IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. IV (Sep – Oct. 2014), PP 06-13
www.iosrjournals.org

Optimization of Horizontal Aggregation in SQL by using C4.5 Algorithm and K-Means Clustering

Ms. Priti Phalak[1], Dr. Rekha Sharma[2] (14)
[1]*(Computer Department, Thadomal Shahani college/ Mumbai University, India)*
[2]*(Computer Department, Thakur college/ Mumbai University, India)*

Abstract: *Datasets in the horizontal aggregated layout are preferred by most of data mining algorithms, machine learning algorithm. Major efforts are required to compute data in the horizontal aggregated format. There are many inbuilt aggregation functions in SQL, namely, minimum, maximum, average, sum and count. These aggregation functions are used with a query evaluation method to retrieve data in the horizontal aggregation format. Optimization techniques used for vertical aggregation is not appropriate for horizontal aggregation. Standard aggregations are hard to interpret when there are many result rows, especially when grouping attributes having high cardinalities. That is why we proposed C4.5 classification algorithm and K-means clustering algorithm with query evaluation method and aggregation function for optimizing horizontal aggregation. Horizontal aggregation is a method which generates SQL code to return aggregated columns in the horizontal tabular layout. It returns a set of numbers instead of one number per row. There are various applications where the horizontal aggregation is used such as electrical billing, banks, hospital management system, pharmacy, and online library etc. [6].*

Keywords: *C4.5 Algorithm, CASE, Horizontal Aggregation, K-means, OLAP, PIVOT, SPJ*

I. Introduction

Data mining is a relatively new field of research whose major objective is to acquire knowledge from large amounts of data. Today, in medical and health care areas, a large amount of data is available in digital format. Healthcare organizations are capable of generating and collecting large amounts of complex data about patients, hospital's resources, diseases, and diagnosis methods, etc. which is not mine to discover hidden information for effective and efficient decision making. This large amount of data needs effective extraction methods. This can be achieved by using data mining techniques. Data mining uses the data warehouse as the source of information for knowledge discovery [3]. In Data warehouse, Online Analytical processing (OLAP) has the capability to provide summarized data from multiple and dynamic view which is a solid foundation for successful data mining.

In general, datasets come from Online Transaction Processing (OLTP) systems and are stored in a relational database (or a data warehouse) where database schemas are highly normalized. Some data mining and machine learning algorithms generally require aggregated data in a summarized format. Major efforts are required to compute aggregation when they are expected in horizontal tabular layout. Horizontal aggregation is a method which generates SQL code to return aggregated columns in the horizontal tabular layout. It returns a set of numbers instead of one number per row. There are a number of inbuilt aggregation functions in SQL provided as for grouping and aggregation purpose, such as minimum, maximum, average, count, and sum. Data mining and OLAP tools are the data summarization/ aggregation tools. These tools are used with some OLAP operation for transposition (pivoting) of results retrieved by data mining. [6]

In literature, horizontal aggregation has been successfully implemented by using three query evaluation methods namely SPJ, CASE and PIVOT to prepare dataset. They conclude that CASE method has similar performance to the PIVOT operator and it is much faster than the SPJ method[1]. The PIVOT and UNPIVOT complementary data manipulation operators or methods are used to exchange the role of rows and columns in a relational table. Pivot transforms a series of rows into a series of fewer rows with additional columns. Unpivot provides the inverse operation, removing a number of columns and creating additional rows that capture the column names and values from the wide form [2]. The K-means clustering algorithm is used to partition data sets after horizontal aggregations [3]

All decision tree algorithms are applied on student's internal assessment data namely ID3,C4.5, SPRINT, CART and SLIQ to predict their performance and the efficiency of various decision tree algorithms can be analyzed based on their accuracy and time taken to derive the tree. From the performance comparision of all algorithm, it is observe that C4.5 is the best algorithm for all datasets among all because it provides better accuracy and efficiency than the other algorithms[5].

In this paper, three query evaluation methods are used with some aggregation function to operate horizontal aggregation of data sets.

Decision Tree C4.5 algorithms are used to optimize the horizontal aggregation in SQL. For efficient retrieval of datasets, C4.5 algorithm classifies the transaction datasets in classified labels. A transactional database consists of a file which contains transaction related information.

K-means is one of the simplest unsupervised learning algorithms used to optimize the horizontal aggregation in SQL. For faster data retrieval, K-means clustering algorithm is used to cluster the classified data.

