2011

A Web Based Fuzzy Data Mining Using Combs Inference Method And Decision Predictor

Shajia Akhter Sharmin
Minnesota State University, Mankato

A WEB BASED FUZZY DATA MINING USING COMBS INFERENCE METHOD

AND DECISION PREDICTOR

BY

SHAJIA A SHARMIN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

MASTER OF SCIENCE

IN

INFORMATION TECHNOLOGY

MINNESOTA STATE UNIVERSITY,

MANKATO, MINNESOTA

November, 2011

This thesis paper has been examined and approved.

Examining Committee:

Dr. Cyrus Azarbod

Dr. Hamed Sallam

Dr. Mahbubur R Syed

ACKNOWLEDGEMENT

It is my pleasure to thank all the people who made this thesis possible. This report is submitted as part of the required Thesis work for IT 699 Thesis credit at Minnesota State University, Mankato and has been examined, supervised by Dr. Cyrus Azarbod (Thesis Advisor), Dr. Hamed Sallam (Committee Member) and Dr. Mahbubur R Syed (Committee Member).

My deepest gratitude to Dr. Azarbod, whose continuous support and guidance made this work possible.

I acknowledge my deepest gratitude to committee members Dr. Sallam and Dr. Syed for reviewing my thesis and helping me with the process with their valuable input.

I would also like to thank the Database Research Groups at Minnesota State University, Mankato and from Te American University of Armenia, especially, Anagha Bankar, Robert Azarbod and Saro DerVanasian.

Lastly, I would acknowledge the support of my family, without whose continuous support, this thesis could not have been accomplished.

ABSTRACT

Fuzzy logic has become a very popular method of reasoning a system with approximate input system instead of a precise one. When qualitative variables are used to determine the decisions then we have to create some specific functions where the membership values of the input can be any number between 0 to 1 instead of 1 or 0 which is used in binary logic. When number of input attribute increases it the combinatorial rules increases exponentially, and diminishes performance of the system. The problem is generally known as "combinatorial rule explosion". The Information Technology Department of Minnesota State University, Mankato has been developing a system to analyze historical data and mining. The research paper presents a methodology to reduce the number of rules used in the application and creating a data prediction system using partial incomplete dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Thesis Introduction

This research mainly focuses on reducing number of rules used in the application avoiding the Combinatorial Explosion problem discussed in Comb's Rules of inference method. Depending on the fuzzy raw input, membership values, rule sets the application has the ability to generate a decision. After running different combination of membership sets and rule sets, this system finally presents the result in tabular and graphical format. The web application has been developed in ASP.net Oracle PL/SQL is used for database design.

1.2 Thesis Statement

The propose a methodology to improve the fuzzy decision making system applying Comb's rule reduction method to analyze any numerical dataset, multiple membership sets and rule sets, in addition predicting future datasets from a given partial data.

1.3 Thesis Outline

Chapter 2 presents the background information for all the background information of fuzzy logic, relational databases and the data warehousing, data mining, Materialized view, procedures, and triggers. Chapter 3 and 4 provide the basis for the thesis by describing research projects done by database research group. Chapter 5 explains thesis components and its architecture of Web-FDM. At the end of this chapter will guide the user to understand the working of Web-FDM. Chapter 6 explains the Combs Method of Rapid Inference. Chapter 7 explains the working procedure of new version of Web-FDM 2.0. Chapter 8 describes the Knee Replacement Therapy database and data prediction theory based on partial dataset. Chapter 9 explains how data prediction is applied in Web-FDM 2.0. Chapter 10 shows the examples to prove the successful working of the project. Chapter 11 concludes the research and Chapter 12 presents suggestions for further research.

CHAPTER 2

BACKGROUND

2.1 Fuzzy Logic

The concept of fuzzy logic was first introduced by Zadeh in 1973. He was a professor at the University of California at Berkley. He explained the logic as a way of processing linguistic input in crisp output. The concept became very popular and used in various systems, like thermal control system, gaming system, etc. In recent years the concepts are being used in fuzzy cars, railways. [Naranjo, Gonzalez, Garcia, Pedro, & Sotelo, 2006].

Another example of using fuzzy logic in detecting railway wheel defect [Skarlatos, Karakasis & Trochidis, 2004]. This method mainly takes vibration measurements for a healthy rail wheel at different speeds, and then the vibration measurement of a prior known defective rail wheel is taken. These vibrations are analyzed statistically with confidence intervals for healthy and defective wheels for train speed and frequency of analysis. The fuzzy-logic model created for this system stores the obtained data in the database and performs the decision making on damage and provides preventive maintenance results. [Skarlatos, Karakasis & Trochidis, 2004]

Fuzzy logic is a very important technique for incorporating linguistic input variables with numerical data to interpret user's choice in a qualitative way. In recent years, this logic system is being used with many database systems and can be used to retrieve the documents from such complex databases through queries specified in terms of user's criteria.

2.2 Relational Database

A relational database is a collection of data items which is organized by means of related fields. The Data items are a set of tables from which data can be assembled in different ways making queries without the need of reorganizing the tables. In the relational model tables can have data of different types and can be linked with each other. The data type varies according to the database system. The commonly used databases in recent year are Oracle, MYSQL, SQL Server. The most common data fields in this DBMS are numeric, character and date. Data fields might contain NULL values depending on the users' choice which actually means "not defined" or "nothing at all". There can be different kind of keys or identifiers in a table. The primary key or the unique identifier (PK) of a table helps to uniquely distinguish each rows of a table. The foreign key (FK) related fields in one table with another. FK can be a regular attribute in one table but the primary key in another table. Relational database can also have index which is used to improve performance of database query. The difference between index and keys are, indexes are part of the physical structure of a database, whereas keys are parts of logical structure (Connolly & Begg, 2005). Tables in a relational database can have mainly three types of relationships, one-to-one, one-to-many and many-to-many.

Database normalization is a technique used to ensure data integrity in a relational database. Accuracy, validity and consistency of data is what data integrity means. Referential integrity ensures the relationships among tables consistent. The concept confirms that one cannot add a record to the table containing the FK if there is no associated record in the linked table (Connolly & Begg, 2005). Same way, if a record is deleted from the parent table, that record must be deleted from the child tables.

2.3 Materialized view

Relational database can have two types of views. View is a table which does not have any physical existence. It is located virtually in the database and dynamically displays the records created by a query operating on multiple tables. If the base table gets updated, the views related to it are also updated. On the other hand, materialized view is a concrete table created by caching the query result and can be updated from the base table from time to time. The main purpose of materialized view is to improve performance of query response time. The data is pre-computed, so it allows the retrieval process of information very accurate. Views are used to reduce the load of the network, in creating distributed database, for sub setting data and enabling disconnected computing. Materialized views are updated through batch process from a single base site or master materialized view site. Deployment templates help to create a materialized view environment locally. It allows the data replication based on row and column level subset process. Sub setting enables the replication of information that relates to only a particular site. A dedicated network connection is not required for a Materialized view. A schedule job or manual refresh can refresh the data stored in a materialized view.

Aggregate views, single-table aggregate views and join views are the commonly created materialized view for data warehousing. (Connolly & Begg, 2005). When user runs the same query multiple times, use of materialized views can be very beneficial. Queries re-write mechanism rewrites a query when a similar query is given, to use the materialized view. Database Administrator (DBA) may not know what query is needed to run if he wants to use materialized views. Moreover, in data warehouses a new query has to be executed. At this time the query rewrite mechanism comes into play and uses the query even if only a small part of the query can be satisfied using the view. Rewriting query takes place when a query matches exactly with the materialized view. It might happen that the column with the query references to might not be in the actual materialized view (Summary Join back). When the query request aggregates at a higher level, summary rollup and aggregation to all occurs. (Connolly & Begg, 2005).

2.4 Active database

Active database system improves the functionality of a traditional database, where a pattern of data in database invokes a rule (action). The database executes an event monitoring scheme to detect when a specific data is inserted or updated. Then it executes the action automatically in response and results into certain events depending on meeting some conditions. The thesis project implements functions, triggers, and procedures to calculate fuzzy data.

The rules used for active database are mainly in the form of Even Condition Action (ECA) rules form, with the capability to process rules and provide an efficient mechanism for database system applications. The applications include authorization, knowledge-based system, views, integrity constraints and workflow management. The rules are triggered by the event in ECA model, which includes different types of database update operations (insert, update, delete). At the time of an event occurrence, the ECA model evaluates an optional condition. If no condition is specified, the action is executed immediately after the occurrence of the event. Otherwise the condition is specified first and if it meets the criteria, then the action gets executed. The active database is mainly written in SQL Statements which can be an external program or a mere database transaction. (Widom, & Seri, 1996).

CHAPTER 3

FUZZY DATABASE RESEARCHES

The combination of database and fuzzy technology creates fuzzy database [ref-Omron]. Fuzzy techniques are used in many aspects of relational database, example, knowledge discovery from databases [Maddouri, Elloumi, & Jaoua, 1998], modeling uncertainty at the conceptual schema level. [Chaudhry et. Al., 1999]

Minnesota State University, Mankato and American University of Armenia are doing an extensive research to conduct the dynamic generation of fuzzy tables. Minnesota State University, Mankato has also been doing is actively involved in the field of analyzing and implementing different kind of fuzzy databases and fuzzy data mining (FDM). The following paragraphs described some research projects and their background.

3.1 FAOES (Fuzzy Active Order Entry System)

In the Fuzzy Active Order Entry System (FAOES) project mainly focuses on employee performance and decision making process. Any corporate company's success highly depends on the performance of its employees. Evaluating the performance of the employees can be a very time consuming task especially when it involves high amount of data. Instead of depending on manual performance evaluation process, FAOES project can give a more precise and better decision based on employees' performance throughout a time period. In this project a PL/SQL program was developed to apply the fuzzy technique combining with active relational database. The database used in this project is Order Entry System (OES) which is developed in Minnesota State University, Mankato. The goal of this project is to improve the decision making process to evaluate by dynamically creating the fuzzy tables and generating more mathematically based decisions for employee performance evaluation.

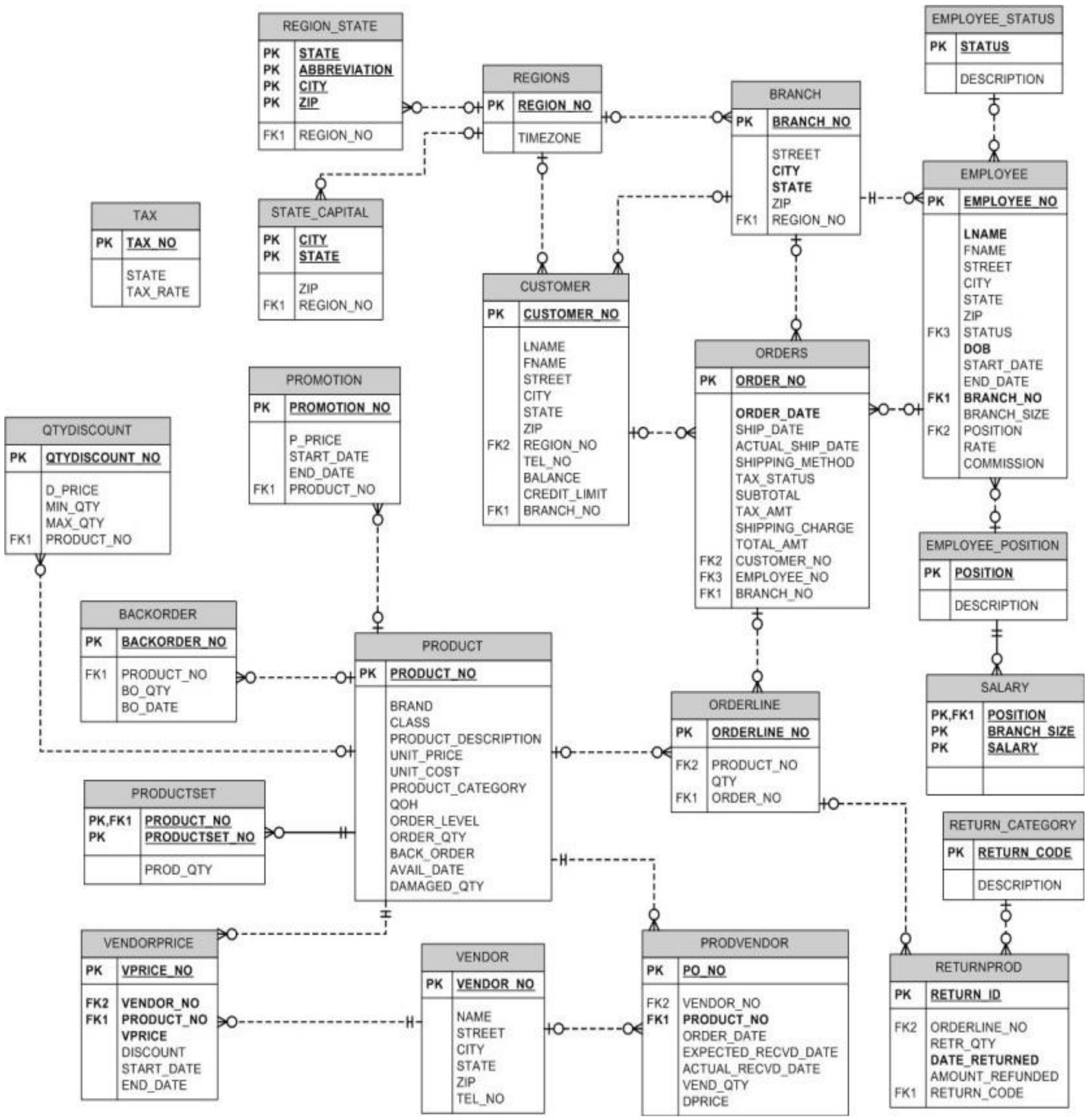


Figure 1: Order Entry System (OES) Database Model

The fuzzy attributes which were considered for the employee performance are Sales, Orders, and Products. To find the number of orders gained by an employee are calculated by joining employee, orders and orderline tables. Once all the relative tables are created the columns can be determined on which fuzzy logic can be applied. These columns are called fuzzy attributes. Two different type of functions are used to calculate the membership values for each fuzzy attributes. These are triangular and trapezoidal function. To evaluate the performance each of the attributes are fallen into a specific fuzzy categories depending on their membership value in that category. In this database the fuzzy categories are – Poor, Below Average, Average, Above Average, and Excellent.

