programming models. It is scalable up to thousands of machines, each of which offers local storage and computational capabilities. The hadoop libraries are capable of detecting and handling failures at application level rather than relying on hardware.

The hadoop framework consists of the following modules:

Hadoop Common: This module contains utilities and libraries required by the other Hadoop modules.

Hadoop Distributed File System (HDFS): It is distributed file system that stores data on the commodity machines and provides high-throughput access to the application data [23].

Hadoop YARN: A framework for resource management in clusters and job scheduling.

HadoopMapReduce: A model for parallel processing of big data sets.

The two primary and important components at the core of Apache Hadoop are the Hadoop Distributed File System (HDFS) and the MapReduce parallel processing framework [23]. We will see them in detail below.

Hadoop Distributed File System (HDFS): HDFS is a portable, scalable, and distributed file-system written in Java for the Hadoop framework [21]. In a Hadoop instance each node has a single name node, and a cluster of data nodes form the HDFS cluster. Name nodes manage the file system namespace and regulate the client access to files and data nodes store data as blocks within files [23].Using a block protocol that is specific to HDFS, every data node serves blocks of data over the network. The TCP/IP layer is used for communication by the file system and RPC (Remote Procedure Call) is used for communication between clients.

Hadoop Architecture:

Figure 10 illustrates the Hadoop architecture in detail. HDFS implements a master/slave design with the name nodes as master and data nodes as slaves.

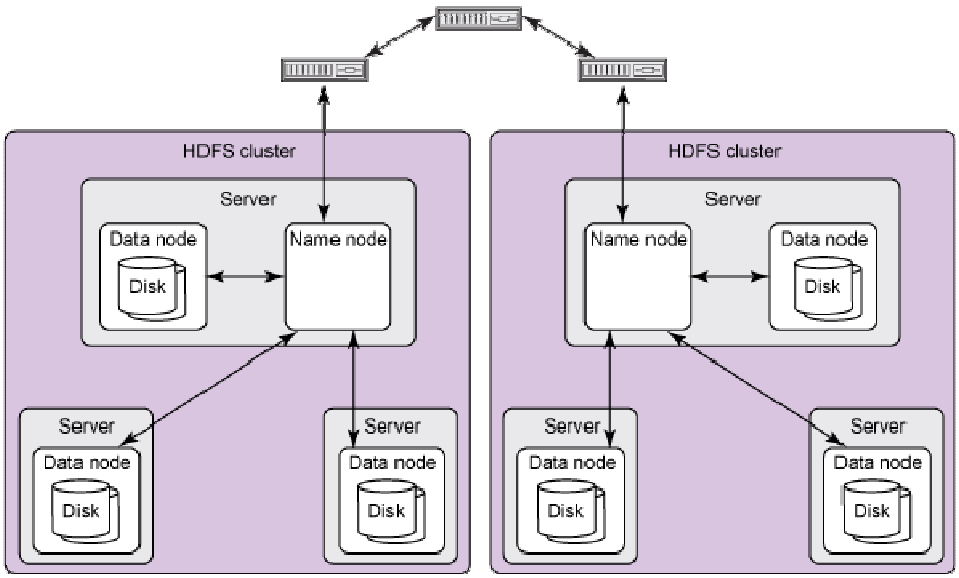


Figure 10: HDFS Architecture [23]

Hadoop MapReduce:

MapReduce is a software framework that enables developers to easily write applications that process large amounts of data up to terabytes in a parallel manner on large clusters containing thousands of nodes of commodity hardware in a reliable and fault-tolerant manner [24].

In a MapReduce job, the input data set is generally split into separate chunks that are parallel processed by the map function. The map output is then sorted by the framework, later used as input to the reduce function. Generally, the input and the output are stored in a file-system. The MapReduce framework manages scheduling, monitoring and re-execution of the failed tasks. Most of the computing task is done on nodes with data on the local disk itself to reduce the network traffic. On successful completion of the scheduled tasks, the data is collected and reduced by the cluster to form a proper result which is sent back to the Hadoop server.

The MapReduce works as follows:

