Journal of Applied Mathematics and Computation (JAMC), 2018, 2(2), 50-57
http://www.hillpublisher.org/journal/jamc
ISSN Online:2576-0645
ISSN Print:2576-0653

Performance analysis of classification Algorithms: A case study of Naïve Bayes and J48 in Big Data

Festim Halili[1], Festim Kamberi[2]

[1]Department of Informatics State University of Tetovo, SUT Tetovo, Macedonia, Email: festim.halili@unite.edu.mk
[2]Department of Computer Engineering International Balkan University, IBU Skopje, Macedonia, Email: kamberifestim@gmail.com

How to cite this paper: Halili, F. & Kamberi, F. (2018) Performance analysis of classification Algorithms: A case study of Naïve Bayes and J48 in Big Data. *Journal of Applied Mathematics and Computation*, **2(2)**, 50-57.
http://dx.doi.org/10.26855/jamc.2018.02.003

Corresponding author: Festim Kamberi, Department of Computer Engineering International Balkan University, IBU Skopje, Macedonia,
Email: kamberifestim@gmail.com

Abstract

In the world of technology the term Big Data has emerged with new opportunities and challenges to deal with the massive amount of data. Big Data is a collection of massive and complex data sets and data volume that include the huge quantities of data, data management capabilities, social media analytics and real-time data. To find the useful information from massive amount of data to organizations, businesses, companies, vendors, we need to analyze and classify the data. Initially, in this paper we have provided an in-depth analysis of 5Vs characteristics of Big Data. In addition, we have used a comparison methodology for the two well-known classifications Naïve Bayes and J48 decision tree algorithms by using QoS parameters: accuracy, sensitivity and specificity. These results will help us deriving the conclusion for which of two algorithms are the best.

Keywords

Big Data, Hadoop, Map Reduce, HDFS, Naïve Bayes, J48, Weka.

1. Introduction

Big Data can be structured, unstructured or semi-structured. Data can be generated on web in various forms like texts, images or videos or social media posts. Big Data is something so huge and complex that it is impossible for traditional systems and traditional warehousing tools to process and work on them. Data is generated by machines, generated by humans and also generated by mother nature. Big Data can neither be worked upon by using traditional SQL like queries nor can the relational database management system (RDBMS) be used for storage [2][3]. Hadoop, an open source distributed data processing system is one of the prominent and well known solutions. [4] The need of Big Data comes from the enterprises and vendors for the purpose of the analysis of big amount of data which is in unstructured form.

1.1. Characteristics

Big Data can be described by the following characteristics, see Figure 1:

1.1.1. Volume

Volume refers to amount of data. Volume of data stored in enterprise repositories have grown from megabytes and gigabytes to petabytes. 40 zettabytes of data will be created by 2020 which is 300 times from 2005. The social networking sites existing are themselves producing data in order of terabytes every day and this amount of data is definitely difficult to be handled using the existing traditional system. [1]

1.1.2. Velocity

Velocity in Big Data is a concept which deals with the speed of the data coming from various sources and the speed at which data moves around. [1]

1.1.3 Variety

The next aspect of Big Data is its variety. Big Data is not always data and it is not always easy to put big data into a relational database. Dealing with a variety of structured and unstructured data greatly increase the complexity of both sorting and analyzing Big Data. [2]

1.1.4. Veracity

When we are dealing with a high volume, velocity and variety of data, it is not possible that all of the data is going to be 100% correct, there will be dirty data. The quality of data being captured can vary greatly. The data accuracy of analysis depends on the veracity of the source data. [3]

1.1.5. Value

Value is the most important aspect in the big data. Though, the potential value of the Big Data is huge. It is all well and good having access to big data but unless we can turn it into value it is become useless. [4]