Attribute Name	**Fuzzy Names**	**Fuzzy range**	**Membership Value (a,b,c)**
Orders	Poor	[0,9]	(0,0,9)
	Below Average	[8,20]	(8,15,20)
	Average	[18,30]	(18,25,30)
	Above Average	[28,39]	(28,35,40)
	Excellent	[39,200]	(39,45,1000000)

Table 1: Fuzzy Membership values for Fuzzy Attribute- Order

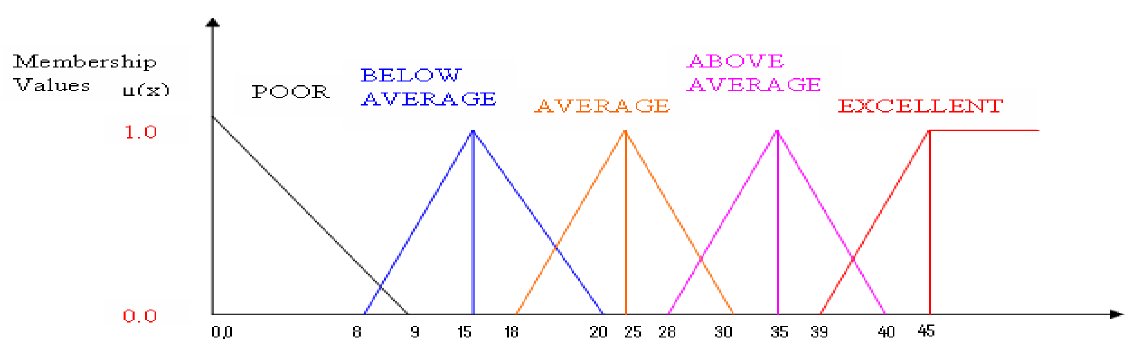


Figure 2: Representation of the five fuzzy categories for Orders Fuzzy Attribute

According to the triangular function the formula which is applied to calculate the membership value is,

If X > α AND α<β

then newValue = (x- α) / (β-X)

Else if X > β AND α<γ then newValue = (γ- X) / (γ- β)

Else if X>= γ then newValue = 0.00

Example: for employee with ID 1069, total sales amount is 7923635. By the mentioned formula,

X= 7923635 β = 7500000

α = 6400000 γ = 8500000

EMPLOYEE_NO	LNAME	FNAME	REPORT_DATE	NUM_SALES	POOR	BELOW_AVG	AVG	ABOVE_AVG	EXCELLENT
1069	Alan	Dusky	1-Sep-05	7923635	0	0	0	0.576365	0
1060	Bixler	Charles	1-Sep-05	2651420	0	0.2962832	0	0	0
1066	Bond	Donna	1-Sep-05	10224635	0	0	0	0	0.9811515

Table 2: Performance table for attribute Sales

Combining fuzzy tables for each attribute, we get table 3-

EMPLOYEE_NO	LNAME	FNAME	REPORT_DATE	ORDERS	SOLD_PRODUT	SALES
1069	Alan	Dusky	1-Sep-05	AVG	AVG	ABOVE_AVG
1060	Bixler	Charles	1-Sep-05	POOR	BELOW_AVG	BELOW_AVG
1066	Bond	Donna	1-Sep-05	POOR	EXCELLENT	EXCELLENT
…	…	…	…	…	…	…

Table 3: Performance table for fuzzy attributes – orders, products, and sales

This is the "Performance" table. This table consists of all the employees showing their performance in each of the fuzzy attributes. The FAOES database also has a rule table consisting of all the rules which are set by the experts.

Rule #	ORDERS	SOLD_PRODUT	SALES	Decision
1	ABOVE_AVG	BELOW_AVG	ABOVE_AVG	Give Raise to Employee
2	ABOVE_AVG	BELOW_AVG	AVG	Give Raise to Employee
3	ABOVE_A	BELOW_A	BELOW_A	Give

	VG	VG	VG	warning to Employee
4	ABOVE_AVG	EXCELLENT	EXCELLENT	Give Raise and Gift to Employee

Table 4: Rules table

Comparing the performance of an employee on each of the attributes and selecting the rule which should fire for that particular employee, the decision table is created.

EMPLOYEE_NO	**LNAME**	**FNAME**	**REPORT_DATE**	**DECISION**
1060	Bixler	Charles	11-SEP-05	Fire
1076	Bullit	Harrison	11-SEP-05	Fire
1061	Harris	Claude	11-SEP-05	Fire

Table 5: Final Decision Table

The proposed Methodology of FAOES has three phases (Azarbod, Sallam & Ali, 2006): Initial Phase, Learning Phase and Mining Phase.

As the second step of research, the modified database is FAOES2 in order to support multiple sets of membership values and rule sets. This enhances the possibility to compare different decision sets depending on the variation of rules and membership value sets.

MEM_SET	PK	STATUS	ATTRIBUTE_NAME	FUZZY_NAME	RANGE_LOW	RANGE_HIGH	MEMBERSHIP_A	MEMBERSHIP_B	MEMBERSHIP_C
101	1	1	ORDERS	POOR	0	9	0	0	9
101	2	1	ORDERS	BELOW_AVG	8	20	8	15	20
101	3	1	ORDERS	AVG	18	30	18	25	30
101	4	1	ORDERS	ABOVE_AVG	28	39	28	35	40
101	5	1	ORDERS	EXCELLENT	39	200	39	45	1000000

Table 6: Membership values for fuzzy set 101

MEM_SET	PK	STATUS	ATTRIBUTE_NAME	FUZZY_NAME	RANGE_LOW	RANGE_HIGH	MEMBERSHIP_A	MEMBERSHIP_B	MEMBERSHIP_C
101	15	1	SALES	EXCELLENT	8.4	1000	8.4	8.5	1000000
102	16	1	ORDERS	POOR	0	12	0	0	12
102	17	1	ORDERS	BELOW_AVG	11	17	11	15	17
102	18	1	ORDERS	AVG	16	28	16	22	28
102	19	1	ORDERS	ABOVE_AVG	28	39	27	31	39
102	20	1	ORDERS	EXCELLENT	38	200	38	45	1000000

Table 7: Membership values for fuzzy set 102

Set 101 and set 102 are two different membership values for triangular function, and set 201, 202 are two sets of membership values for trapezoidal function.

EMPLOYEE_NO	FUZZY_MEM_SET	DEC_SET_RULE	LNAME	FNAME	REPORT_DA	DECISION
1003	201	1	Wooton	Bruce	26-FEB-09	Give Raise to Employee
1003	201	2	Wooton	Bruce	26-FEB-09	Give Raise to Employee
1003	202	1	Wooton	Bruce	26-FEB-09	Give Raise to Employee
1003	202	2	Wooton	Bruce	26-FEB-09	Give Raise to Employee
1003	203	1	Wooton	Bruce	26-FEB-09	Give Raise to Employee
1003	203	2	Wooton	Bruce	26-FEB-09	Give Raise to Employee
1004	101	1	Widdes	Albert	26-FEB-09	Fire Employee
1004	101	2	Widdes	Albert	26-FEB-09	Fire Employee
1004	102	1	Widdes	Albert	26-FEB-09	Give Warning to Employee
1004	102	2	Widdes	Albert	26-FEB-09	Give warning toEmployee

Table 8: Decision table with fuzzy set 101, 102 and rule set 1, 2

The highlighted part of table 8 has been changed depending on the change of fuzzy membership set. Experimenting with different fuzzy membership sets and rule sets, and optimize fuzzy set can be finalized (Bankar, 2008).

The limitation of this database was the membership values are not optimized and decisions made were not realistic. Out of 70 employees, 42 employees were getting fired. Also the positions of the employees were not being considered.

3.2 FDM

In FDM Project (Yanala, 2010), a new set of membership values and rule set with 75 rules were added. The new dataset contained only sales related data and only employees related to sales were considered for performance.

3.2.1 Proposed Methodology

The methodology proposed by FDM helps user to add data initially, then learn the pattern of the obtained data and finally obtaining an optimum fuzzy decision to mine new data.

The FDM Methodology consists three phases: Initial Phase, Learning Phase, and Mining Phase.

Initial Phase:

Step 1: Identify the data set

Step 2: Determine the fuzzifiable attributes

Step 3: Define fuzzy functions (linguistic categories) for each fuzzifiable attribute

Step 4: Identify the membership set values for each fuzzy attribute

Step 5: Identify the rules

Step 6: Identify fuzzy graphs (Trapezoid and Triangle)

Step 7: Create fuzzy tables for each fuzzy attribute

Step 8: Create performance table (to consolidate all fuzzy attributes and fuzzy categories)

Step 9: Create decision table

Step 10: Execute the procedure to generate result for each fuzzy attribute

Step 11: Analyze the result with SQL query tool (SQL)

Learning Phase:

Step 12: Analyze the result with data warehousing designs and OLAP (Pivot)

Step 13: Define new fuzzy graph (to trapezoid), repeat step 6 through step 12

Step 14: Define new set of membership values, repeat step 4 through step 12

Step 15: Define new set of rule, repeat step 5 through step 12

Step 16: If needed, create clusters for each fuzzy category for each fuzzy attribute and repeat step 12

Step 16.1: Identify the desired number of clusters

Step 16.2: Create clusters using weight values from performance fuzzy table

Step 17: Define new number of clusters, go to step 12 and repeat until step 16.2

Step 18: Define new fuzzy categories, repeat step 3 through step 12

Step 19: Create new data set, repeat step 1 through step 12

Step 20: Define new fuzzy attributes (requires new data set), repeat step 1 through step 12

Mining Phase:

Step 21: Mining the final FAOES database version

3.2.2 Components

The following table (Table 9) shows the fuzzy attributes, membership values for FAOES dataset:

Attribute Name	**Fuzzy Categories**	**Fuzzy Range**	**Membership Value (a,b,c)**
Number Orders	Poor	[0,9]	(0,0,9)
	Below Average	[8,20]	(8,15,20)
	Average	[18,30]	(18,25,30)
	Above Average	[28,39]	(28,35,40)
	Excellent	[39,200]	(39,45,1000000)
Number Products	Poor	[0,40]	(0,0,40)
	Below Average	[38,60]	(38,50,60)
	Average	[58,80]	(58,70,80)

Above Average	[78,100]	(78,90,100)
Excellent	[99,1000]	-991,001,000,000
Sales (Mega$) Poor	[0,2.5]	(0,0,2.5)
Below Average	[2.4,4.5]	(2.4,3.5,4.5)
Average	[4.4,6.5]	(4.4,5.5,6.5)
Above Average	[6.4,8.5]	(6.4,7.5,8.5)
Excellent	[8.4,1000]	(8.4,8.5, 1000000)

Table 9: FDM Components

3.2.3. FDM-12 Schemas

Total 12 Schemas were created for FAOES database, each with different membership value and rule sets to obtain a final optimized decision making process.

FAOES2a-Removed hard coded data and moved data to the tables.

FAOES2b- Data cleaning is done (include only sales force).

FAOES2c- expanded membership values for triangle and trapezoid graphs.

FAOES2d- Cleaned data to include return products (Net sales).

FAOES2e- New Rule Sets added and addition of data warehousing and OLAP.

FAOES2f- Considered Gross attributes and tax in fuzzy calculations.

FAOES2g, h, I, j, k, l - Considered yearly sales from year 1999-2003.

FAOES is implemented in Oracle 9i DBMS. The active database is combined of a series of functions, stored procedure, triggers, and materialized views. Figure 3 shows a flowchart for implementing FAOES database. (Haritha, 2010).

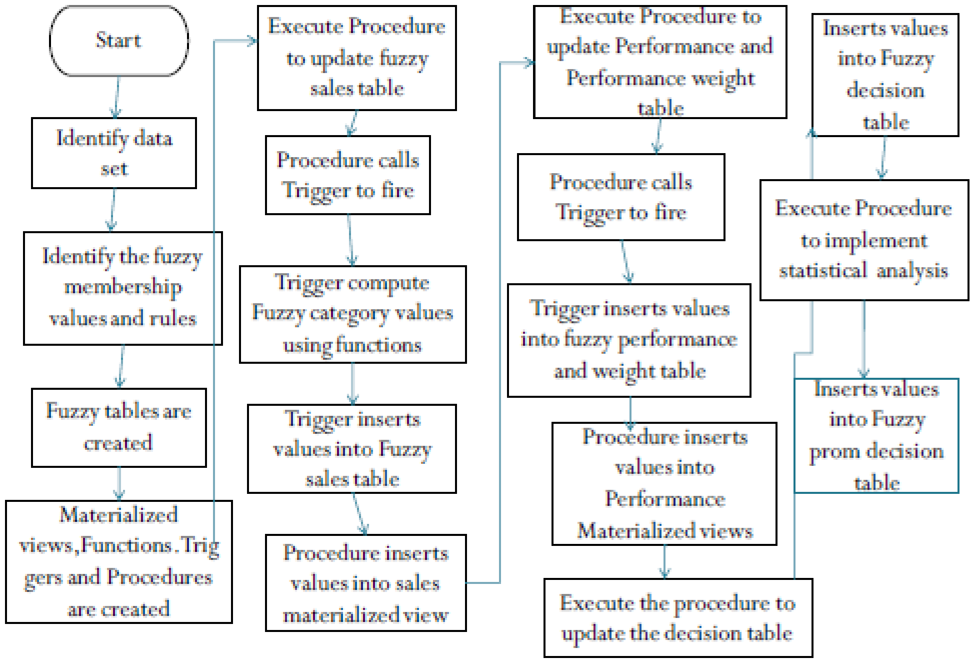


Figure 3: FAOES Flowchart

3.2.4 Sample Query:

A details analysis is done on FAOES database Schemas (Haritha 2010).