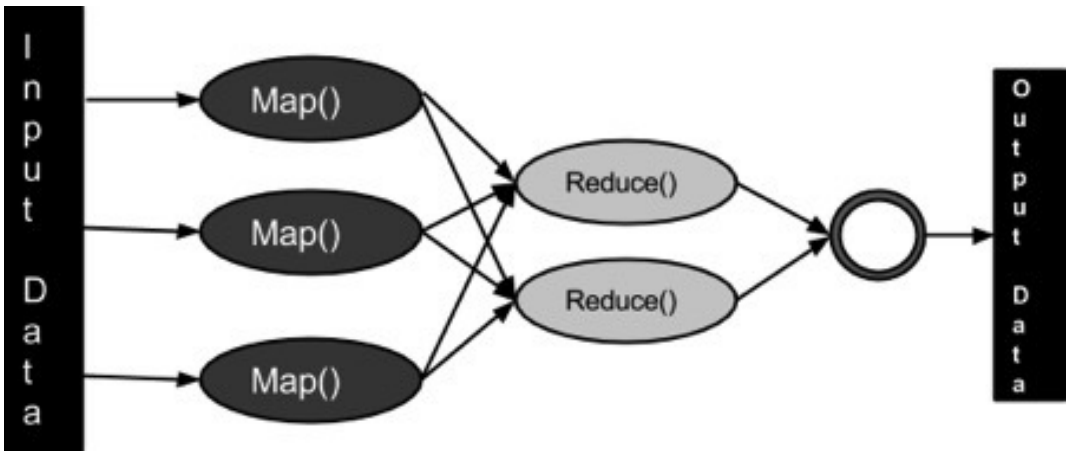


Figure 13: MapReduce [24]

3. Environment Setup

It is arguably one of the hardest steps in the implementation of MapReduce. With our choice of Hadoop, the most appealing environment is to have a platform with a GUI and necessarily running JVM. Then installing an IDE such as Eclipse and plugging in the MapReduce. This is a very complex task, that is why we installed a virtual machine [18] used for learning purposes, and launched it using virtual box.

4. Practical approach for Data Mining Implementation with MapReduce

With the environment set we introduce in this section the following steps to implement the MapReduce program. The implementation is on a single node cluster because of its simple configuration requirements. But with MapReduce, the program for one or few nodes is easily scalable to hundreds or thousands of nodes in a transparent fashion.

4.1 Case Study

The first step is to find a data set and a need for prediction to implement data mining. The dataset used in our work [19] [20] is of a Portuguese bank available to public for research purposes. The file is composed of records (45,211), each record holds data about the bank's clients: age, job, education. The last class, which is the target class, is of categorical type with values yes or no, which indicates whether or the client accepted the product (bank term deposit) offered by the bank. Our objective is to analyze the data to be able to predict the target class or the customer's decision using the client's data (explanatory classes). The file is in csv format and can be found on the link [20] in the references. The file bank-names.txt explains the classes and their values in details. Next we introduce how we applied the ZeroR and oneR algorithms.

4.2 Applying Data Mining algorithms

In the following table we breakdown the algorithms by explaining what is happening in each phase: driver, mapper and reducer.

Point of comparison	**ZeroR**	**OneR**
Driver	Normal configuration for mapper, reducer classes. Input class format is TextInputFormat. Output key is Text, Output value is float writable.	
Mapper	**Output key** It is not of use in the reducer so we just output the text "target" for each record. **Output value** The value of the target class	**Output key** The name of each the predictor class eg. "education"," job" … for each record. **Output value** text of two strings separated by space: value of predictor and value of target class.
Unit Test record	57;entrepreneur;married;secondary;no;2971;no;no;cellular;17;nov;361;2;188;11;other;no	
Unit Test outcome	Target no	age average no
Reducer	Count yes and no count and outputs the percentage of the value with the higher percentage.	Count yes and no for each value of each class. Outputs a rule for each predictor expecting the target value of the higher count for each value of the predictor. Outputs the accuracy for each rule if applied the data. The rule with the highest accuracy is the concluded rule of the algorithm.
Unit Test outcome	Yes 60.0	AGE: young => yes average => no old => no 0.5871424

Table 2

Table 2 shows the logic used in each algorithm, now we will examine the results and detail the experiments done on data using the OneR algorithm.

4.3 Comparing Results

ZeroR results

ZeroR learning algorithm was run on the whole data, it gave an accuracy of 53.97 %. As mentioned before, this percentage is useful as a benchmark. With this result in hand, we now examine the results of the OneR with the capability of judging how good it is, compared to the benchmark.

OneR results

We divided the implementation of OneR into two parts: building the classifier and testing it.

a) Building the classifier

This phase is the one explained in table 2. We wanted to examine what is the least possible portion of data can be used to build the classifier. Generally, the more data used for learning the more accurate the classifier would be as it learns from a wider range of data. In our search for a good threshold to build a classifier with good accuracy we started by learning from 10% of the data. Next, we use the rule obtained and modify the reducer to predict the target class according to the rule then check the prediction against the actual value on the remaining 90% of the data and outputs the accuracy. We repeat the same procedure for 20%, 30% of the data until the accuracy reached is satisfactory and remains almost constant with the increase of data used in learning. The following table shows the obtained results.

