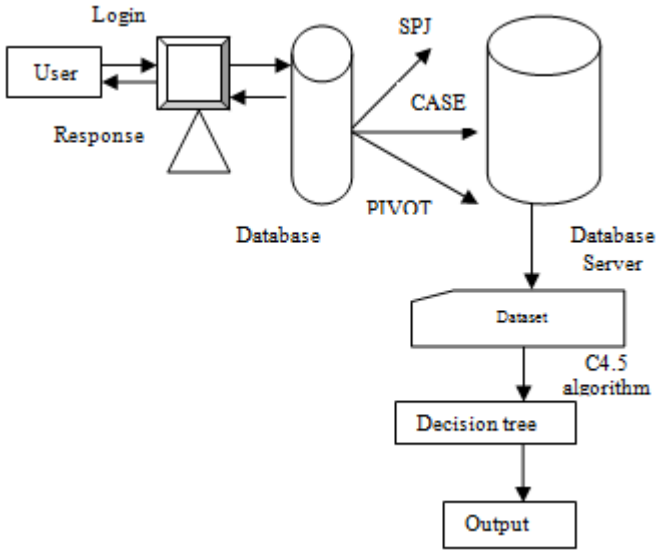


Figure 1: System Architecture for Proposed model

5. Proposed Methodology

In this section proposed horizontal aggregations provide process for creating datasets in data mining analysis. The main goal of this paper to define creates an outline to produce SQL code in tabular layout from data mining tools, these SQL code can be created using SQL complex queries, joining tables and used for prepared datasets in SQL for data mining analysis. A Second goal of this paper to create decision tree by using C4.5 algorithm in WEKA on the prepared datasets. Proposed methodology is explained for four modules given as below.

1) Login Application for User

Proposed work will be able to upload various details regarding separate username and password. Firstly it will create new login application form, will be registered with various user name and password. It access only authorized person. It provides privacy for prepared datasets within DBMS. If new user want to create account in login application form. It will be created then the dialog box shows the message as login successfully. If username and password are mismatched then dialog box shows incorrect username and password. Datasets are created using SPJ practically. Database created in Microsoft SQL Server 2008. The Microsoft SQL Server 2008 Database Engine is a service for accumulate and giving out data in either a relational (tabular) format or as XML documents. Datasets are preparing from the output of all these three method and linked with java language .Net Beans 7.3.1 used for IDE. so by applying SPJ, PIVOT and CASE methods on given database. Dataset are found in .arff file format. On the database, C4.5 algorithm applied and decision tree is generated.

2) Implementation of aggregated SPJ, CASE and PIVOT Methods

Addressing the problem with the prepared datasets this paper proposes three fundamental methods are SPJ, CASE and PIVOT used for horizontal aggregation in SQL to prepare datasets. The methodology adopted for the proposed plan of implementation, transposition and aggregations by following methods:

- SPJ method :

In SPJ Method sub query execute first. After that parent query execute. Select-projection-join (SPJ) method depends on the relational operator. Vertical operations are used in SPJ method. For every column one table is generated in this model. Afterwards, the tables generated are joined in order to obtain final horizontal aggregations Left Outer Join is use in SPJ method, the left outer join is evaluated in between two table's i.e. right part of table and left part of table. The common fields of both the right and left tables are returned and uncommon fields of left column are also returned. Consider dataset is created with 300 attributes. In a horizontal aggregation having four input parameters to create SQL code:-

(i) The known input table F
(ii) The record of GROUP BY columns L1, …… ,Ln
(iii) The column which to be aggregate into (A) and
(iv) The record of transposing columns R1… Rk.

The actual implementation is based on the details given in data sets having outlook, temp., weather and windy
(SELECT outlook, avg (temperature) AS temp, avg (humidity) as hum FROM weather

WHERE play='yes' GROUP BY outlook) F1 left outer join
(SELECT outlook, avg (temperature) AS temp, avg (humidity) as hum FROM weather
WHERE play='No' GROUP BY outlook) F2 on F1.outlook=F2.outlook;

The execution of this query in detail by breaking down the query into sub-query.