Sample: *In the decision table, how many employees are in each category? (FAOES_v2a)*

REPORT_DA	FUZZY_MEM_SET	DECI_SET_RULE	EMP_COUNT	FIRE	WARNING	GIVE_RAISE	GIVE_RAISE_GIFT	NO_DECISION
2-Sep-09	101	1	70	37	1	10	3	19
2-Sep-09	201	1	70	36	1	11	3	19

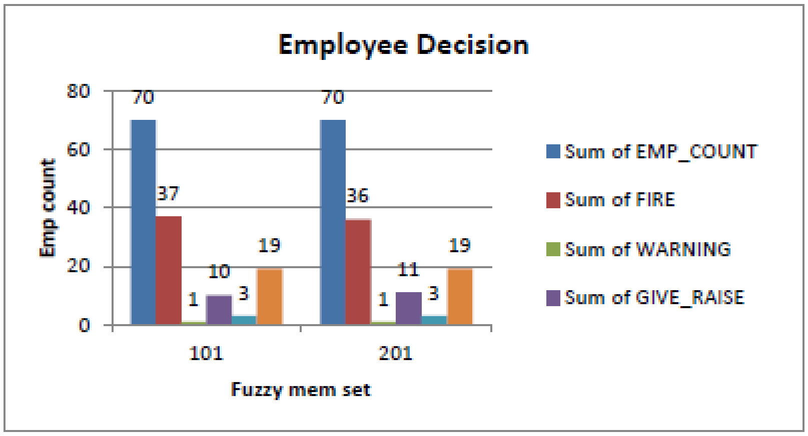


Figure 4: Sample Query Result using iSQL and Pivot chart

Using iSQL and pivot chart the results are analyzed. The figure 4 shows that out of 70 employees 19 are getting "No Decision" as the database was not populated with enough rules. Again, 37 employees are getting fired which cannot be a feasible solution. The decision shows that the database needs to be revised in order to get an optimized solution.

3.2.5. Limitations

- The FDM methodology still consist of some limitations though it resolves some from the FAOES2 project.
- The project only works for OES database.
- The project cannot be dynamically used for any other database
- The solution used for FAOES requires huge amount of programming if is used in any other database.

CHAPTER 4

ARDIF PROJECT

ARDIF project (Automatic Extension of Relational Database to Incorporate Fuzzy Logic) (Deravanesian, 2007) provides a solution to overcome the limitations of FDM project. This is a generic version of program which can run any database and. The system dynamically can create the fuzzy tables for each of the databases. ARDIF was implemented on Oracle 9i using extensive use of PL/SQL language. ARDIF accepts a series of meta-data of any database in the form of a work table, fuzzy attributes, fuzzy categories, membership values for both trapezoidal and triangular function With the help of functions, stored procedures, triggers generates all the decision tables ARDIF calculates the weight of each attributes and fills each fuzzy table. From this table, the program populates the performance table and combining with rules table the program generates the final decision (Deravanesian 2007). The generated decision table is stored for future evaluation and all the temporary tables are deleted.

4.1 Proposed Methodology

The proposed Methodology is similar to the Initial phase of HDM. Figure 4 shows the steps involved in the processing and calculating the final decisions using ARDIF model.

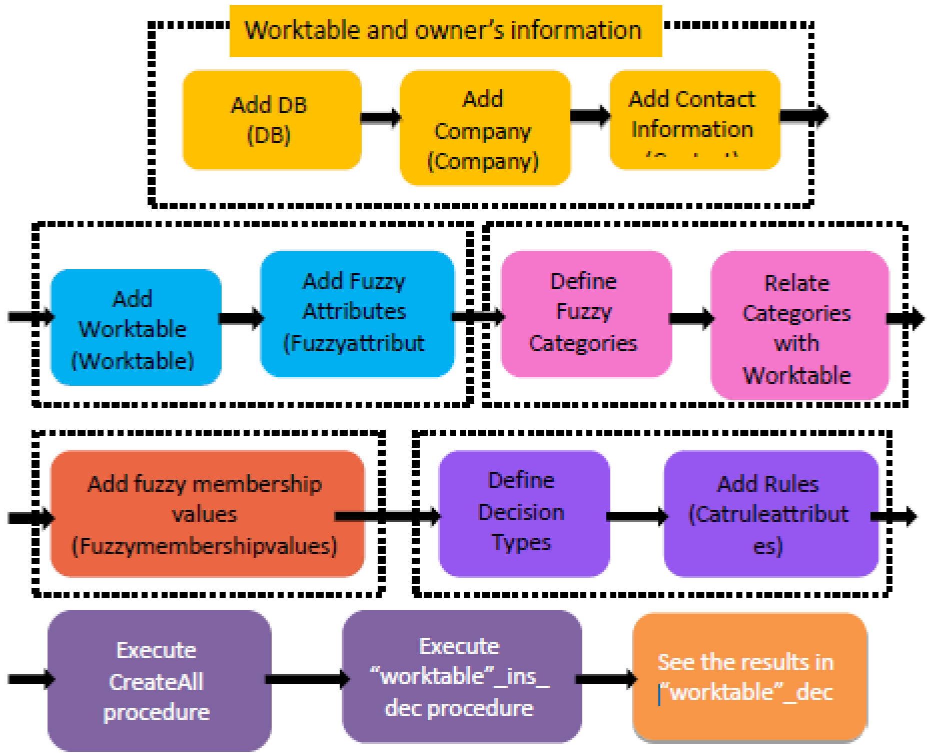


Figure 5: ARDIF: Proposed Methodology

4.2 Components

ARDIF first takes input of dataset table and dataset name with owner information which is inserted in the worktable. This information is used later to identify the dataset for a particular user. Information related to fuzzy attributes, categories and rules are then inserted. All the fuzzy tables are connected and identifiable by worktableID. (Deravanesian, 2007)

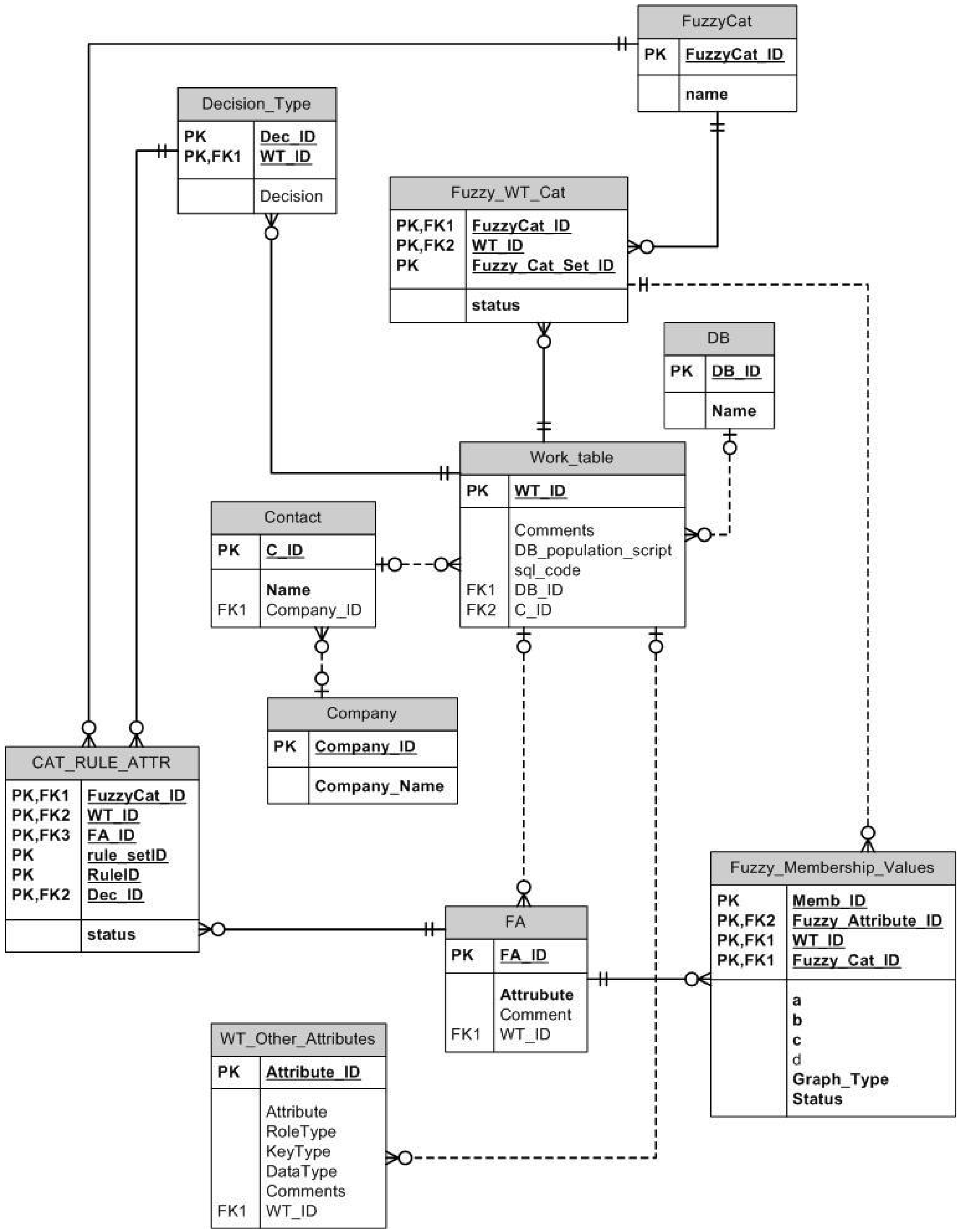


Figure 6: ARDIF Data Model

Figure 7 shows the components of the active database that run behinds to generate all the fuzzy tables.

- Stored procedures
    - CREATEFUZZYTABLES (to query metadata, and create fuzzy tables according to the result)
    - CREATEFUZZYMV (create materialized views identical to fuzzy tables)
    - CREATEFUZZYPERFORMANCE (to create fuzzy performance table by querying the metadata)
    - CREATEFUZZYWEIGHT (to create fuzzy performance weight & fuzzy performance table)
    - CREATETRIGGER (to calculate the fuzzy categories and insert them into fuzzy table) The code of these triggers might be different from database to database.
    - CREATEFUZZYPERFTRIGGER (to populate the fuzzy performance weight table)
    - CREATEDECISIONTABLE (to create the decision table by querying the metadata)
    - CREATEDECISIONRULETABLE
    - GENERAL_UPDATE_FT (inserts the raw data into the fuzzy materialized views to fill the fuzzy tables)
    - P_UPDATE_FUZZY_PERF (inserts the raw data into the fuzzy performance materialized views to fill fuzzy performance and fuzzy performance weight tables)
    - GENERALINSERTDECISION (by analyzing the fuzzy performance table and rules table created by CREATEDECISIONRULETABLE procedure)
    - DROPALL (drops all the temporary tables to save data in the server)
    - CREATEALL (creates all procedures, tables, sub program, & then passes control to the subprogram)
- Functions (to calculate weight values for triangular and trapezoidal functions)
- Metadata (for each database, maintain info such as: contact, database id, work table, fuzzy attribute, fuzzy categories, membership values and decision rule types)
- Automatically generated stored procedures and triggers

Figure 7: ARDIF Program Components

4.3 Implementing FAOES Database using ARDIF

For FAOES the fuzzy attributes are gross_products, gross_orders and gross sales. The defined fuzzy categories are Poor, Below Average, Average, Above Average and Excellent. The categories are connected to the related dataset using datasetID and CategoryID. Membership Values table is used to store the membership values for both triangular and trapezoidal functions.

WT1	The main work table for OES database. it‟s the detail about each employee
WT1_DEC_RUL	Decision rules provided by the manager.
WT1_SALES_TRI WT1_SALES_TRP	Category of total sales calculated for each employee using 2 functions.
WT1_PRODUCTS_TRI WT1_PRODUCTS_TRP	Category of total sold products calculated for each employee.
WT1_ORDERS_TRI WT1_ORDERS_TRP	Category of total orders calculated for each employee.
WT1_WEIGHT WT1_PERF WT1_DEC	Weight, performance and decisions calculate

Table 10: Tables in database

Then the decision types are added and the rules are built depending on these decision types, categories and attributes. A column in the rule table is set as “active” or “inactive” to specify which rule set is currently under consideration. (Deravanesian, 2007)

Decision types for FAOES database are- Fire, Give Warning, Do nothing, Give raise, give gift and raise.

Some rules are-

If gross_products is poor, gross_orders is poor and gross_sales is poor, then Fire.

If gross_products is excellent, gross_orders is above average and gross_sales is excellent, then Give gift and raise.

Then the following procedures are executed to generate the decision

1. Execute createALL(“Work table Name”, worktable_ID)

   For FAOES it would be

   Execute createALL(‘WT1’,1);

   This procedure creates all the fuzzy tables and a procedure which will fill the decision table.

2. `exec "WORK TABLE NAME"_INS_DEC;`

   `exec wt1_ins_dec;`

4.4 Limitations

- ARDIF only supports the initial phase of FDM methodology.
- Data warehousing and OLAP is not implemented in ARDIF.
- As ARDIF removes all the fuzzy tables after decision table is filled up, so there is no way of comparing different strategies.
- Extensive knowledge of database programming is needed to implement ARDIF.
- ARDIF does not have any User interface for better user interaction.