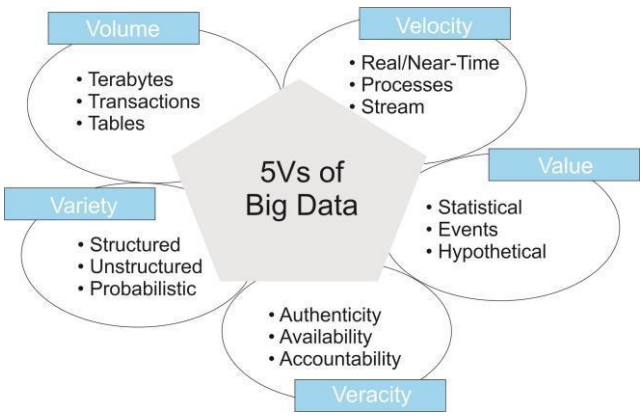

Figure 1. 5Vs of Big Data

2. Hadoop: Solution for Big Data Processing

Hadoop is a Programming framework used to support the processing of large data sets in a distributed computing environment. [4] Hadoop was developed by Google's MapReduce that is a software framework where an application breaks down into various parts. The currect Apache Hadoop ecosystem consists of the Hadoop Kernel, MarReduce, HDFS and numbers of various components like Apache Hive, Base and Zookeper. HDFS and MapReduce are explained in the following points, see Figure 2.

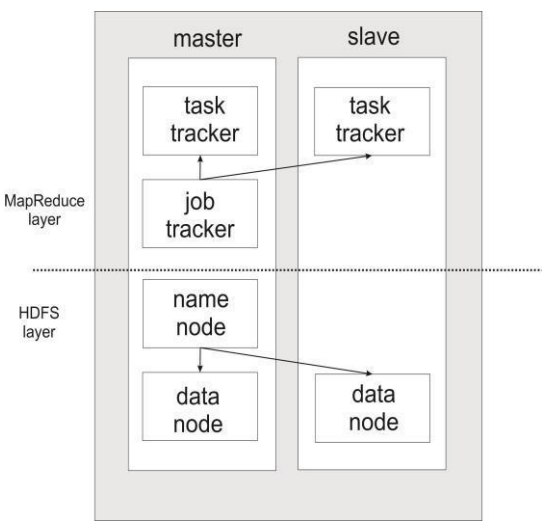

Figure 2. Hadoop Architecture

2.1. HDFS Architecture

Hadoop includes a fault-tolerant storage system called Hadoop Distributed File System, or HDFS. HDFS is able to store huge amounts of information, scale up incrementally and survive the failure of significant parts of the storage infrastructure without loosing data. Hadoop creates clusters of machines and coordinates work among them. Clusters can be built with inexpensive computers. If one fails, Hadoop continues to operate the cluster without loosing data or interrupting work, by shifting work to the remaining machines in the cluster. HDFS manages storage on the cluster by breaking incoming files into pieces, called "blocks" and storing each of the blocks redundantly across the pool of servers. In the common case, HDFS stored three complete copies of each file by copying each piece to three different servers as shown in Figure 3.

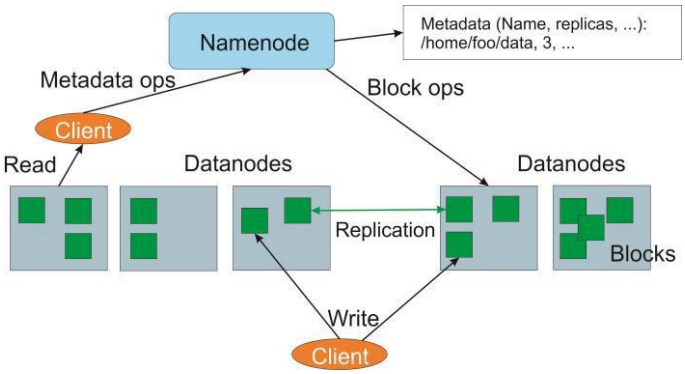

Figure 3. HDFS Architecture

2.2. MapReduce Architecture

The processing pillar in the Hadoop ecosystem is the MapReduce framework. The framework allows the specification of an operation to be applied to a huge data set, divide the problem and the data, and run it in parallel, shown in Figure 4. From an analyst's point of view, this can occur on multiple dimensions. For example, a very large dataset can be reduced into a smaller subset where analytics can be applied. In a traditional warehousing scenario, this might entail an ETL operation on the data to produce something usable by the analyst. In Hadoop, these kind of operations are written as MapReduce jobs in Java. There are number of higher level languages like Hive and Pig that make writing these programs easier. There are two functions in MapReduce as follows:

- **map** – the function takes key/value pairs as input and generated an intermediate set of key/value pairs
- **Reduce** – the function which merges all the intermediate values associated with the same intermediate key

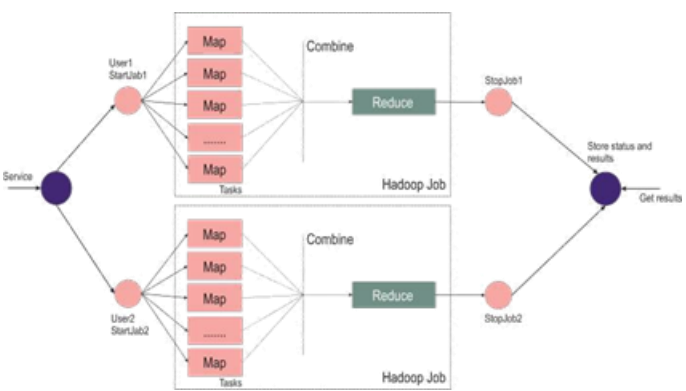

Figure 4. MapReduce Architecture

3. Related Work

[1] illustrated that in olden days data was less and easily handled by RDBMS but recently it is difficult to handle huge data through RDBMS tools, which is preferred as "big data". In this they told that big data differs from other data in 5 dimensions such as volume, velocity, variety, value and complexity. They illustrated the hadoop architecture consisting of name node, data node, HDFS to handle big data systems.
[2] enhanced us with the knowledge that Big Data is combination of stuctured, semi-structured homogenous and heterogeneous data. The author suggested to use nice model handle transfer of huge amount of data over the network. Under this model, these transfers are relegated to low demand periods where there is ample, idle bandwidth available. This bandwidth can then be repurposed for big data transmission without impacting other users in system.
[3] used Hadoop which is currently the large – scale data analysis " hammer" of choice, but there exists classes of algorithms that aren't " nails" in the sense that they are not particularly amenable to the MapReduce programming model . He focuses on the simple solution to find alternative non-iterative algorithms that solves the same problem.
[5] introduced Big Data Mining as the capability of extracting useful information from these large datasets or streams of data that due to its volume, variability and velocity it was not possible before to do it.
[4] stated that streaming data analysis in real time is becoming the fastest and most efficient way to obtain useful knowledge, allowing organizations to react quickly when problem appear or detect to improve performance. Huge amount of data is created everyday termed as "big data". The tools used for mining big data are apache hadoop, apache big, cascading, scribe, storm, apache hbase, apache mahout, MOA, R, etc. Thus, he instructed that our ability to handle many exabytes of data mainly dependent on existence of rich variety dataset, technique, software framework.

4. Methodology and QoS for data classifications

In this paper, two classifiers, Naïve Bayes algorithm and J48 decision tree algorithm are used for comparison. Comparison is made on accuracy, sensitivity and specificity using true positive and false positive in confusion matrix generated by the respective algorithms. In this paper, we have used weka (Waikato environment for knowledge analysis) tool for comparison of naïve bayes and J48 algorithm and calculating efficiency based on accuracy regarding correct and incorrect instances generated with confusion matrix. We have used here bank-data-train.arff for data classification available on web URL http://www.cs.bme.hu/~kiskat/adatb/bank-data-train.arff . This bank relation consists of attributes age, gender, region, income, married, children, car, mortgage, pep with 300 instances.

4.1. Decision tree algorithm J48:

J48 classifier is a simple C4.5 decision tree for classification. It creates a binary tree. The decision tree approach is most useful in classification problem. With this technique, a tree is constructed to model the classification process. Once the tree is built, it is applied to each tuple in the database and results in classification for that tuple. One example of decision tree is shown in the Figure 5.