The first sub-query as follows,

SELECT outlook, avg (temperature) AS temp, avg (humidity) as hum FROM weather
WHERE play='yes' GROUP BY outlook

In this part of query selection of Outlook, average of temperature, average of humidity is selected from Weather database when play is yes. Group by outlook means all the value for same outlook having play = yes are Aggregated. Consider one example, when outlook is overcast see for play when play is yes. Aggregate all the temperature value for this condition looking at this database.

a) CASE Method

This task is based on the CASE construction provided by SQL. It has a lot of built in Boolean expressions. Out of them one of the expressions is returned. Aggregation or Projection is like to this from relational query point of view. In SQL CASE constructs are available in the SQL CASE programming .It can be done by using many conditions with

conjunctions. In this case horizontal aggregations exhibit two strategies.

1) Firstly, the computations of query can be done directly from the given input database table.
2) Secondly, evaluate vertical aggregation and the results are sent to an arbitrary table. This table is used again in horizontal aggregation generation.

RDBMS has built in PIVOT operator. This is used for the PIVOT operation proposed in this paper. This construct can provide transpositions. It transposes the fewer of rows into additional new column. Therefore, for evaluating horizontal aggregations pivot operator is used to transfer the data from row into column in it. By applying SPJ, CASE and PIVOT methods give equivalent horizontal aggregated result.