Training set %	**Dataset %**	**Rule**	**Accuracy**
10	90	Job: If Student => yes, otherwise => no	53% (lower than ZeroR, totally rejected)
20	80	Job: management, student => yes, otherwise => no	69 % (Satisfactory)
30	70	Job: management, student, retired => yes, otherwise => no	74% (Good)
40	60	Job: management, student, retired => yes, otherwise => no	74% (Good and the same as the previous experiment)

Table 3

As can be seen from table 3, using only 10% of the data produced a rule that is not acceptable as its accuracy is less than ZeroR. Increasing the proportion of data used in building the classifier increases accuracy. We noticed that there is no change between 30% and 40% so we wanted to ensure that learning from 30 % of the data provides a rule that forms a good representation of the data. Thus, we repeated the OneR algorithm for another 30% of random data and the same rule was derived for four trials, where we ensured that each record was used at least once in building the classifier. Consequently, we were able to derive a rule using only 30% of the data which provides a good representation of the whole data as it gives an accuracy which is much higher than the accuracy obtained from ZeroR.

5. Conclusion

As illustrated in the work, data mining proves to be a very powerful process to be applied in business entities to help make efficient decisions. In our case study, the bank will concentrate more on customers satisfying the rule obtained from OneR, so that higher percentages of offers are accepted. In real life, the data available are much bigger which makes the learning process even stronger and more precise. Moreover, MapReduce model is a suitable tool to implement data mining, because of its parallel nature and ability to perform efficiently in a distributed environment. Its main strength points are: the model’s applicability to almost all data analysis problems, handling distributed systems challenges implicitly, sending programming tasks to where data resides in the cloud, rather than the opposite which minimizes network traffic. Finally, the fields of data-mining, cloud computing and map reduce are very active in today's industry, therefore, enhancing one-self's technical skills in those domains is a huge advantage.

Reference

[1] White, Tom. *Hadoop: The definitive guide*. "O'Reilly Media, Inc.", 2012.

[2] Hao, Chen, and Qiao Ying. "Research of Cloud Computing based on the Hadoop platform." In *Computational and Information Sciences (ICCIS), 2011 International Conference on*, pp. 181-184. IEEE, 2011.

[3] Chu, Cheng, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y. Ng, and Kunle Olukotun. "Map-reduce for machine learning on multicore." Advances in neural information processing systems 19 (2007): 281.

[4] Zhou, Lijuan, Hui Wang, and Wenbo Wang. "Parallel implementation of classification algorithms based on cloud computing environment." *TELKOMNIKA Indonesian Journal of Electrical Engineering* 10.5 (2012): 1087-1092.

[5] Sayas, S. Classification http://www.saedsayad.com/classification.htm May 2015

[6] Hämäläinen, Wilhelmiina, and Mikko Vinni. "Classifiers for educational data mining." *Handbook of Educational Data Mining, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series* (2010): 57-71.

[7] Data Mining- Classification and Prediction http://www.tutorialspoint.com/data_mining/dm_classification_prediction.htm 2015

[8] Agrawal, Rakesh, and Ramakrishnan Srikant. "Mining sequential patterns." *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*. IEEE, 1995.

[9] Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases." *ACM SIGMOD Record*. Vol. 22. No. 2. ACM, 1993.

[10] Agrawal, Rakesh, and Ramakrishnan Srikant. "Fast algorithms for mining association rules." *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. 1994.

[11] Han, Jiawei, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation." *ACM SIGMOD Record*. Vol. 29. No. 2. ACM, May 2000.

[12] Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Vol. 1. Boston: Pearson Addison Wesley, 2006.

[13] Fayyad, U. "Data Mining and Knowledge Discovery: Making Sense Out of Data" in IEEE Expert October 1996 pp. 20-25

[14] Han, Jiawei, Micheline Kamber, and Jian Pei. *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.

[15] Data Requirements for process mining http://fluxicon.com/blog/2012/02/data-requirements-for-process-mining/ Feb 2015

[16] Sayad, S. An Introduction to data mining http://www.saedsayad.com/ , June 2015

[17] Sadawi, N. Classification , https://www.youtube.com/watch?v=SAUIDEhGC8w, August 2014

[18] MapReduce Academy https://training.mapr.com/ June 2015

[19] Moro, Sérgio, Raul Laureano, and Paulo Cortez. "Using data mining for bank direct marketing: An application of the crisp-dm methodology." (2011).

[20] P. Cortez and M. Embrechts. Using Sensitivity Analysis and Visualization Techniques to Open Black Box Data Mining Models. In Information Sciences, Elsevier, In press, ISSN 0020-0255. http://dx.doi.org/10.1016/j.ins.2012.10.039 March 2013

[21] Sachin P Bappalige Introduction to Apache Hadoop http://opensource.com/life/14/8/intro-apache-hadoop-big-data August 2014

[22] Apache Hadoop, https://hadoop.apache.org/, June 2015

[23] J. Jeffrey Hanson http://www.ibm.com/developerworks/library/wa-introhdfs/ Feb 2011

[24] Hadoop Mapreduce ://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm June 2015