4.5 ARDIF2 - ARDIF Expansion

ARDIF2 – is an enhancement of ARDIF which solves the problem of single set of decision storage in database. The user can compare different results of different membership and rule sets to finalize which decision set is the optimized one. (Bankar 2010)

The modified architecture has the following procedures.

Procedure Description	Procedure Name
Save generated Decisions into the decision Table	(“WorktableName”, WorktableID)
Ability to analyze the dataset for particular combinations of rule set # and membership set#	SetActiveInactive(“WorktableID, FuzzySetID, RuleSetID)
Ability to select the structure of data set used in the mining phase	CreateDatamineTable(“TableName”,WorktableID)
Ability to identify the optimum selection of learned dataset, membership values, and rules	CreateDatamineTable(“TableName”,WorktableID

Table 11: ARDIF2- added procedures

4.6 Limitations of ARDIF2

- User must be a knowledgeable database programmer
- This system does not have any OLAP or visualization support
- There is no user interface
- User have very limited access to add, update and analyze data
- Difficult to add new user.

CHAPTER 5

WEB BASED FUZZY DATA MINING (Web-FDM)

The Web-FDM project is developed to overcome the limitations of ARDIF, ARDIF2 projects. The following issues are solved in Web-FDM (Bankar 2010):

- Support all the steps of FDM methodology: initial Phase, Learning Phase and Mining Phase
- User interface for better user interaction
- Through the learning phase optimal result generation
- User roles allocation
- Web-FDM provides support for visualization of decision results to get a clear understanding of the pattern.

5.1 Architecture of Web-FDM

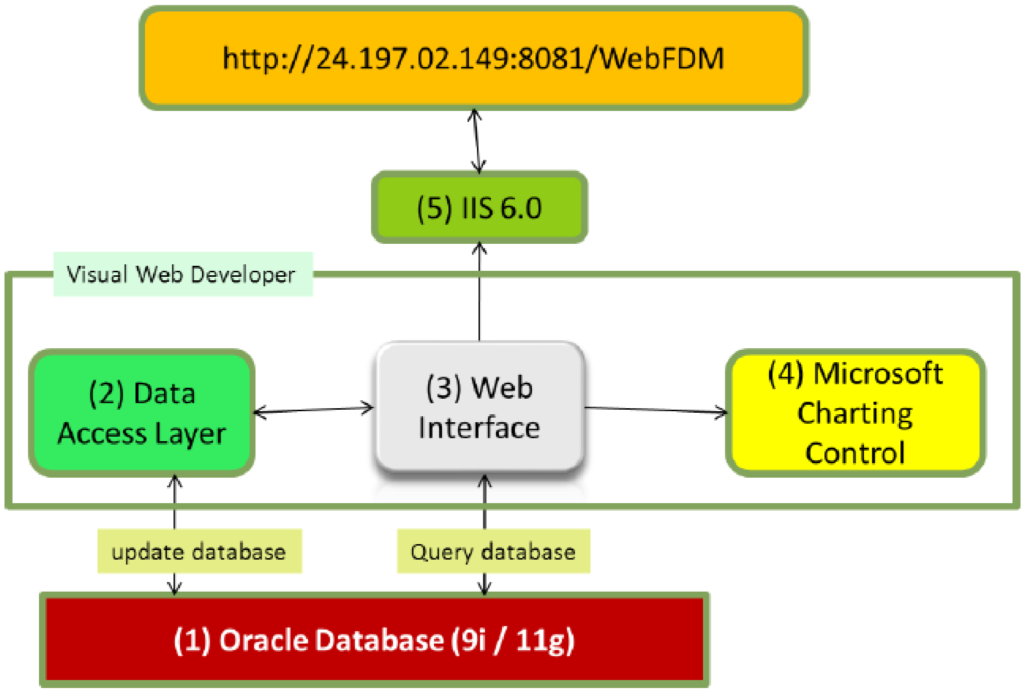


Figure 8: Web-FDM Architecture

- Database: Oracle 9i is used to create ARDIF2 database for Web-FDM.
- The Web interface is created using C# programming language.
- Data Access Layer (DAL) is used as middle-ware to add, update and retrieve data. For some complex queries, DAL is bypassed to connect to the database directly.
- Microsoft charting Control: Used for graphical representation of the data.
- IIS: used to publish the website on the internet.

5.2 Components

- Oracle 9i: The database is hosted in a Oracle 9i server

- FDB3 Database: The database schema name is FDB3 which consists of all the related tables to create the ARDIF Engine, stored procedures, functions, triggers, sequences to support the fuzzy logic.

- ASP.net Charting control: To visually represent the decision of the databases Charting Control DLL file is used.

- Data Warehousing: To store data to analyze fuzzy logic and compare results of different sets.

- TierDeveloper 6.1: The software is used to generate Data Access Layer. This application maps oracle table in different objects and gives direct access to different columns of the tables to have a faster calculation. DAL layer also provides clean code and code reuse flexibility.

- IIS6: IIS is Internet server where the server side of the application is stored. The server separates the C# code written in the application from the HTML and helps to provide a streamline process.

5.3 Data Access Layer (DAL)

DAL makes the access to stored database simple by adding a separate program layer. Using the layered approach to increase the application performance, code reuse and code encapsulation. The main objective of DAL is to provide data to business layer without using any database specific code.

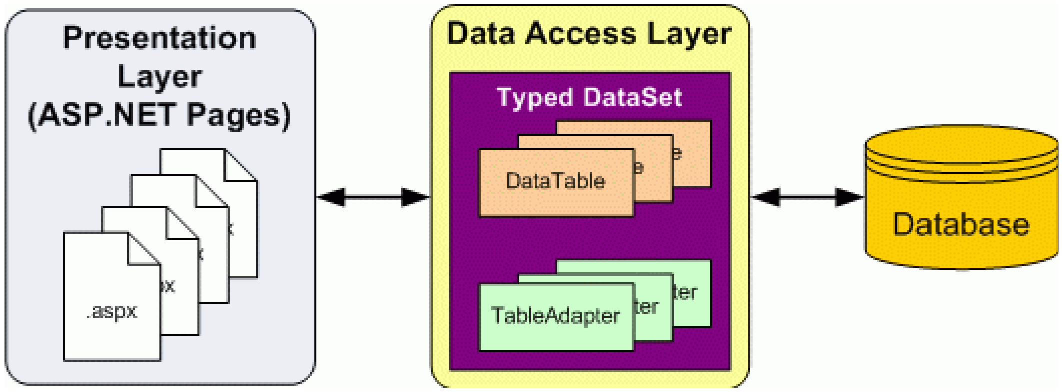


Figure 9: Data Access Layer (DAL) Architecture

TierDeveloper is a very powerful object relational mapping tool which maps the oracle tables to C# objects. Any application can be built on top of the generated code.

Some key benefits of TierDeveloper are:

- It speeds up the development as code is generated instantly.
- Code quality is improved as design pattern is implemented
- Generated code is pre-tested which reduces the testing effort.

5.4 Initial phase of Web-FDM:

This phase consists of five steps

5.4.1 Step 1

User's basic information and worktable is added in the database in this step. (Figure 10)

5.4.1.1 Add Database: User inputs the name of the database they will work on. The name has to be unique.

5.4.1.2 Add Company Name: This name is used for login purpose. This name is used to relate each database with its own company name. The name should be unique.

5.4.1.3 Add contact name: This is the name of the person whom the system would contact relating to the results.

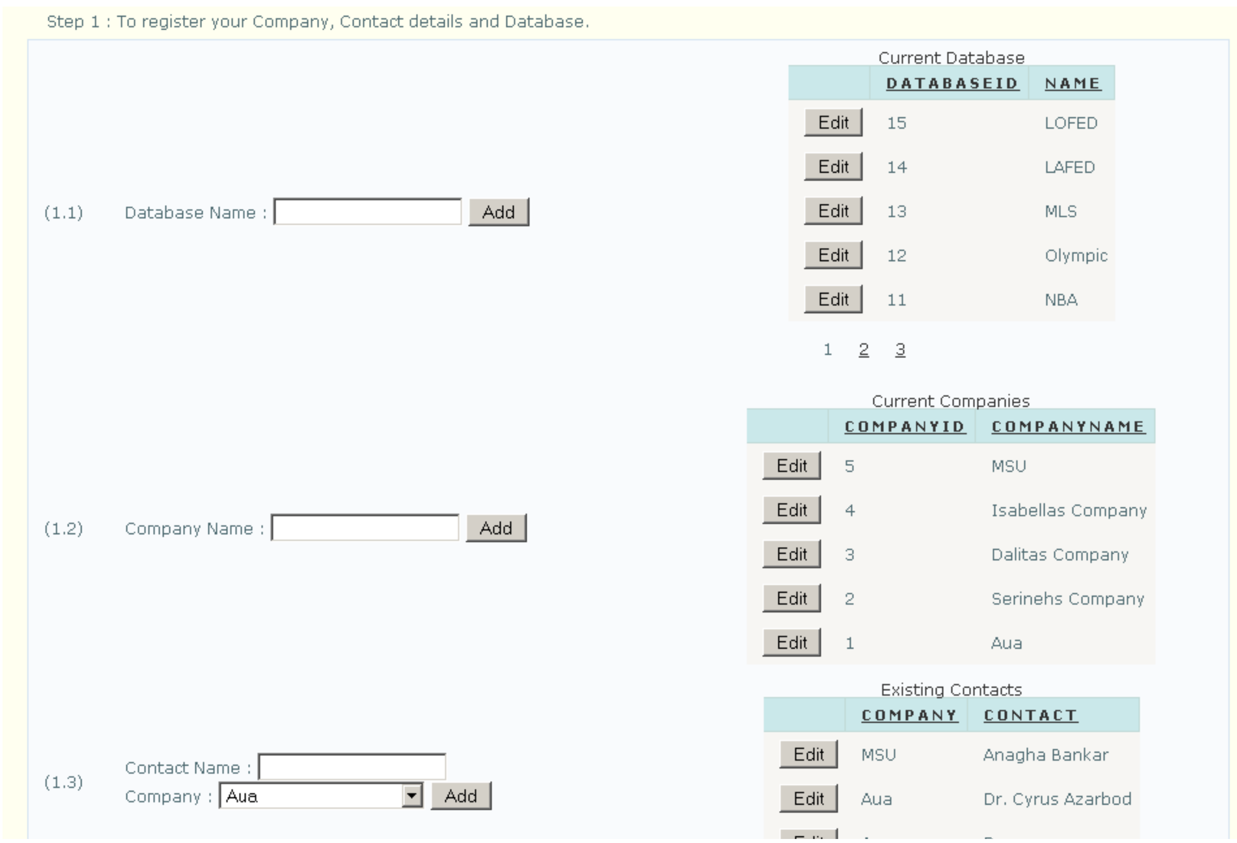


Figure 10: Initial phase- step 1

5.4.2 Step 2

5.4.2.1 Add Dataset Name: In this step user will input the name of the dataset and select the appropriate contact name to create a relationship between the dataset name and contact.

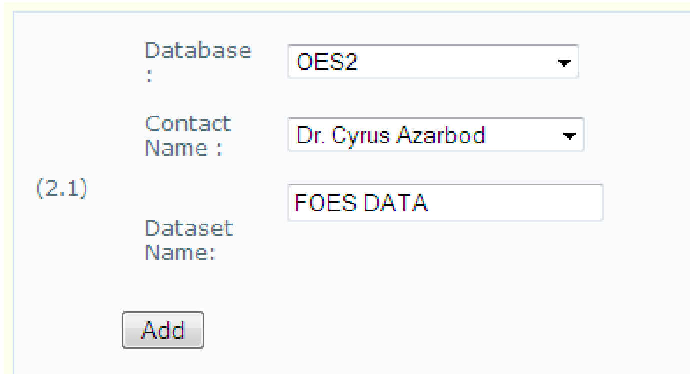


Figure 11: Initial set-up- Step 2

5.4.2.3 Add Dataset: In this step the user will load an oracle script to populate the database tables. Figure 12 shows a screenshot of the User interface. Appendix contains the insert script for FAOES database.

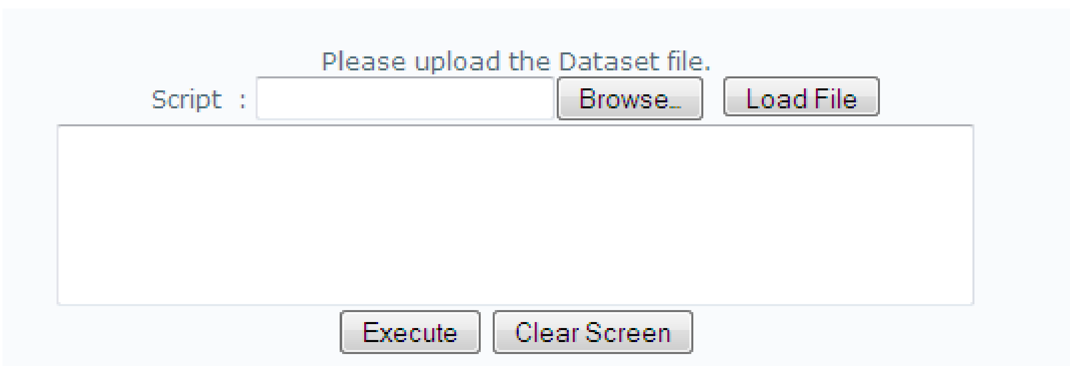


Figure 12: Initial setup- Step 2- Dataset insert

5.4.2.4 Add fuzzy Attributes: The application retrieves the fuzzy attributes from the column names of the insert script. Figure 13 shows the fuzzy attributes ad other attributes retrieved from the FAOES dataset.