b) PIVOT Method

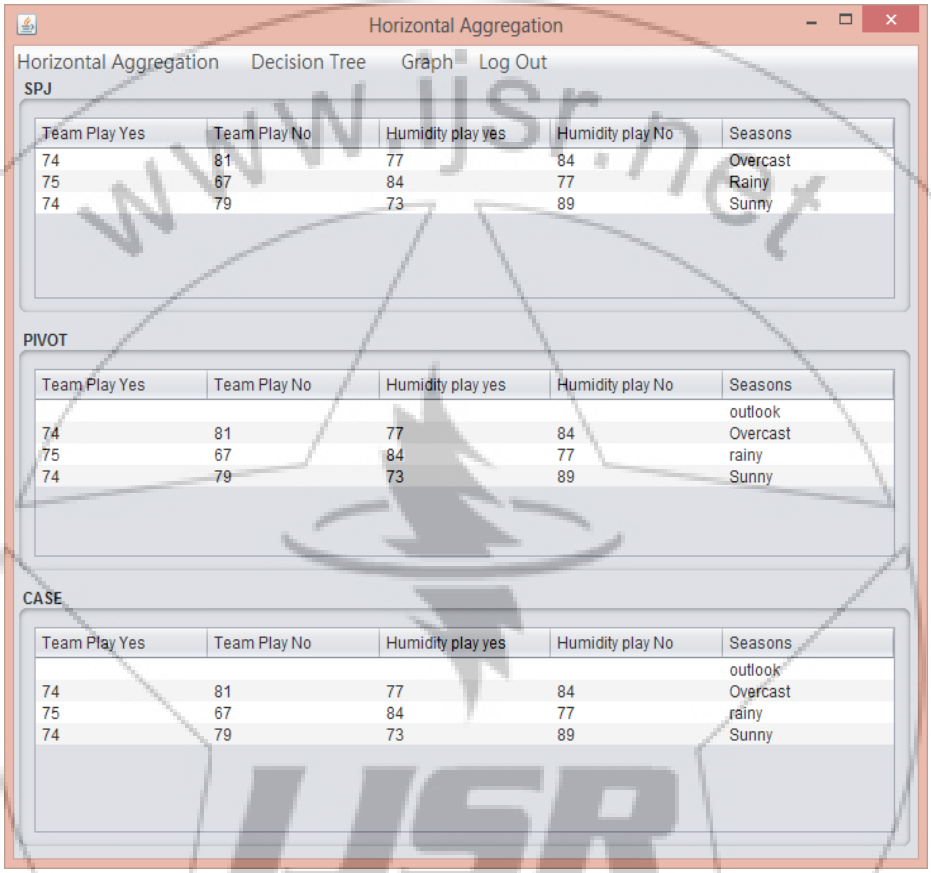


Figure 2: Horizontal Aggregated result

3) Implementation of C4.5 algorithm

Decision tree builds classification models in the form of a tree structure. It splits down in a dataset into smaller and smaller subsets while at that time a related decision tree is incrementally developed. The last outcome is a tree with decision nodes and leaf nodes.

C4.5 algorithm is the latest version of ID3 algorithm. In this module implementation of the C4.5 algorithm will be perform by using the Weka tool. The dataset created by three SPJ, PIVOT and CASE methods, this Prepared dataset is given as an input to C 4.5 algorithm with the help of WEKA tool to generate Decision tree or classification. On that prepared dataset calculate the Entropy and Information gain .Operation are then performed and an appropriate decision rules are produce. Depends on that rule Decision Tree is created. Entropy of each attribute is calculated in every branch. C4.5 algorithm is implemented in WEKA and linked with java file.

4) Decision Tree

Graphical representation of the output of all the methods implemented before in proposed model. Decision tree is generated on the basis of the prepared datasets.

B) Data Flow Work

Data flow chart fig. 2 shows initially create new login form it will be registered with various user name and password. It access only authorized person. If new account is created then the dialog box shows the message as login successfully.SPJ, PIVOT and CASE method applied on the query. Give the same result by three methods. Prepared datasets store in the form of .arff format. On the prepared datasets c4.5 algorithm is applied and finally the decision tree is generated.

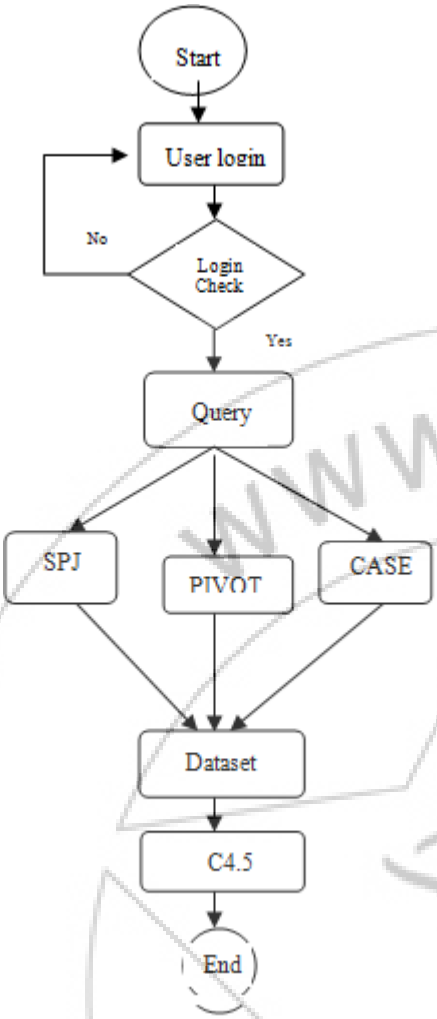


Figure 2: Data Flow Chart

6. Vi. Algorithm Design

A) C4.5 Algorithm

C4.5 algorithm constructs classification model in the form of tree structure and its predecessor, that summaries training data in a decision nodes and leaf nodes. Along with final result is a tree with child nodes leaf nodes makes logical rules to satisfy IF-Else condition. Leaf node represents a categorization or decision. The topmost decision node in a tree which corresponds to the top predictor called root node. Decision trees have capacity to handle both definite and numerical data. C4.5 Algorithm is latest version of ID3 algorithm. C4.5 Algorithm is a well-liked tree based classifier, is used to generate decision tree and from a set of training examples. Nowadays C4.5 is named as J48 classifier in WEKA tool, an open source data mining tool. The function of heuristic used in this classifier is depending on the concept of information entropy.