This paper consists of six sections. Section 2 contains the description of the horizontal aggregation with the help of example, and three query evaluation methods. Section 3 contains proposed system has been discussed Section 4 consists of a description of the optimization algorithms used. Section 5 provides the results obtained after application of various techniques. The conclusion is given in Section 6

II. Horizontal Aggregation

Horizontal aggregation means aggregate the datasets in horizontal layout. It is same as traditional SQL, or standard SQL aggregation, which return set of values in horizontal layout instead of one number per row. It is a new class of aggregate functions that aggregate, numeric expressions and transpose results to produce a data set with a horizontal layout.

There are several advantages of horizontal aggregations as follows:

1) They represent a template to generate SQL code from a data mining tool. This SQL code automates writing SQL queries, optimizing them, and testing them for correctness.
2) SQL code reduces manual work in the data preparation phase in a data mining.
3) SQL code is more efficient than SQL code written by an end user as it is automatically generated. As a result, data sets can be created in less time.
4) The data set can be created entirely inside the DBMS. In modern database environments, it is common to export de-normalized data sets to be further cleaned and transformed outside a DBMS in external tools. Unfortunately, exporting large tables outside a DBMS is very slow and it creates inconsistent copies of the same data and effects database security.

Horizontal aggregations are a small extension of aggregate functions called in a SELECT statement. Alternatively, they can be used to generate SQL code from a data mining tool to construct data sets for data mining analysis[1][6].

Example of Horizontal aggregation in SQL is as follows:

In following example, in OLAP term 'F' is the fact table having a key 'K' represented by an integer. 'PID' is the patient id which is a primary key in F table. Column T indicates transactions done by PID across various modules. 'A' is the aggregated value of billing across each module. Column K will not be used to compute aggregations.

Table I i.e. F table consist of three PID 1, 2 and 3 repeatedly for distinct module in T, which consist four distinct modules OT, BB, Pharmacy, test. A patient whose PID is 1 has made three transactions across different modules, first is in OT module, and second and third is in Pharmacy module. The sum () aggregate operation is used in this example, so the aggregation function sum () applies to second and third transaction. This sum () function adds aggregated values of these rows and store in the vertical aggregated format. This vertical aggregated data FV can be transformed into horizontal layout FH by generating new columns countBB, countOT, countTest and countPharmacy.

Table1: Original Table (F)

Key K	PID	T	Aggregation billing(A)
1	1	OT	6000
2	2	BB	7000
3	3	Test	4000
4	2	BB	2000
5	1	Pharmacy	3000
6	1	Pharmacy	5000
7	3	OT	10000

Table II : dataset is in vertical aggregated format (fv)

PID	T	A(Sum)
1	OT	6000
1	Pharmacy	8000
2	BB	9000
3	Test	4000
3	OT	10000

Table III: dataset is in horizontal aggregated format

PID	Count BB	Count OT	Count Test	Count Pharmacy
1	Null	6000	Null	8000
2	9000	Null	Null	Null
3	Null	10000	4000	Null

III. Query Evaluation Methods

There are three query evaluation methods to evaluate horizontal aggregation

1) SPJ
2) PIVOT
3) CASE

SPJ method based on standard relational operator such as select, project, and join (SPJ) queries. The CASE method based on the SQL CASE construct. PIVOT method uses a built-in operator in a commercial DBMS that is not widely available [1]

A. SPJ Method

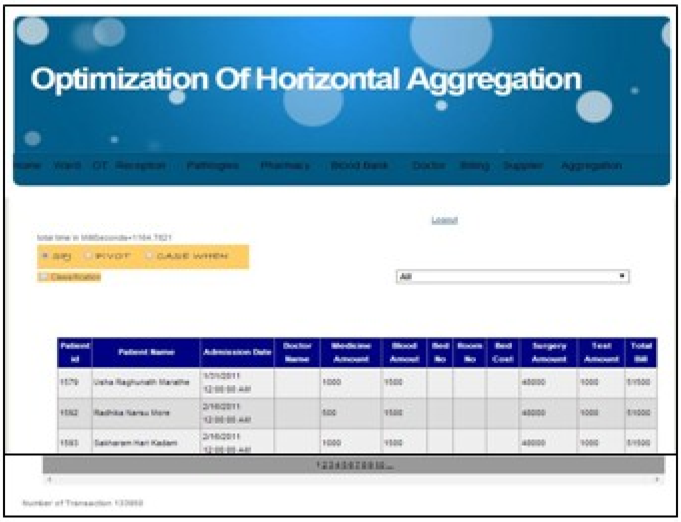
It depends only on relational operations means it only does select, project, join, and aggregation

The main idea of this method is to create one table with a vertical aggregation for each result column, and then join all those tables to produce horizontal aggregation.