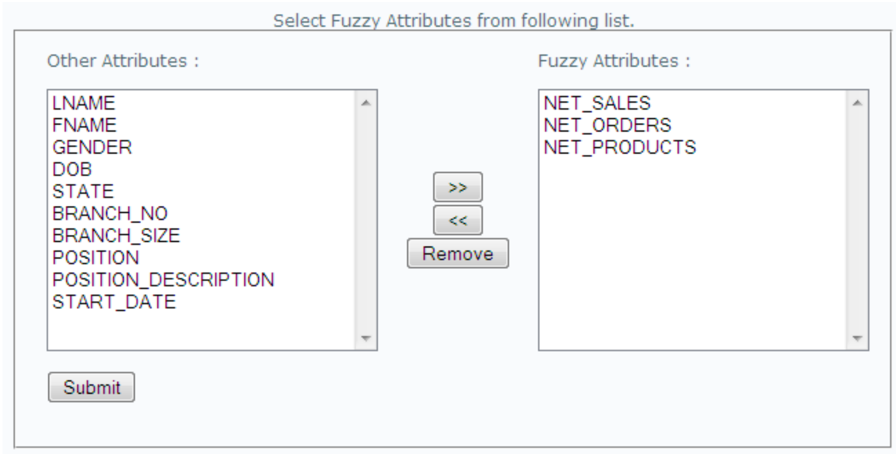


Figure 13: Initial set-up – Step 2: selection of fuzzy attributes

5.4.3 Step 3

5.4.1.1 Add Categories: In this step the user adds fuzzy categories. For FAOES fuzzy categories are Poor, Below Average, Average, Above Average, and Excellent.

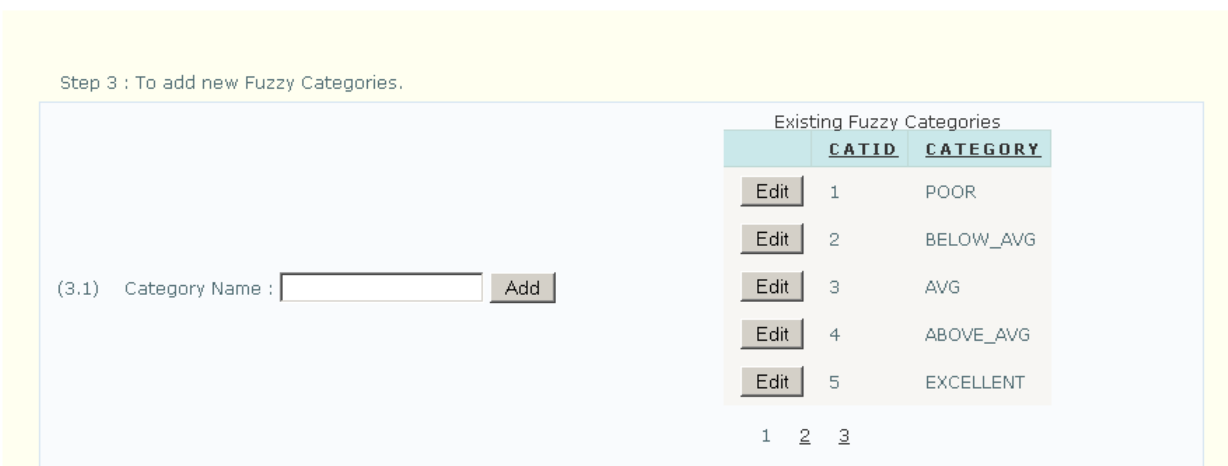


Figure 14: Initial setup- Step 3- Category Name entry

5.4.3.2 Relate Categories to a specific dataset: In this step user makes a relation between the dataset and created categories. Each dataset can have multiple sets of categories where one set is set “active”. At any time user can add additional categories in an existing category set.

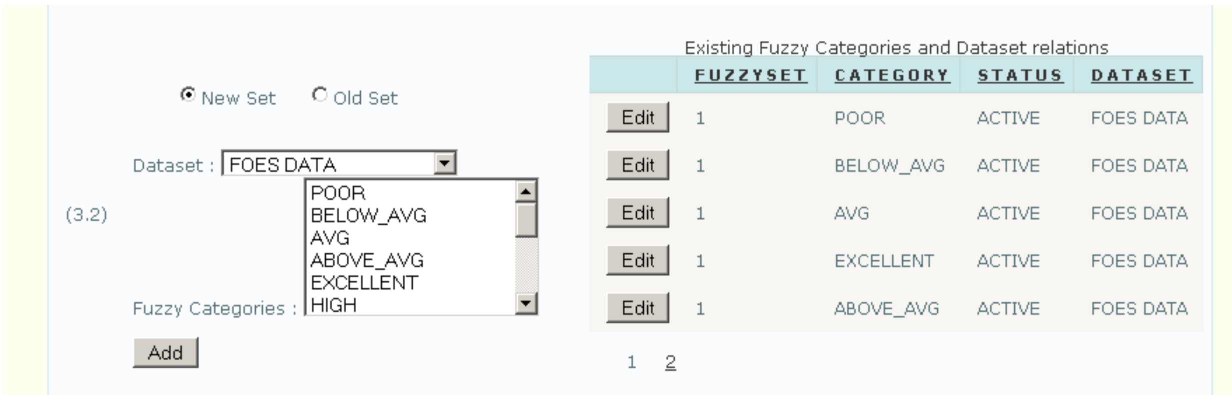


Figure 15: Initial setup- Step 3- Relate categories to a specific dataset

5.4.4 Step 4

5.4.4.1 Selection of Dataset, Fuzzy Attribute and rounding value:

In this step user is presented with a graph containing the membership values for fuzzy attributes for a specific dataset. User can adjust the rounding value using the slider shown in figure 16.

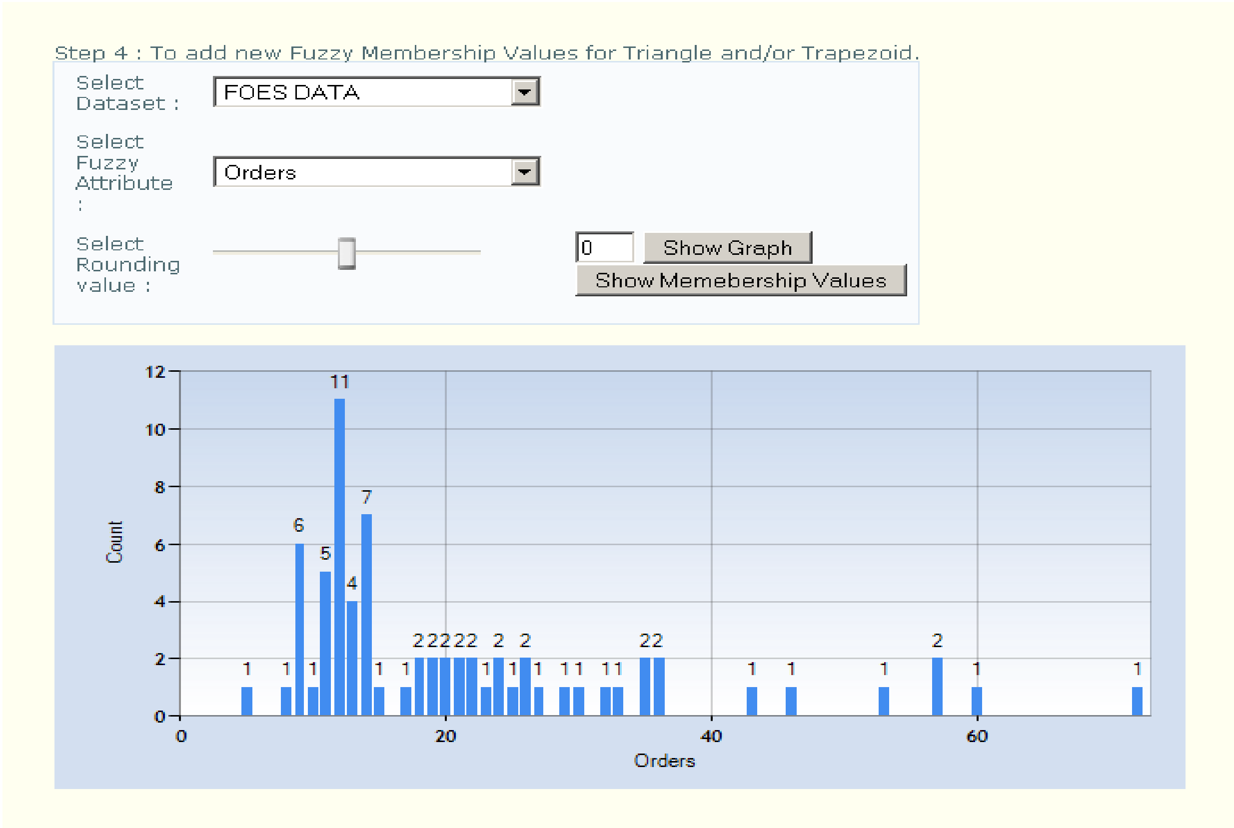
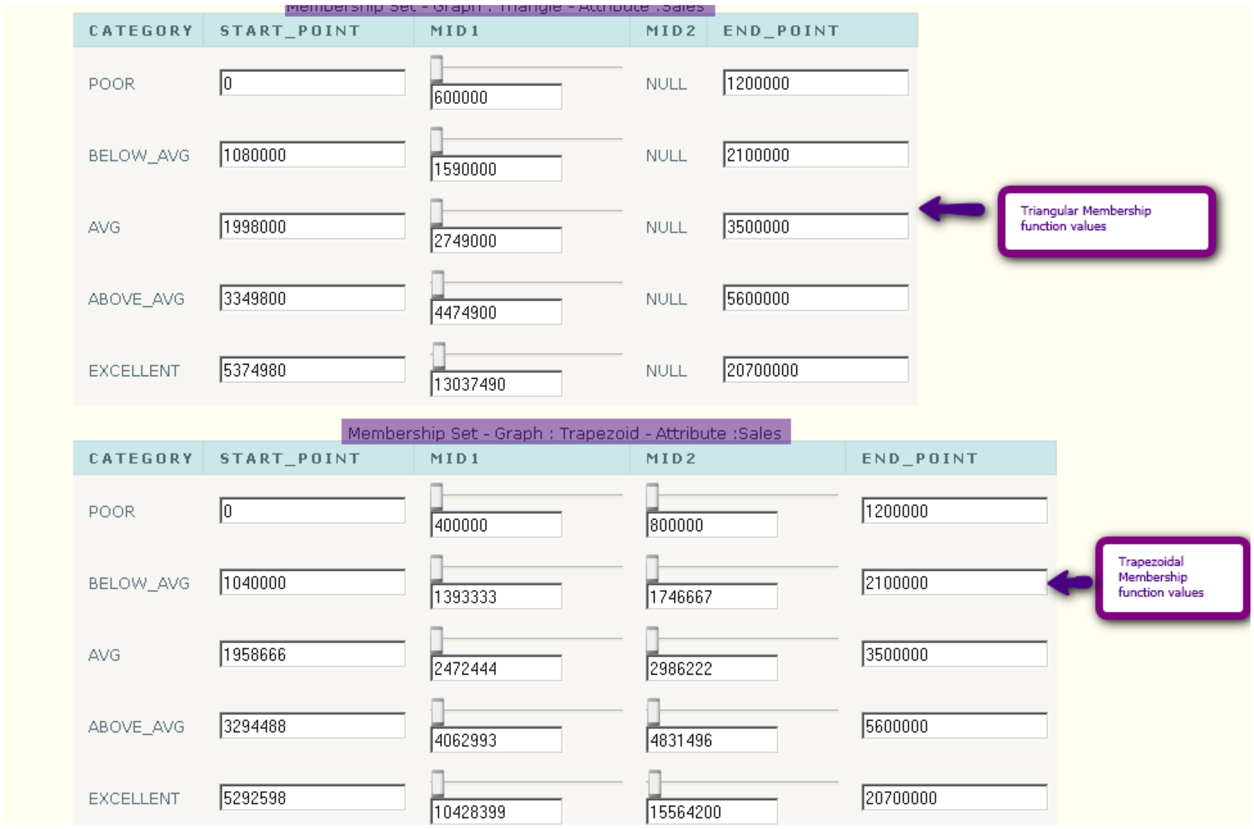


Figure 16: Initial setup- Step 4- Graphical representation of input data

5.4.4.2 Auto generating membership value: Gaussian distribution is used to generate membership values for a fuzzy attributes. The values are generated for both triangle and trapezoidal function. By default, there is a 70% overlap of between two graphs which can be changed by manipulation. At the end of this step user saves the membership values in the database by clicking on the “Submit” button.