In general, steps in C4.5 algorithm to build decision tree are:
step1: Choose attribute for root node
Step2: Create branch for each value of that attribute
Step3: Split cases according to branches
Step4: Repeat process for each branch until all cases in the Branches have the same class .Choosing which attributes to be a root is based on highest gain of each attribute.

To count the information gain, we use formula 1, below:

Gain(S,A)=Entropy(S)--

$$\sum_{i=1}^{n} \frac{Si}{S} * Entropy(Si) \ldots\ldots\ldots\ldots(1)$$

With:
{S1,... Si,… Sn} = partitions of S according to values of Attribute A
n = number of attributes A
|Si| = number of cases in the partition Si
|S| = total number of cases in S
While entropy is gotten by formula 2 given as below:

$$\text{Entropy(S)} = \sum_{i=1}^{n} - pi * \log_2 \text{pi} \ldots\ldots\ldots\ldots\ldots\ldots(2)$$

With:
S : Case Set
n : number of cases in the partition S
pi : Proportion of Si to S

- **Tool used**

Weka tool was developed at the University of Waikato in New Zealand. It is most popular. Weka is a set of machine learning algorithms for data mining tasks. Weka include tools for classification (e.g. KNN, C4.5 Decision Tree, Neural Networks), data pre-processing (e.g. Data Filters), clustering, association rules, and visualization etc. Input data in the Weka tool is in the form .arff format. This tool is an open source data in Java. Generally, WEKA tool apply on the given dataset it consists Relation attributes, data sections. he frame looks like below in fig .4

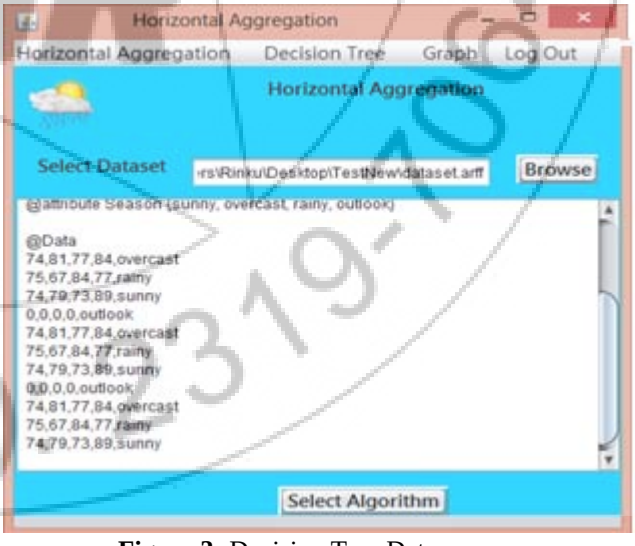


Figure 3: Decision Tree Data

After that select the frame choose the algorithm and WEKA tool to visualize the decision tree.