The actual implementation is based on the details given in data sets.

Proposed syntax is as follows.

SELECT (L1… Lj), H (A BY R1, … ,Rk)
FROM F
GROUP BY (L1… Lj);[6]



total time in MilliSeconds=1164.7821

● SPJ ○ PIVOT ○ CASE WHEN

□ Classification

All

Patient id	Patient Name	Admission Date	Doctor Name	Medicine Amount	Blood Amout	Bed No	Room No	Bed Cost	Surgery Amount	Test Amount	Total Bill
1579	Usha Raghunath Marathe	1/31/2011 12:00:00 AM		1000	1500				48000	1000	51500
1582	Radhika Narsu More	2/16/2011 12:00:00 AM		500	1500				48000	1000	51000
1583	Sakharam Hari Kadam	2/16/2011 12:00:00 AM		1000	1500				48000	1000	51500

12345678910...

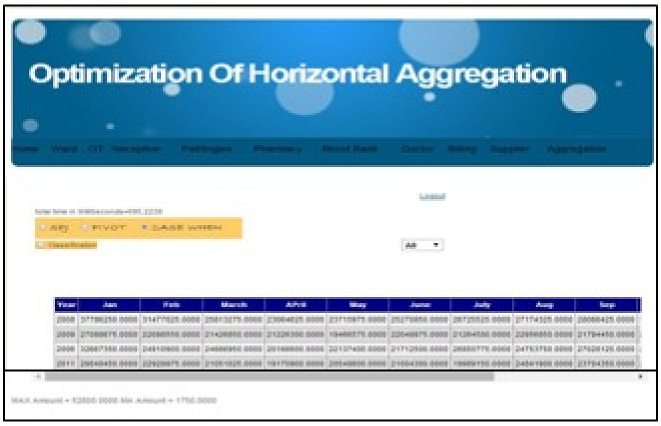
Number of Transaction 133850

Fig 1: SPJ Method

Fig 1 shows that SPJ method is used to retrieve all patient data related to all modules wherever they have done the transaction in hospital.

B. CASE Method

In this method, the "case" programming construct available in SQL is used. A value selected from a set of values based on Boolean expressions is returned by the case statement. From a theory point of view, a relational database is equivalent to doing a simple projection/aggregation query where each non key value is given by a function that returns a number based on some conjunction of conditions [1].


Optimization Of Horizontal Aggregation
Home Ward OT Reception Pathology Pharmacy Blood Bank Doctor Billing Supplier Aggregation

[illegible]

○ SPJ ○ PIVOT ● CASE WHEN

□ Classification

All ▾

Year	Jan	Feb	March	APril	May	June	July	Aug	Sep
2008	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2009	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2011	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

MAX Amount = [illegible] Min Amount = [illegible]

Fig 2:CASE Method

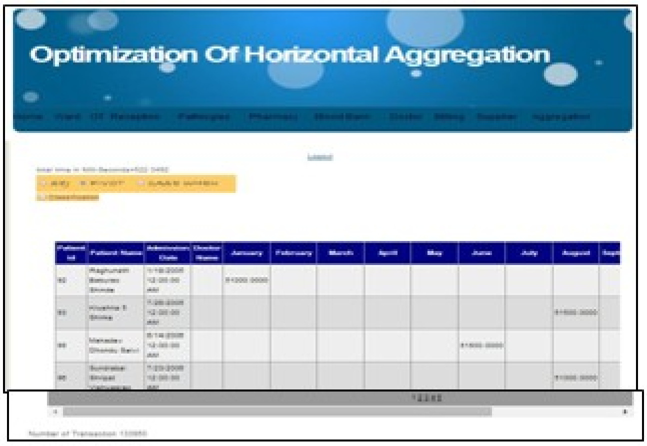
Fig 2 shows all year wise and month wise total transaction amount of all patient's.

C. PIVOT Method

Pivot (also called rotate) is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data. Pivot transforms a series of rows into a series of fewer rows with additional columns.Data in one source column is used to determine the new column for a row, and another source column is used as the data for that new column[2].

In a commercial DBMS, the PIVOT operator is a built-in operator. Since the PIVOT operator can perform transposition it can help evaluating horizontal aggregations. The PIVOT

method internally needs to determine how many columns are needed to store the transposed table and it can be combined with the GROUP BY clause[1].