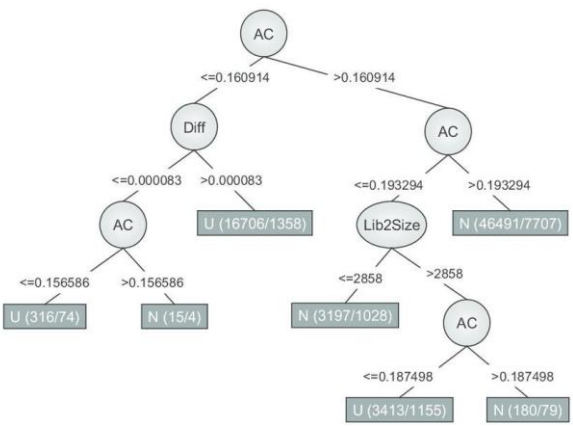

Figure 5. Example of a decision tree generated by the induction algorithm J48, available in the Weka program

Algorithm J48:

```
INPUT:
    D    //Training Data
OUTPUT:
    T    //Decision Tree
DTBUILD(*D)
{
T=φ;
T=Create root node and label with splitting attribute;
T=Add arc to root node for each split predicate and label;
For each arc do
    D=Database created by applying splitting predicate to D;
    If stopping point reached for this path, then
        T'=create leaf node and label with appropriate class;
    Else
        T'=DTBUILD(D);
    T=Add T' to arc;
}
```

While building a tree, J48 ignores the missing values i.e. the value for that item can be predicted based on what is known about the attribute values for the other records. The basic idea is to divide the data into range based on the attribute values for that item that are found in the training sample. J48 allows classification via either decision trees or rules generated from them.

4.2. Na ïve Bayes classifier

The Na ïve Bayes algorithm is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. The algorithm uses Bayes theorem and assumes all attributes to be independent given the value of the class variable. This conditional independence assumption rarely holds true in real world applications, hence the characterization as Na ïve yet the algorithm tends to perform well and learn rapidly in various supervised classification problems. Na ïve Bayesian classifier is based on Bayes' theorem and the theorem of total probability The probability that a document d with vector x=<x_1,…,> belongs to hypothesis:

$$\mathrm{P}(h_1|x_i) = \frac{\mathrm{P}(x_i|h_i)\mathrm{P}(h_1)}{\mathrm{P}(x_i|h_1)\mathrm{P}(h_1) + \mathrm{P}(x_i|h_2)\mathrm{P}(h_2)}$$

Here, $(h_1|x_i)$ is posterior probability, while (h_1) is the prior probability associated with hypothesis h_1. For m different hypothesis, we have:

$$\mathrm{P}(x_i) = \sum_{j=1}^{n} \mathrm{P}(x_i|h_j)\mathrm{P}(h_j)$$

Thus, we have:

$$\mathrm{P}(h_1|x_1) = \frac{\mathrm{P}(x_i|h_1)\mathrm{P}(h_1)}{\mathrm{P}(x_i)}$$

4.3. Confusion matrix

A confusion matrix illustrates the accuracy of the solution to a classification problem. Given n classes a confusion matrix is a $m\ x\ n$ matrix where $C_{i,j}$ indicates the number of tuples from D that were assign to class $C_{i,j}$ but where the correct class is C_i.

5. Methodology and experimental work

We have performed classification using Na ïve Bayes algorithm and J48 decision tree algorithm on bank-data-train.arff dataset in weka tool provide inbuilt algorithms for Na ïve Bayes and J48.

5.1. Results for classification using J48:

Mortgage attribute has been chosen randomly for bank data set. J48 is applied on the data set and the confusion matrix is generated for class gender having two possibilities i.e. YES or NO, see Table 1.

Table 1. Confusion Matrix using J48 Algorithm

	YES	NO
a	33	25
b	72	170

For above confusion matrix, true positives for class a=”YES” is 33 while false positives is 72 whereas, for class b=”NO”, true positives is 170 and false positives is 25 i.e. diagonal elements of matrix 33+170=203 represents the correct instances classified and other elements 25+72=97 represents the incorrect instances. The true and false positive rate for class *a* and *b* using J48 algorithm are shown in the Table 2.

$$\text{True positive rate} = \frac{\text{diagonal element}}{\text{sum of relevant row}}$$

$$\text{False positive rate} = \frac{\text{non} - \text{diagonal element}}{\text{sum of relevant row}}$$