Membership Set - Graph : Triangle - Attribute :Sales

CATEGORY	START_POINT	MID1	MID2	END_POINT
POOR	0	600000	NULL	1200000
BELOW_AVG	1080000	1590000	NULL	2100000
AVG	1998000	2749000	NULL	3500000
ABOVE_AVG	3349800	4474900	NULL	5600000
EXCELLENT	5374980	13037490	NULL	20700000

Membership Set - Graph : Trapezoid - Attribute :Sales

CATEGORY	START_POINT	MID1	MID2	END_POINT
POOR	0	400000	800000	1200000
BELOW_AVG	1040000	1393333	1746667	2100000
AVG	1958666	2472444	2986222	3500000
ABOVE_AVG	3294488	4062993	4831496	5600000
EXCELLENT	5292598	10428399	15564200	20700000

Figure 17: Step 4- Membership values generated for fuzzy attribute – Sales

5.4.5 Step 5

5.4.5.1 Add decision: In this step a decision type is added for a specific dataset (Figure 18)

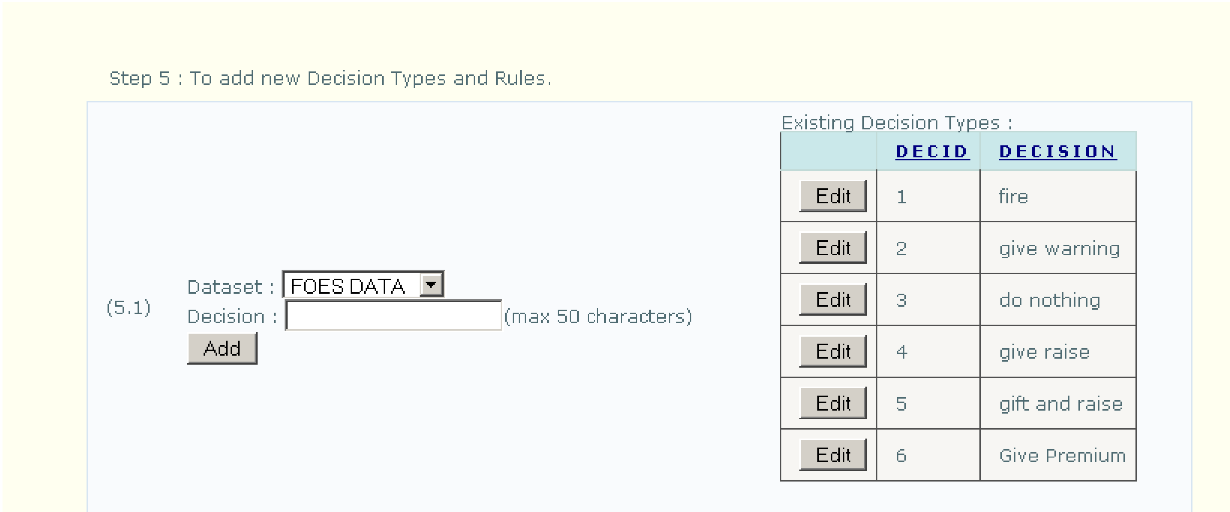


Figure 18: Initial setup- Step 5- Insert Decision Type

5.4.5.2 Create decision set: Decision set is created based on the decision type. The page automatically generates a combination of all the fuzzy attributes, categories and list of decision types so that user can define rules by selecting the right combination.

5.4.6 Pre- Analysis Stage: After completing the initial setup, pre-analysis stage runs the system for a particular dataset. The performance report shows the analysis of fuzzy logic.

5.4.6.1 Execute the System: In this step user proceeds with the data processing for specific dataset, fuzzy set and rule set. Stored procedures, functions and triggers are run to create decision set.

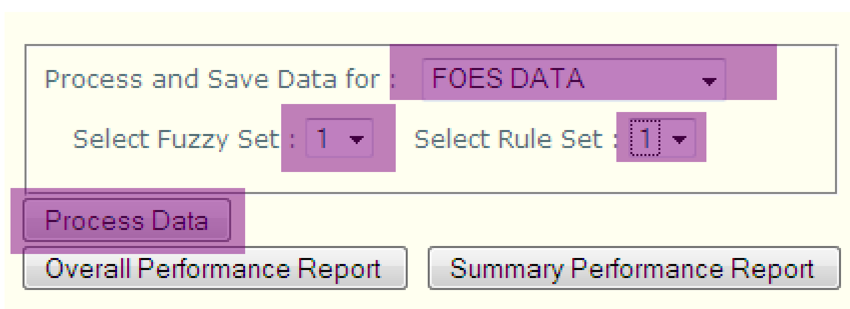


Figure 19: Processing the data set using fuzzy logic

5.5 Learning Phase of Web-FDM

This step includes the analysis and revising of existing data. A customized tool is provided to analyze datasets.

User can select the required dataset, fuzzy set, rule set. There are options for three types of analysis. Figure 20 shows the customized toolbox. The optimal result is recorded in the database by selecting the “freezed” checkbox. This result set is used for mining phase in future.

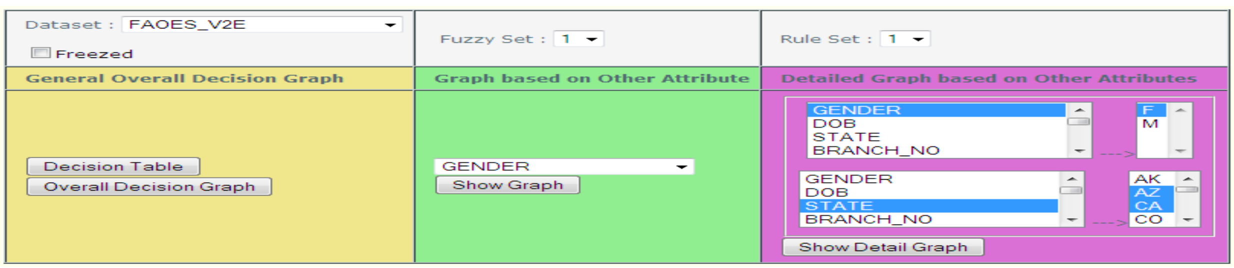


Figure 20: Analysis toolbox

Three types of analysis is available in this step

5.6 Overall Analysis: This type of analysis is shown by graphical report based on the decision table. The data can be viewed percentage-wise or with actual values.

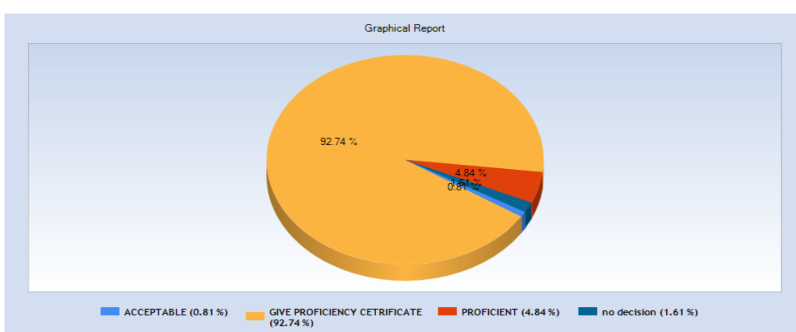


Figure 21: Overall analysis graph for LAFED dataset

5.6.1 Analysis based on other attributes: for this type of analysis, attributes other than fuzzy attributes are chosen. The graph shows the distribution of records for decision types. For each graphical report, only one attribute can be selected. Figure 22 shows the graphical report generated for other attributes.

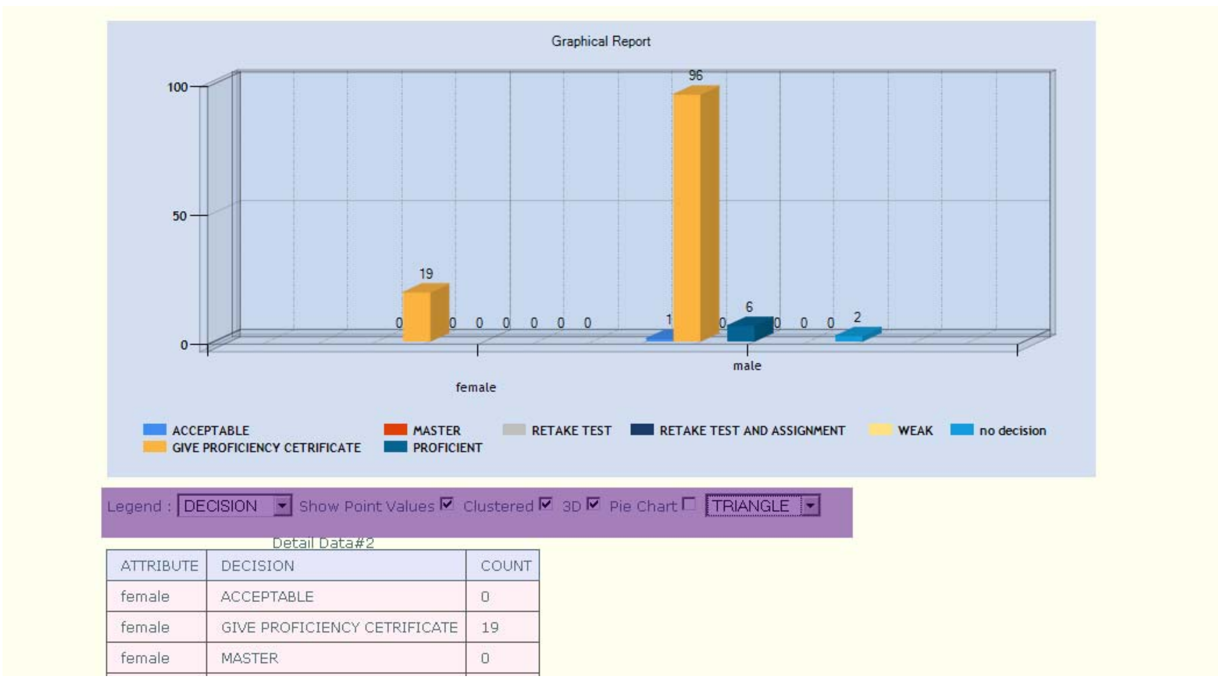


Figure 22: Gender wise decision graph

5.6.2 Detail Graphical analysis: In this analysis, two lists of other attributes are provided. The first list allows user to select a single item from the list, the second list allows multiple item selection. Figure 23 shows the result of the analysis in bar graph and pie chart for LAFED Dataset, fuzzy set=1, rule set= 1.

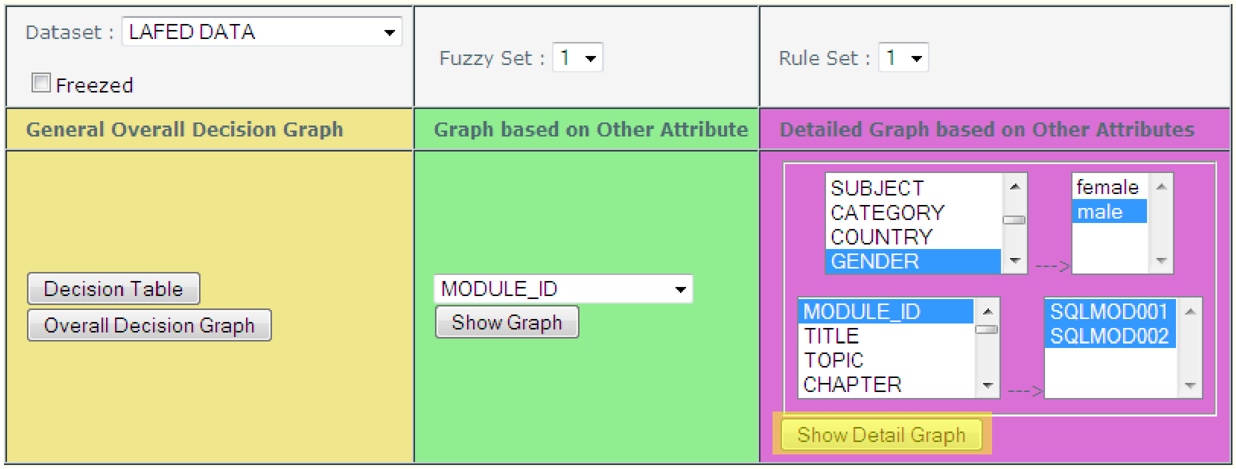


Figure 23: Detail analysis- Customized control selection

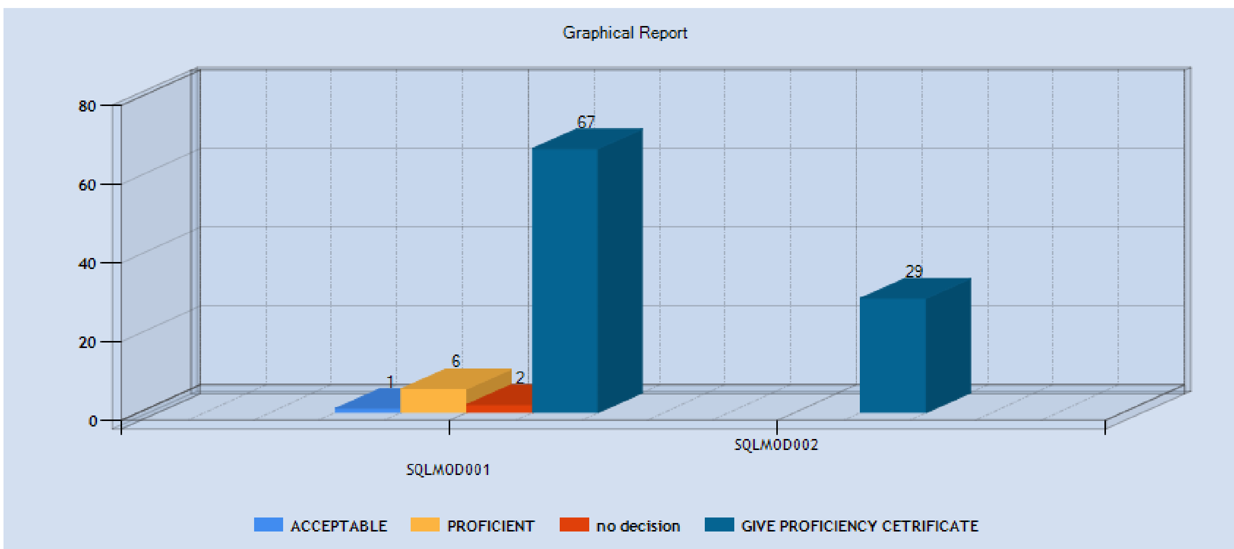


Figure 24: Detail analysis- Bar chart for gender 'Male'

5.6.4 Graph tools: Graphical analysis is provided with graphical tools.

- Chart types: Two types of chart is available in the application. Bar and Pie chart.
- 2D and 3D: Application support 2D and 3D charts.