Optimization Of Horizontal Aggregation
Home Ward OT Reception Pathology Pharmacy Blood Bank Doctor Billing Supplier Aggregation

[illegible]

○ SPJ ● PIVOT ○ CASE WHEN

□ Classification

Patient Id	Patient Name	Admission Date	Doctor Name	January	February	March	April	May	June	July	August	Sept
92	[illegible]	[illegible] 12:00:00 AM		[illegible]								
93	[illegible]	[illegible] 12:00:00 AM									[illegible]	
[illegible]	[illegible]	[illegible] 12:00:00 AM							[illegible]			
[illegible]	[illegible]	[illegible] 12:00:00 AM									[illegible]	

Number of Transaction [illegible]

Fig 3: PIVOT Method

Fig 1,2,3, shows result of existing systems for 133k. In this PIVOT is better than CASE and SPJ methods

IV. Proposed System

Conceptual flow of this proposed system is as follows:

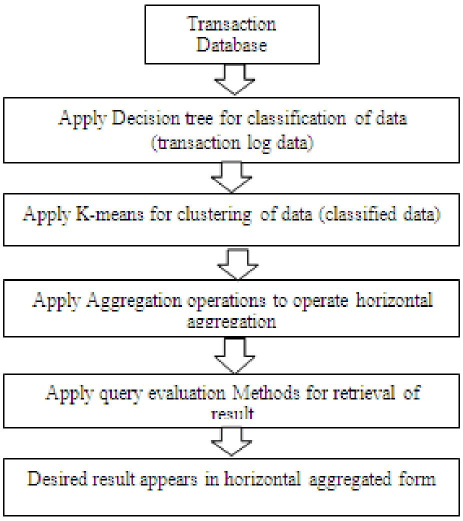

Transaction Database
Apply Decision tree for classification of data (transaction log data)
Apply K-means for clustering of data (classified data)
Apply Aggregation operations to operate horizontal aggregation
Apply query evaluation Methods for retrieval of result
Desired result appears in horizontal aggregated form

Fig 4: Proposed System

Proposed system steps are as follows:

1. In the proposed system, the patient's data come from the transaction database (OLTP) are classified according to specific modules depending upon wherever patients did the transaction, namely Blood bank, OT, Pathology and pharmacy by using C4.5 algorithm.
2. This data can be retrieved by using OLAP operation or query evaluation methods.
3. C4.5 algorithm is chosen because of its higher frequency usage, specificity & high accuracy compared to other algorithms because of its simplicity, robustness and effectiveness [9]
4. Rathee, mathur's paper [9] they did comparative analysis between five decision tree algorithm namely ID3, C4.5, CART, SLIQ and SPRINT. They concluded that C4.5 algorithm is the best algorithm among all the five because it provides better accuracy and efficiency than the other algorithms.
5. K-Means clustering is used to cluster the classified data for efficient and faster retrieval of data. By using this algorithm admin module, retrieve all patients' data according to specific surgery type, test, blood group and a list of patients who purchase the same medicines.
6. One query evaluation method is applied out of SPJ, CASE or PIVOT with some aggregate functions, namely SUM, MIN, MAX and COUNT For retrieval of data in a horizontal aggregated format [6].

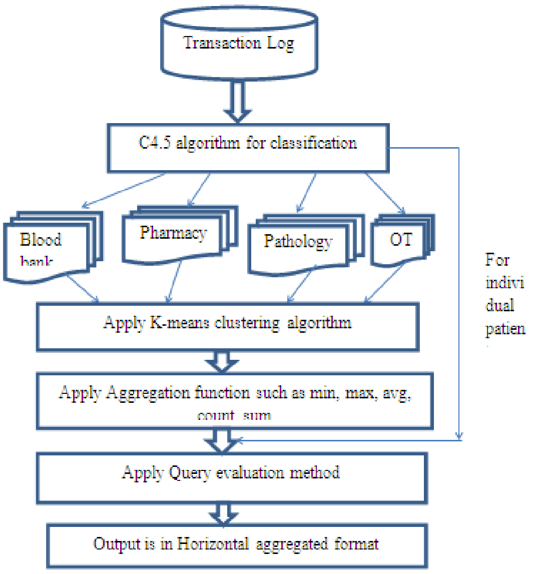

Transaction Log
C4.5 algorithm for classification
Blood bank
Pharmacy
Pathology
OT
For indivi dual patien
Apply K-means clustering algorithm
Apply Aggregation function such as min, max, avg, count, sum
Apply Query evaluation method
Output is in Horizontal aggregated format

Fig 5: Block diagram of proposed system

V. C4.5 Classification Algorithm

C4.5 algorithm is a supervised classification algorithm which is an enhanced version of ID3. It uses Gain Ratio as splitting measures. C4.5 handles both discrete and continuous attributes. For handling continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it & the data is sorted at every node of the tree in order to determine the best splitting attribute. The main advantages of C4.5 is it handles training data with missing attribute values[6].