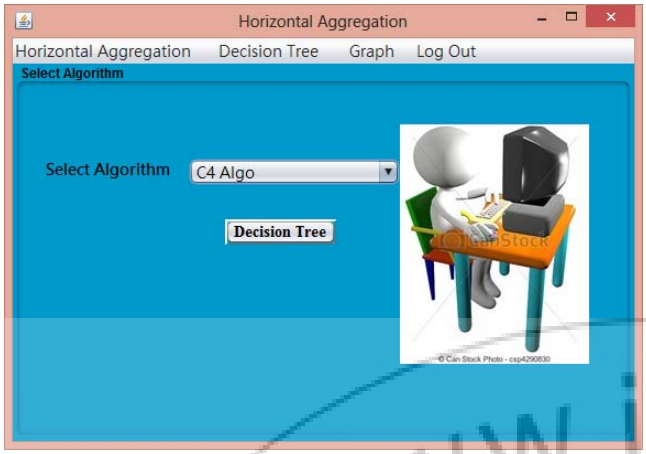
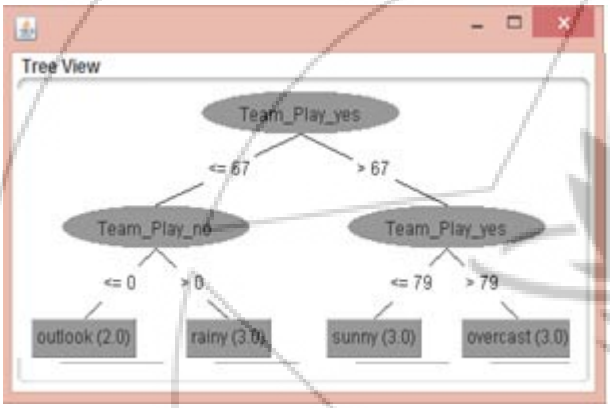


Figure 4: Select Algorithm after Selecting Algorithm

Decision tree is generated by using C4.5 algorithm in WEKA tool as follows:



There is search time comparison in between SPJ, CASE and PIVOT methods. Search time is calculated in millisecond.

Dataset size	100	200	300
SPJ	100	100	100
CASE	10	20	100
PIVOT	20	20	30

Time to get result (in milliseconds)

7. Conclusion and Future Scope

This paper presents methods to support horizontal aggregation through Structured Query languages (SQL) queries in RDBMS. This paper achieves horizontal aggregation through some constructs joining tables, complex queries, SQL queries. The fundamental methods SPJ, CASE and PIVOT are used to estimate horizontal aggregation in SQL to prepare datasets. It gives equivalence results by using SPJ, CASE and PIVOT methods practically and this result is capable for data mining operations. But it is very time consuming task. CASE and PIVOT methods much better than SPJ method. So, proposed plan works on this prepared datasets and the decision tree is generated by using C4.5 algorithm in WEKA. Model built by C4.5 algorithm is require less time than that of ID3 algorithm. Memory used for storing C4.5 Dataset is comparatively less than ID3.In future, use of C4.5 algorithm will helps to decrease time limit required for building model of a particular dataset and also it require less memory to store its Datasets.

8. Acknowledgment

I would like to express my sincere thanks to my Guide **M. S. Gayathri, Associate Professor of Alard College of Engg. And Management, Pune** for her consistence support and valuable suggestions.

References

[1] Rajesh Reddy Muley, Sravani Achanta and Prof. S. V.Achutha Rao, "Query Optimization Approach in SQL to prepare Data Sets for Data Mining Analysis", *International Journal of Computer Trends and Technology (IJCTT)* – volume 4 Issue 8 , pp 1-5,August 2013.

[2] Durka. C and Kerana Hanirex. D, '' An Efficient Approach for building Dataset in Data Mining", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 3,pp 1-5, March 2013.

[3] Carlos Ordonez and Zhibo Chen," Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis", *IEEE transactions on knowledge and data engineering*, vol. 24, NO. 4, pp 1-14, APRIL 2012.

[4] Nisha. S and B. Lakshmipathi, "Optimization of Horizontal Aggregation in SQL by Using K-Means Clustering", *International Journal of Advanced Research in Computer Science and Software Engineering* , Volume 2, Issue 5, ISSN: 2277 128X, PP. 1-6, May 2012.

[5] +Pradeep Kumar and Dr. R. V. Krishna, "Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis" *IOSR Journal of Computer Engineering (IOSRJCE)* ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 5, PP. 36-41, (Nov. - Dec. 2012).

[6] K. Anusha, P. Radhakrishna and P. Sirisha," Horizontal Aggregation using SPJ Method and Equivalence of Methods", *IJCST, Vol. 3, Issue 1, Spl. 5*, pp 1-4, Jan. - March 2012 .