- Clustered and non-clustered: clustered graph is helpful for viewing multiple series bar graph whereas non-clustered graph shows the bars in a bar graph in different axes.
- Point values: This type of graph is used to show or hide y values of points.
- Legend: The legends can be changed to “decision” or pre-selected “attribute”.
- Membership graph: Triangular and trapezoidal membership graphs are supported in this application.

5.6.5 Revise: User may want to make changes in the existing dataset to generate a better result set. This step helps user to do so.

5.7 Mining Phase: Mining phase helps to analyze a raw input using within the application depending on already learned data. After the fuzzy sets are finalized and stored in the database, the application provides support to analyze untrained data.

Steps involved in the mining phase are:

Step 1: Select dataset- This step allows the application to retrieve the latest optimum data for a selected dataset. The data includes fuzzy set and rule set.

Step 2: Add new data- In this step user can define a new set of single or multiple records and creates schema for selected dataset of step 1.

Step 3: Process data and see the results- After step 2, user can click on “Done” button to process the data. The processing is done against optimum fuzzy set and rule set for both triangular and trapezoidal membership functions.

5.8 Limitations of Web-FDM:

- This tool does not support clustering decisions. By clustering it would be possible to see distribution format.
- The tool does not support multiple roles.
- The tool assumes that the user knows PL/SQL language. Insert data process requires knowledge of insert script query of Oracle.
- Number of rules in the system is huge. For a 3 fuzzy attribute, 5 fuzzy category database user needs to input 125 rules.
- The tool deals with full dataset only. If we want to perform data mining, the data needed from the user has to be a full data set. Data mining with partial dataset is not possible.
- No future data prediction is available in the tool. Data mining is only possible with full dataset for the full time length of data of other objects.

To overcome these limitations, this thesis has considered Comb's Method of Rapid Inference method. This rule reduction technique significantly decreases the number of rules needed for the decision making process. The following steps have been taken to overcome the limitations of Web-FDM:

- ARDIF Database has been expanded to handle Comb's inference Method.
- Interface have been modified to make Combs inference Method work.

- To control the exponential increase in number of rules, the Combs method has been applied.

- Data mining is combined with data prediction of partial dataset.

CHAPTER 6

COMBS METHOD OF RAPID INFERENCE MODEL

Rules in a fuzzy system are defined by intersection of input subsets. When the input subset increases, number of combination for rules increases exponentially. Suppose a fuzzy model has 5 input subsets. To generate the rule matrix, a two input, one output system would generate 25 rules. And a six input system would create 15, 625 rules. So each time we are going to increase input number by one, it actually increases the number of rules exponentially, which is totally not feasible.

There are already many algorithms in the industry that decreases the number of rules and select those rules which makes most contribution the inference output, like genetic algorithm, neural networks, but they do not work on the main reason of the explosion problem.

By using propositional logic rules Combs showed that for a two input system, instead of making all the inputs triggering the output , we can make the rules dependent on only one input and combine all the inputs with “OR” to get the final decision.

6.1 Combinatorial Explosion Problem

6.1.1 Conventional approach: Intersection Rule Configuration (IRC):

The general format of the rule is: If the temperature is cool and the rate of temperature change is decreasing then the furnace output should be moderately high.

Here if *p*=temperature

q= rate of temperature change

r= furnace output

That means if (*p* and *q*) then *r* …………………………… (1)

w= intersection value of any number of antecedent elements in the rule configuration and called Intersection rule Configuration (IRC)

Then (1) becomes

If *w* then *r* ……………………………………………… (2)

P and *q* changes according to the state of the two attributes and results change of *w* hence different relation with *r* hence producing a different rule.

7.1.2 Union Rule Configuration (URC):

Using propositional logic we can transfer the rule configuration to a different structure.

[(*p* and *q*) then *r*] is equivalent to [(*p* then *r*) or (*q* then *r*)] (3)

[(*p* or *q*) then *r*] is equivalent to [(*p* then *r*) and (*q* then *r*)] (4)

[*p* then (*r* and *s*)] is equivalent to [(*p* then *r*) and (*p* then *s*)] (5)

[*p* then (*r* or *s*)] is equivalent to [(*p* then *r*) or (*p* then *s*)] (6)

From basic propositional logic we know that

P implies *r*

⇔ If *p* then r

⇔ Not *p* or r……………………………………………………………… (7)

We can build a truth table to prove this:

P	**Not *p***	**R**	**Not *p* or r**	**If *p* then r**
T	F	T	**T**	**T**
T	F	F	**F**	**F**
F	T	T	**T**	**T**
F	T	F	**T**	**T**

Figure 25: Proof by truth table

The truth table fig: 25 prove equation (4). This means that if *p* is true then *r* must also be true.

The bolded column shows it is true.

This starter truth table can help us prove eq (4). The bolded columns are identical.

P	Q	R	not p	Not q	*P* and q	Not(*p* and q)	(*p* and q) then r ⇔not(*p* and q) or *r*	*P* then r ⇔ Not *p* or r	*Q* then *r* ⇔ Not *q* or r	(*p* then *r*) or (*q* then r)
T	T	T	F	F	T	F	**T**	T	T	**T**

T	T	F	F	F	T	F	**F**	F	F	**F**
T	F	T	F	T	F	T	**T**	T	T	**T**
T	F	F	F	T	F	T	**T**	F	T	**T**
F	T	T	T	F	F	T	**T**	T	F	**T**
F	T	F	T	F	F	T	**T**	T	T	**T**
F	F	T	T	T	F	T	**T**	T	T	**T**
F	F	F	T	T	F	T	**T**	T	T	**T**

Fig: 26 Truth table to prove the equation

Then it can be proved (3) with formal propositional logic [Combs, 1997]:

(*p* and q) then *r* ……………. The initial IRC

Not (*p* and q) o*r r* ………………. Material Implication

(not *p* or not q) or *r* ………… by DeMorgan's Law

Not *p* or (not *q* or r) …………………….. by association

(not *q* or r) or not *p* ………………..by commutation

(*q* then r) or not *p* ………………by material implication

((*q* then r) or not p) or *r* ………………..by addition

(*q* then r) or (not *p* or r)………………… By association

(*q* then r) or (*p* then r)…………………… By material implication (*p* then r) or (*q* then r)…………by commutation yields the URC

6.1.3 Multiplication of rules

Let us consider an example with which Comb has tried to present a very common example to prove his theory [Combs, 1997]. When people gets auto insurance, the most common attributes those are considered to calculate a person's premium is age, healthier status, health history, job risk, foreign risk, etc. Using fuzzy logic, Combs tried to build the premium calculation technique and tried to find out how one's premium may vary depending on these attributes.

Let us begin with two inputs, one output model.

Age/Health	**Youthful**	**Young**	**Middle aged**	**Mature**	**Old**
Excellent	Very Low (1)	Low (6)	Mod. Low (11)	Mod. Low (16)	Mod. (21)
Good	Low (2)	Mod. Low (7)	Mod. Low (12)	Mod (17)	Mod. High (22)
Average	Mod. Low (3)	Mod. Low (8)	Mod. (13)	Mod. High (18)	Mod. High (23)
Below Avg	Mod. Low (4)	Mod (9)	Mod. High (14)	Mod. High (19)	High (24)
Poor	Mod (5)	Mod. High (10)	Mod. High (15)	High (20)	Very High (25)

Figure: 27: Conventional method configuration of rules

This means:

Rule 1: If the person is Youthful and his health is excellent, THEN the insurance premium will be very low

…………

Rule 25: IF the person is Old and his health is Poor, THEN the insurance premium will be Very High.

6.1.4 Fuzzy Logic Inference Methods:

There are some popular methods available for the defuzzification process that can be used in fuzzy logic [Combs, 1997]

6.1.4.1 Mamdani Method of inference:

The method was first proposed in 1975 by Ebrahim Mamdani [Mamdani, 1975] to control a steam engine and boiler combination by adding a set of linguistic rules of human operators. This method was established based on the research of Lotfi Zadeh's paper of fuzzy algorithms for complex systems and decision processes [Zadeh, 1973]

In this method, the output membership functions are fuzzy sets. In most cases the output membership function is considered as a single spike rather than a fuzzy set. This is called *singleton* output membership function.

The steps involved in the Inference method is :

6.1.4.1.1 Fuzzify Inputs: In this step the degree of inputs in the system are calculated based on the membership functions. The range of membership values are always between 0 and 1.

6.1.4.1.2 Apply fuzzy operator: when the degree of each input is determined, the fuzzy operator is used to get one number that represents the result of antecedent of a particular rule.

For FAOES, when we have a rule:

(If sales is poor AND orders is average AND products is poor) then (Give Warning)

Here the input antecedents are related with

AND operator which supports 2 methods,

- Min (minimum)
- Prod (product).

OR Operator supports

- Max (Maximum)
- Probor (Probabilistic Or): Probor (a,b)= a+b –ab

For WEBFDM 1.0 AND operator is used. For WEB-FDM 2.0 two the OR operator is used which means the Maximum weight of each of the input antecedent is considered.

6.1.4.1.3 Apply Implication Method: Weight is assigned to each rule which is usually 1. If we apply the MIN method in this step, the output is truncated at the appropriate fuzzy set which is determined from the input process.

6.1.4.1.4 Aggregate All Output: In this step all the outputs weight are combined considering all overlap values of input antecedents. The output of the aggregation is a fuzzy set for each output variable.

6.1.4.1.5 Defuzzify: The input at this step is the aggregated output fuzzy set, and output is a single number. The most popular method is Centroid Defuzzification. This fuzzification can be done by:

- Centroid Calculation - In language of physics, centroid is the geometric center of any object. It is also known as the center of gravity or center of mass.

  The centroid of a combined object can be calculated by, finding the centroid Ci for each area Ai, and then compute the total centroid by summation.

  C= Sum (CiAi)/ Sum (Ai)

- Bisector
- Middle of Maximum
- Largest of Maximum - If any multiple rules can be fired for any certain input set, the maximum relative membership value will be used for the output.
- Smallest of Maximum

For WEB-FDM we have 2.0 we have implemented the Largest of Maximum Method to find out the output value in decision.

The degree to which the antecedent is true effects on the strength, how the rule will fire.

After using these four steps we can create a union rule matrix for the antecedent membership values, where the summation of columns of the inputs can represent an output membership value.

6.1.4.2 Comparing conventional and proposed rule configuration method with example:

For the Insurance premium system, let us consider a person whose age is 40, and health status is 0.2.

Then from the Membership function we find out,

For Age= {µYf=0, µY=0.3, µMA=.7, µM=0, µO=0}

Health Stat= {µE=0, µG=0, µA=0, µBA=0.8, µP=0.2}

Based on Fig 27: we see that rule 9, 10, 14, 15 may fire.

Age/Health	**Youthful**	**0.3**	**0.7**	**Mature**	**Old**
Excellent	Very Low (1)	Low (6)	Mod. Low (11)	Mod. Low (16)	Mod. (21)
Good	Low (2)	Mod. Low (7)	Mod. Low (12)	Mod (17)	Mod. High (22)
Average	Mod. Low (3)	Mod. Low (8)	Mod. (13)	Mod. High (18)	Mod. High (23)
0.8	Mod. Low (4)	Mod (9)	Mod. High	Mod. High	High (24)

			(14)	(19)	
0.2	Mod (5)	Mod. High (10)	Mod. High (15)	High (20)	Very High (25)

Fig: 28 Conventional method rule distribution

Age/Health	**Youthful**	**0.3**	**0.7**	**Mature**	**Old**
Excellent	Very Low (1)	Low (6)	Mod. Low (11)	Mod. Low (16)	Mod. (21)
Good	Low (2)	Mod. Low (7)	Mod. Low (12)	Mod (17)	Mod. High (22)
Average	Mod. Low (3)	Mod. Low (8)	Mod. (13)	Mod. High (18)	Mod. High (23)
0.8	Mod. Low (4)	0.3	0.7	Mod. High (19)	High (24)
0.2	Mod (5)	0.2	0.2	High (20)	Very High (25)

Fig: 29 Conventional method rule configuration Example

Now we need to apply some inference method to find out exactly which rule we need to fire. Based on this situation the premium can be both Moderate and Moderate High, which is not feasible.

Using Min Operation:

Premium= {Low, Moderate Low, Moderate, Moderate High, High}

= {0,0,0.3,0.2,0}

After applying it to the fuzzy decision diagram

We find out:

Applying certain defuzzification method:

Max Method: taking the maximum of the output membership values, we get the premium should be 0.3 which falls in Moderate range.

Centroid Method:

Premium= Sum (centroid for all the triangle)/ sum(all output membership values)

=(0x.125)+(0x0.25)+(0.3x0.5)+(0.2x0.75)+(0x1.0)/(0+0+0.3+0.2+0) = 0.525

This falls within the Moderate range too.

Instead if we change the output subset from seven to five and use Combs Rapid inference method, using Union rule matrix we can create figure 30.

Age	Youthful Then Low (1)	Young Then Moderately Low (3)	Middle Aged Then Moderate (5)	Mature Then Moderately High (7)	Old Then High (9)
Health status	Excellent	Good	Average	Below Average	Poor

	Then Low(2)	Then Moderately Low (4)	Then Moderate (6)	Then Moderately High (8)	Then High (10)

Figure 30: Combs rule configuration method with two fuzzy attributes

Here we can say that *p* represents Age and *q* represents Health Status, *r* represents the insurance premium. Since each of these 10 fuzzy rules can be separated using logical OR, we can say that there are only two rules, one for each input. Then our rules become:

[(Youthful Then Low) OR (Young then Moderately Low) OR (Middle Aged then Moderate) OR (Mature then Moderately High) OR (Old then High)]

OR

[(Excellent Then Low) OR (Good then Moderately Low) OR (Average then Moderate) OR (Below Average then Moderately High) OR (Poor then High)]

Looking at this example, one can be easily confused thinking, if a young person has poor Health Status, Then his premium would be either Moderately Low or High. How can one person have both kinds of premiums? This OR we are using here is logical OR, Either (*p* then r) or (*q* then r) but not both.