VI. K-Means Clustering Algorithm

K-means clustering is a simple unsupervised technique to group items into k clusters.

The K-Means clustering algorithm is as follows:
Arbitrarily choose k objects from D as the initial cluster centers
(Re) assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2$$

Here n is data point

$||x_i^{(j)} - c_j||^2$ - distance between data point Xi and cluster center Cj

K – Cluster

Update the cluster means, i.e., calculate the mean value of the objects for each cluster;

Ki ={ti1, ti2, ti3……… tim}

Clusters mean (mi) = $\{\frac{ti1, ti2, ti3.........tim}{m}\}$

Here {ti1, ti2, ti3…. tim} are elements in cluster
And m is number of element in the cluster

Until no change;

Each object is placed in its closest cluster, and the cluster centers are then adjusted based on the data placement. This repeats until the positions stabilize. The results come in two forms: Assignment of entities to clusters, and the cluster centers themselves. The k-means algorithm also requires an initial assignment (approximation) for the values/positions of the k means. This is an important issue, as the choice of initial points determines the final solution [12].

VII. Results

The query evaluation techniques and classification algorithm are implemented on Intel(R) Core(TM) i5-3337U CPU @1.80 GHz,6 GB RAM. Microsoft visual studio and MS SQL Server 2008 is used as a platform. We have applied these techniques on 100000 records. The result of the existing system is as follows:

Table 4:Comparision of query execution time of existing and proposed system for PIVOT method

data size	PIVOT (time in millisecond) old system	PIVOT (time in millisecond) proposed system
1k	200.1311	31.0252
5k	213.1432	65.0399
10k	230.1517	112.0752
25k	266.1757	143.096
50k	333.2136	159.1078
75k	386.2575	179.116
100k	458.3056	194.1311

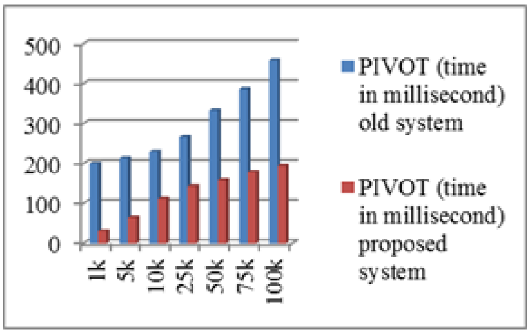


Fig 6: Comparison of query execution time of existing and Proposed system for PIVOT method

Table 5::Comparison of query execution time of existing and proposed system for CASE method

data size	CASE (time in millisecond) old system	CASE (time in millisecond) proposed system
1k	134.1204	35.024
5k	276.2683	64.0438
10k	281.263	117.0747
25k	345.2658	141.0918
50k	481.3508	161.1074
75k	667.4743	200.1311
100k	816.5749	210.1363

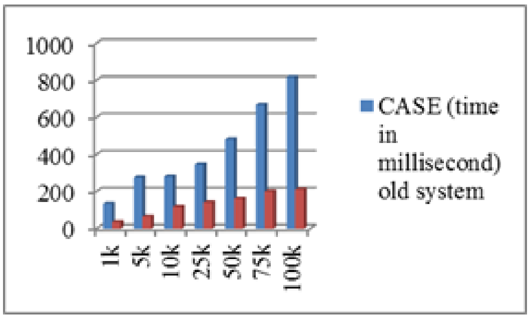


Fig 7: Comparison of query execution time of existing and proposed system for CASE method

From fig 6and fig 7 it is observed that the performance of PIVOT method is better than the performance of the CASE method in case of existing methods and proposed method. It is also observed that there is approximately 50-60% improvement as compared to existing methods.