[7] C. Ordonez, "Horizontal Aggregations for Building Tabular Data Sets," *IEEE Trans. Knowledge and Data Eng*," VOL. 24, NO. 4, pp. April 2012.

[8] Mr. Ranjith Kumar K and Mrs. Krishna Veni," Prepare datasets for data mining analysis by using horizontal aggregation in SQL", *Ranjith Kumar K et al, Int.J. Computer Technology & Applications*, Vol 3(6), 1945-1949 IJCTA, pp. 1-5, Nov-Dec 2012.

[9] Sunil Kumar, N. Surya Prakash Raju,"Horizontal Aggregations in SQL to Prepare Data Sets for Data

Mining Analysis". *IJCST*, Vol. 3, Issue 3, July – Sept. 2012.
[10] P. Goel, and B.R. Iyer, "Hyper graph Based Reordering of Outer Join Queries with Complex Predicates," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95)* , pp. 304-315, 1995.
[11] Swetha .Palabindela and Ch. Rajya Lakshmi, "Custom Aggregations for Generating Datasets for Data mining", *International Journal of Advanced Research in Computer and Communication Engineering*,Vol. 2, Issue 9, PP. 1-4, September 2013.
[12] K. Anusha, P. Radhakrishna and P. Sirisha "Horizontal Aggregation using SPJ Method and Equivalence of Methods", *IJCST* ,Vol. 3, Issue 1, Spl. 5,PP 854-857 Jan. - March 2012.
[13] C. Ordonez, "Data Set Preprocessing and Transformation in a Database System," *Intelligent Data Analysis*, vol. 15, no. 4, pp. 613-631, 2011.
[14] C. Cunningham, G. Graefe, and C.A. Galindo-Legaria, "PIVOT and UNPIVOT: Optimization and Execution Strategies in an RDBM S," *Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04)*, pp. 998-1009, 2004.
[15] C. Ordonez, "Vertical and Horizontal Percentage Aggregations," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '04)*, pp. 866-871, 2004.
[16] H. Wang, C. Zaniolo, and C.R. Luo, "ATLAS: A Small But Complete SQL Extension for Data Mining and Data Streams," *Proc. 29th Int'l Conf. Very Large Data Bases (VLDB '03)*, pp. 1113- 1116, 2003.
[17] Han and M. Kamber, "Data Mining: Concepts and Techniques, first ed. Morgan Kaufmann", 2001.
[18] Mark A. Hall, "Correlation-based Feature Selection for Machine Learning", *Department of Computer Science Hamilton, NewZealand*, PP.1 -198, April 1999.
[19] S. Sarawagi, S. Thomas, and R. Agrawal, "Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '98)*, pp. 343-354, 1998.
[20] P. Goel, and B.R. Iyer, "Hyper graph Based Reordering of Outer Join Queries with Complex Predicates," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95)* , pp. 304-315, 1995.

Author Profile

Krupali Rupesh Dhawale, received her B.E. degree from M.I.E.T. Gondia, Nagpur University, and Maharashtra state, India in the year 2008. Currently, Pursuing M.E. in Computer Engineering from Alard College of engineering and management, Marunje, Pune, Maharashtra state, India. Her research interests includes Data mining, Mobile computing. She has published 2 International and 1 national Publications. Her recent publication includes IJCEMR, IJSETR, ETIT national conference. Paper name: - Horizontal Aggregation in SQL to prepare dataset for Data Mining Analysis.

M. S. Gayathri, Associate professor, received the Bachelor's degree in Computer Science & Engg. from P.S.N.A college of Engg & Tech., Madurai Kamaraj University, Tamilnadu state, India and Master degree in computer science and Engg. from Jeusalem College of Engineering, Anna University, Tamilnadu state, India, Her research interests includes network security, biometrics, image processing, data mining, artificial intelligence, compilers and natural language processing.