Table 2. True and False positive rate

	TP rate	FP rate
a	0.314	0.128
b	0.871	0.685

For above confusion matrix, true positive rate for class *a* is 0.314 while false positive rate is 0.128, whereas, true positive rate for class b is 0.871 while false positive rate is 0.682 i.e. diagonal elements of matrix 0.314+0.685=0.999 represents the correct instances classified and other elements 0.871+0.128=0.999 represents the incorrect instances. The precession for class *a* and *b* using J48 algorithm is shown in the Table 3.

$$\text{Precision} = \frac{\text{diagonal element}}{\text{sum of relevant column}}$$

Table 3. Precession and F-Measure

	Precession	F-Measure
a	0.568	0.404
b	0.702	0.778

For above confusion matrix, the precision for class a is 0.568 while F-Measure is 0.128, whereas, the precision for class b is 0.702 while F-Measure is 0.778 i.e. diagonal elements of matrix 0.568+0.778=1.346 represents the correct instances classified and other elements 0.702+0.404=1.106 represents the incorrect instances.

5.2. Results for classification using Naïve Bayes:

Here same, Mortgage attribute has been chosen for bank data set. Naïve Bayes is applied on the data set and the confusion matrix is generated for class gender having two possible values i.e. YES or NO, see Table 4.

Table 4. Confusion Matrix using Naïve Bayes Algorithm

	YES	NO
a	10	21
b	95	174

For above confusion matrix, true positives for class a= "YES" is 10 while false positives is 95 whereas, for class b= "NO", true positives is 174 and false positives is 21 i.e. diagonal elements of matrix 10+174=184 represents the correct instances classified and other elements 95+21=116 represents the incorrect instances. The true and false positive rate for class *a* and *b* using Naïve Bayes algorithm are shown in the Table 5.

Table 5. True and False positive rate

	TP rate	FP rate
a	0.095	0.108
b	0.892	0.905

For above confusion matrix, true positive rate for class a is 0.095 while false positive rate is 0.108, whereas, true positive rate for class b is 0.892 while false positive rate is 0.905 i.e. diagonal elements of matrix 0.095+0.095=0.19 represents the correct instances classified and other elements 0.892+0.108=1 represents the incorrect instances. The precession for class *a* and *b* using Naïve Bayes algorithm is shown in the Table 6.

Table 6. Precession and F-Measure

	Precession	F-Measure
a	0.323	0.147
b	0.647	0.75

Though results of cost and benefit analysis for mortgage is same for J48 and Naïve Bayes, but in case of gender cost/benefit analysis for J48 is lesser than that of Naïve Bayes as shown in the table and chart below (Table 7, Figure 6).

Table 7. Analysis of gender cost/benefit

	Cost of J48	Cost of Naïve Bayes
Male	142	155
Female	144	147

For above confusion matrix, Cost of J48 for male is 142 while for female is 144, whereas, cost of Naïve Bayes for male is 155 while for female is 147.

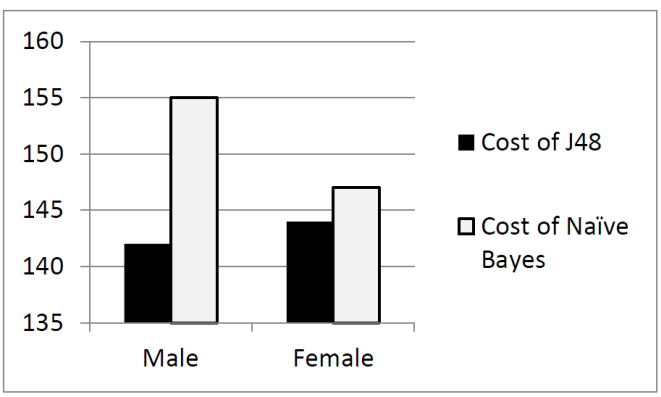

Figure 6. Analysis of gender cost/benefit

6. Conclusion and future work

From above experimental work we can conclude that correct instances generated by J48 are 203 and Naïve Bayes are 184, as well as performance evolution on the basis of matage is as shown in the Table 8.