Table 6: Comparison of query execution time of existing and proposed system for SPJ method

data size	SPJ (time in millisecond) old system	SPJ (time in millisecond) proposed system
1k	184.1204	30.0183
5k	326.2683	61.0401
10k	331.263	145.0938
25k	395.2658	152.1009
50k	531.3508	166.1084
75k	717.4743	218.1472
100k	866.5749	249.1633

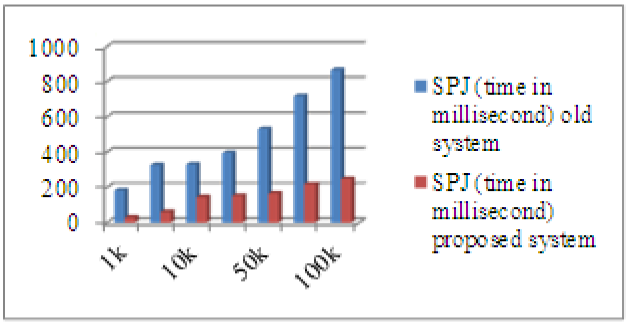


Fig 8: Comparison of query execution time of existing and proposed system for SPJ method

From fig 6, fig 7 and fig 8, it is observed for less number of transaction records SPJ method takes less query execution time as compared to PIVOT and CASE. However, for larger transaction records PIVOT method takes less query execution time as compared to SPJ.

VIII. Conclusion

In this paper, optimization of horizontal aggregation is proposed by using C4.5classification algorithm and K-Means clustering algorithm as standard (Vertical) Aggregation is not suitable for Horizontal Aggregation. These algorithms are used with an aggregation function such as count, sum, min, max, etc. and query evaluation methods namely SPJ, PIVOT and CASE. The proposed horizontal aggregations perform faster as they are made pre-compiled objects. This will save time as the horizontal aggregations are pre-compiled objects. They are executed faster when compared to normal SQL queries. From the analysis of proposed system, it is concluded that for less number of transaction records SPJ method takes less query execution time as compared to PIVOT and CASE. However, for larger transaction records PIVOT method takes less query execution time as compared to SPJ. It is also observed that, query execution time required for the proposed system is better than existing methods.

References

[1] C. C. Ordonez, and Zhibo Chen, Horizontal Aggregation in SQL to prepare Data Sets for Data Mining Analysis," IEEE Transactions on Knowledge and Data Engineering (TKDE), April 2012.

[2] C. Cunningham, G. Graefe, and C.A. Galindo-Legaria, PIVOT and UNPIVOT:Optimization and Execution Strategies in an RDBMS, Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04), pp. 998-1009, 2004.

[3] Venkatadri.m, Lokanatha C. Reddy,"A Comparative Study On Decision Tree Classification Algorithms In Data Mining" ISSN: 0974-3596, April '10 – Sept '10, Volume 2 : Issue 2, Page: 24.

[4] R. Rakesh Kumar, A. Bhanu Prasad," K Means Clustering Algorithm for Partitioning Data Sets Evaluated From Horizontal Aggregations, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 5 (Jul. - Aug. 2013), PP 45-48.

[5] Anuja Priyam, Abhijeet, Rahul Gupta, Anju Rathee, and Saurabh Srivastava," Comparative Analysis of Decision Tree Classification Algorithms" International Journal of Current Engineering and Technology, ISSN 2277 – 4106, Vol.3, No.2 (1June 2013)

[6] Priti Phalak,Rekha Sharma," Optimization of Horizontal Aggregation in SQL by using C4.5 Algorithm", International Journal of Computer Applications (0975 – 8887)Volume 93 – No.13, May 2014

[7] Joyce Jackson, Data Mining: A Conceptual Overview. Communications of the Association for Information Systems (Volume 8, 2002) 267-296.

[8] XindongWu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yangb,Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinbergand, "Top 10 algorithms in data mining", Received: 9 July 2007 / Revised: 28 September 2007 / Accepted: 8 October 2007 Published online: 4 December 2007 © Springer-Verlag London Limited 2007.

[9] J. R. Quinlan, "C4.5: Programs For Machine Learning". Morgan Kaufmann Los Altos, 1993.

[10] Matthew N. Anyanwu, Sajjan G. Shiva , "Comparative Analysis of Serial Decision Tree Classification Algorithms", International Journal of Computer Science and Security, (IJCSS) Volume (3) : Issue (3).

[11] Durka.C, Kerana Hanirex.D,"An Efficient Approach for Building Dataset in Data Mining", IJARCSSE, Volume 3, Issue 3, March 2013.

[12] Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", second edition, 2006.