Table 8. Performance evolution on the basis of mortgage

Classification Accuracy			Cost Analysis	
Mortgage	Naïve Bayes	J48	Naïve Bayes	J48
YES	9 %	31 %	105	105
NO	89 %	87 %	195	195

This proves that the J48 is a simple classifier technique to make a decision tree. Efficient result has been taken from bank dataset using Weka tool in the experiment. Naïve Bayes classifier also shows good results. The experiments results shown in this study are about classification accuracy and cost analysis. J48 gives more classification accuracy for class mortgage in bank dataset having Boolean data type. Though here in this example, cost analysis valued same for both classifications with gender attribute, we can conclude that J48 is cost efficient than the Naïve Bayes classifier.
For future work, we will compare the decision tree algorithm J48 with other classification algorithms.

References

[1] S.V. Phaneendra & E.M. Reddy "Big Data – solutions for RDBMS problems – A survey" In 12th IEEE/IFIP Network Operations & Management Symposium (NOMS 2010) (Osaka, Japan, Arp 19-23 (2013).
[2] K. K. Reddi & D. Indira "Different Technique to Transfer Big Data: survey" IEEE Transactions on 52(8) (Aug.2013).
[3] J. Lin, MapReduce Is Good Enough? The control project. IEEE Computer 32 (2013).
[4] A. Bifet, "Mining Big Data in Real Time" Informatica 37 (2013) 15-20 DEC 2012.
[5] M. J. Berry, G. Linoff, Data Mining Techniques: For Marketing, Sales, and Customer Support, New York: John Wiley & Sons, Inc, 1997.
[6] D. T. Larose, Data Mining Methods and Models, Canada: A John Wiley & Sons, Inc, 2006.
[7] A. Kumar, O. Singh, V. Rishiwal, R. K. Dwivedi, R. Kumar, "Association Rule Mining On Web Logs For Extracting Interesting Patterns Through Weka Tool," International Journal of Advanced Technology In Engineering And Science, vol. 3, no. 1, pp. 134-140, 2015.
[8] C. D., Discovering Knowledge in Data: An Introduction to Data Mining, Canada: John Wiley & Sons, 2014.
[9] S. Rajagopal, "Customer Data Clustering Using Data Mining Technique," International Journal of Database Management Systems, vol. 3, no. 4, pp. 1-11, 2011.
[10] S.V. Phaneendra, E.M. Reddy,"Big Data solutions for RDBMS problems- A survey", In 12th IEEE/ IFIP Network Operations & Management Symposium (NOMS 2010) (Osaka, Japan, Apr 19{23 2013).
[11] Aveksa Inc. (2013). Ensuring "Big Data" Security with Identity and Access Management. Waltham, MA: Aveksa.
[12] Hewlett-Packard Development Company. (2012). Big Security for Big Data. L.P.: Hewlett Packard Development Company
[13] A. Katal, M. Wazid, Goudar, R. H. (2013). Big Data: Issues, Challenges, Tools and Good Practices. IEEE, 404-409.
[14] D. Zhu, Y. Zhang, X. Wang, et al.: Research on the methodology of technology innovation management with big data. Sci. Sci. Manage. S. & T. 4, 172–180 (2013)
[15] Q. Yu, J. Ling, Research of cloud storage security technology based on HDFS. Comput. Eng. Des. 8, 2700–2705 (2013)
[16] B. Huang, S. Xu, W. Pu, Design and implementation of MapReduce based data mining platform. Comput. Eng. Des. 2, 495–501 (2013)
[17] J. Song, X. Liu, Z. Zhu, et al.: An energy efficiency optimized resource ratio model for MapReduce. Chin. J. Comput. 1, 59–73 (2015)
[18] J. Zheng, Y. Ye, T. Tai, et al.: Design of live video streaming, recording and storage system based on Flex, Red5 and MongoDB. J. Comput. Appl. 2, 589– 592 (2014)
[19] M. H. Danham, S. Sridhar, Data mining, Introductory and Advanced Topics, Person education , 1st ed., 2006
[20] W. Lee, S. J. Stolfo, K. W. Mok, A Data Mining Framework for Building Intrusion Detection Models.
[21] F. Halili, A. Dika. Integrated Orchestration of Web Services and the Impact of the Query Optimization. Department of Informatics State University of Tetovo, SUT Tetovo, Macedonia
[22] F. Halili, M. K. Halili, I. Ninka. A new framework of Qos-based web service Discovery and Binding. Department of Informatics State University of Tetovo, SUT Tetovo, Macedonia