So For a person if his age is 40, and health status is 0.2

Then from the Membership function we find out,

For Age= {μYf=0, μY=0.3, μMA=.7, μM=0, μO=0}

Health Stat= {μE=0, μG=0, μA=0, μBA=0.8, μP=0.2}

Age	Youthful Then Low (1)	Young Then Mod. Low (3)	Middle Aged Then Mod (5)	Mature Then Mod. High (7)	Old Then High (9)
Health status	Excellent Then Low(2)	Good Then Mod. Low (4)	Avg Then Mov. (6)	Below Avg. Then Mod. High (8)	Poor Then High (10)

Age	Youthful Then Low (1)	0.3	0.7	Mature Then Mod. High (7)	Old Then High (9)
Health status	Excellent Then Low(2)	Good Then Mod. Low (4)	Avg Then Mov. (6)	0.8	0.2

By Min Method (AND) we can say Age= 0.3

Health Stat= 0.2

Age	0	0.3	0.7	0	0

Health status	0	0	0	0.8	0.2

By using the union rule matrix:

We get ->

Premium	0	0.3	0.7	0.8	0.2

If we apply the various de fuzzification:

a. Max Method: premium would be 0.7 = Moderate

b. Centroid Premium would be = 5.524= Moderate

We are getting the same kind of result as we did in URC, instead we are using only 10 rules, not 25.

Comb is not advocating the transformation of existing rules from one format to another. Instead this is a suggestion for a different model where the problem domain can be pictured using a totally separate rule configuration (URC) method.

CHAPTER 7

WEB-FDM 2.0

As part of this thesis, The Web-FDM 2.0 project is developed to overcome the limitations of Rule explosion problem of previous WEB-FDM project. The following issues are solved in Web-FDM 2.0:

- Using Combs method of rapid inference, number of rules is reduced at a significant rate.
- Improved user interface design
- Implemented 3 forms of calculation to apply Combs rapid inference method.
- Comparison graphical presentation of the previous rule approach and Combs Rule approach.

7.1 Components

- Oracle 9i: The database is hosted in a Oracle 9i server
- FDB3 Database: The database schema name is FDB3 which consists of all the related tables to create the ARDIF Engine, stored procedures, functions, triggers, sequences to support the fuzzy logic.
- ASP.net Charting control: To visually represent the decision of the databases Charting Control DLL file is used.

- Data Warehousing: To store data to analyze fuzzy logic and compare results of different sets.

- TierDeveloper 6.1: The software is used to generate Data Access Layer. This application maps oracle table in different objects and gives direct access to different columns of the tables to have a faster calculation. DAL layer also provides clean code and code reuse flexibility.

- IIS6: IIS is Internet server where the server side of the application is stored. The server separates the C# code written in the application from the HTML and helps to provide a streamline process.

7.2 Working procedure of Web-FDM 2.0:

7.2.1. In this step steps to add combs rule in the system will be displayed. An extra step to add rules for Combs is added. User will select this step to add new rule set.

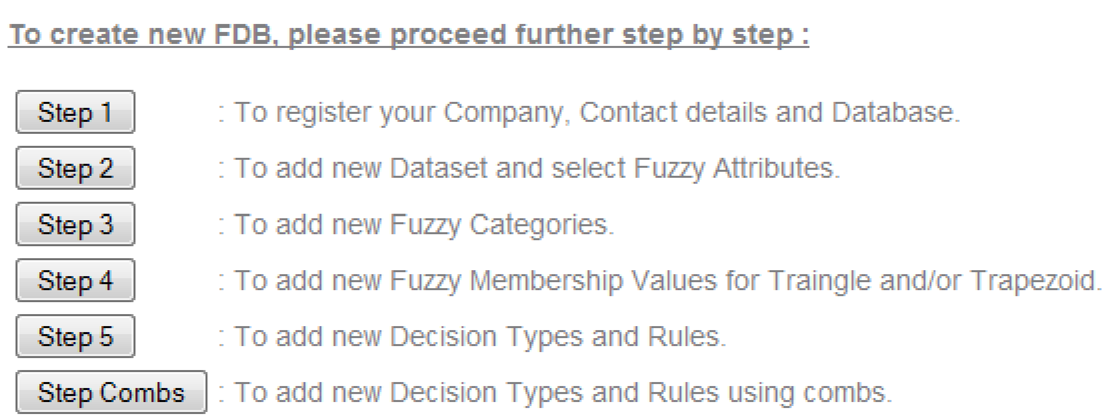


Figure 30: Steps of Web-FDM 2.0 for Combs Method implementation

7.2.2 In this step user will add a new rule set. Each rule has a one to one mapping with the fuzzy attribute and decision.

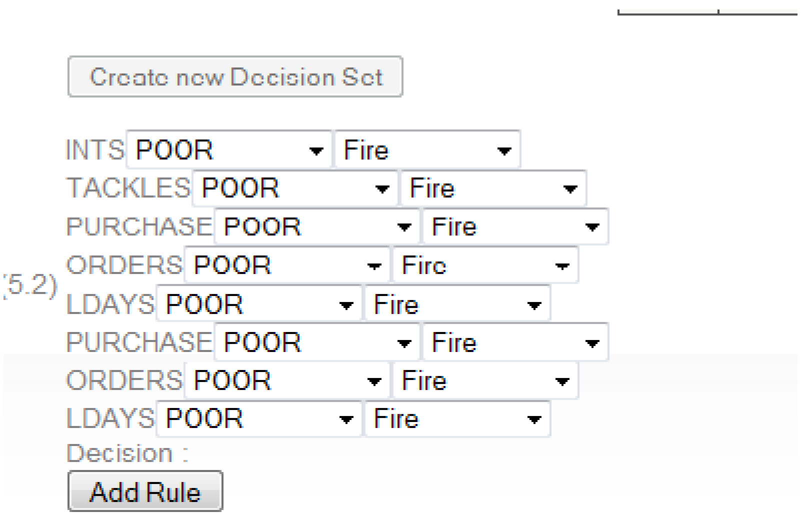
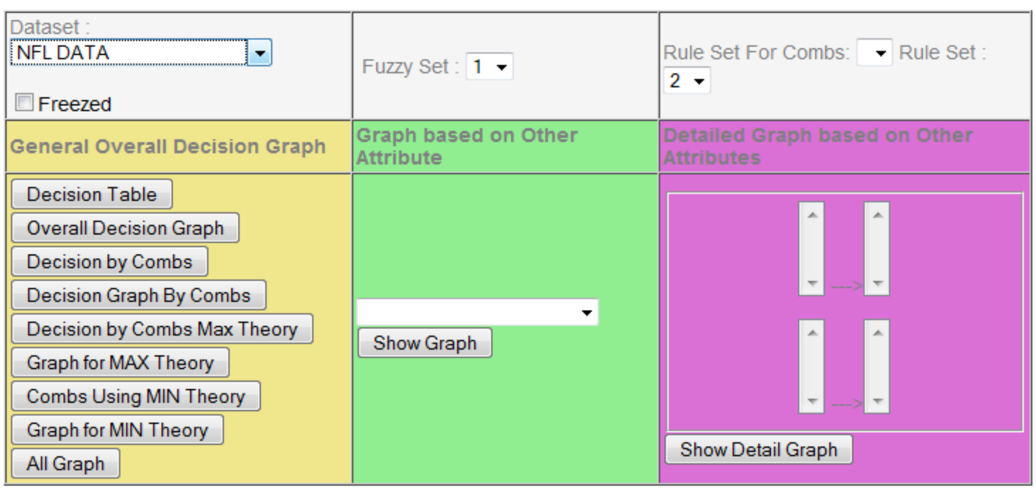


Figure 31: Adding a new rule set in Combs Method in Web-FDM 2.0

7.2.3 Analysis: In this step, user can view graphically the decisions made by rule set without Combs inference and 3 process used for Combs inference method. All four graphs are shown side by side for an easy comparison.



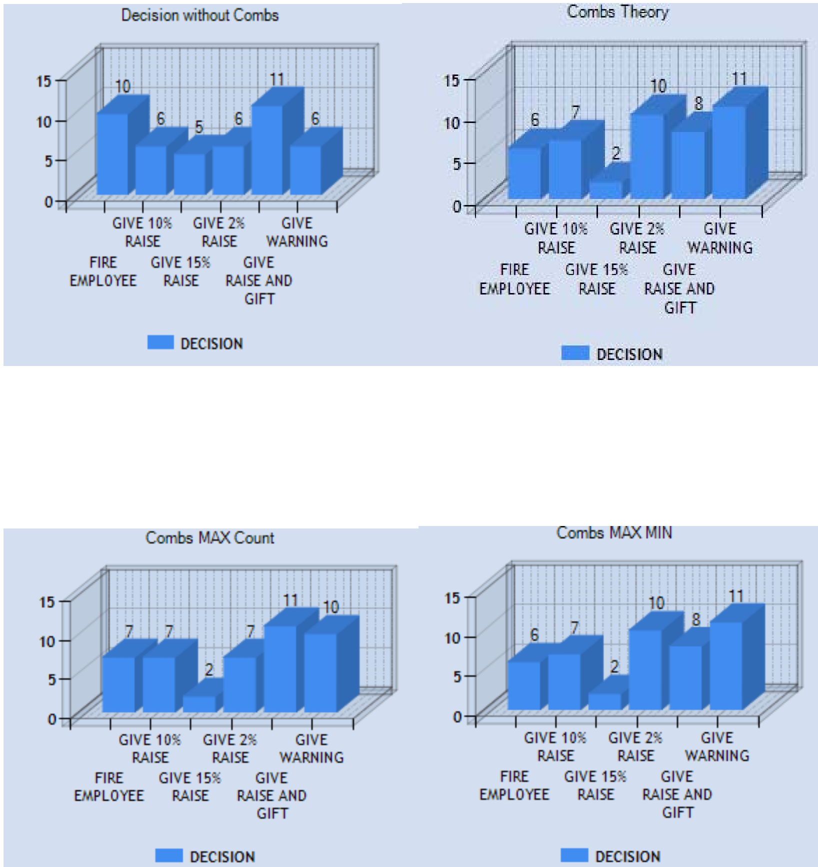


Figure 32: Analysis of 4 approaches of rule set in Web-FDM 2.0

Web-FDM 2.0 focuses mainly on reducing the huge number of rules needed in Web-FDM. By implementing 3 forms of calculation, we find out that the results found out using Combs and not without Combs are much more similar, in some cases better and evenly distributed over the decision subsets. From Figure 32, the following result set is established

Method used	**Fire Employee**	**10% raise**	**15% raise**	**2% Raise**	**Raise and Gift**	**Give Warning**
Without Comb	10	6	5	6	11	6

Max theory using Combs	6	7	2	10	8	11
Max Count theory using Combs	7	7	2	7	11	10
Max Min Theory using Combs	6	7	2	10	8	11

Table 12: Result from Web-FDM for FAOES_2E Database

From the result set we can see that we are getting same kind of decisions using Web-FDM but having only 15 rules whereas Web-FDM, we had more than 100 rules. This will make the rule input system easier and more straight forward.

7.3 Mathematical model used in Web-FDM 2.0:

3 different approaches have been used in Web-FDM 2.0 for the rule reduction technique. All three approaches show nearly similar distribution of decisions.

7.3.1 Max Approach: when there is an overlap in the fuzzy categories, the maximum overlap amount is chosen, then in the decision making process, the fuzzy attribute with the highest weight is chosen as the final decision

7.3.2 Max Count approach: in this approach, the decision making process depends on the fuzzy category that appears maximum time for all the fuzzy attributes. For FAOES_2E, if an employee has “Poor” category in “Sales”, “Poor” category in “Orders” and “Excellent”category in “Products”, then “Poor is chosen in the final category for decision making process and the weight of that category and attribute contributes more.

7.3.3 Max Min Method: In this approach, when there is an overlap in the fuzzy categories membership values, the minimum overlapping amount is chosen, then in the final decision making process, the maximum fuzzy attribute weight is chosen.

CHAPTER 8

DATA PREDICTION

8.1 Background of the research:

In the mining phase of Web-FDM, only full dataset is accepted to predict any given object to fall within a certain group of objects. If all the criteria and attributes of the objects are known, then it can be categorized. Minnesota State University, Mankato Database research group came up with an idea to predict, what would be the status of an object in a future data considering its current status and attributes. It will help the system to give a proper direction how the decision can be improved. For OES database, depending on the current sales, orders and products number, an employee's future performance can be predicted. This can help the employee have a better understanding of his work progress and can improve the performance following an optimum path proposed by Web-FDM 2.0.

The idea of extension of WEBFDM to data mine using predicted data was evolved from the Thesis paper written for total or Partial Knee Replacement database [Azarbod, 2011]. Total knee replacement is very frequent operation in today's society. The increase of this procedure has also increased the need of rehabilitations from this surgery. How a knee replacement acts over time of the rehabilitation process shows the patient's progress of recovery.

In Knee Replacement Database (KRT), a set of patient data is used to predict the recovery time period in the rehabilitation process. In many cases, it can be possible that we have only data of few weeks. It would be more useful, if we can predict an optimal path for the patient to recover in a certain amount of time period. KRT database is an ideal case for implementing the data prediction methodology with both full and partial dataset.

8.2. KRT Database (Knee Replacement Therapy Database)

8.2.1 Background of KRT (Knee Replacement Therapy Database)

The KRT database acts as an expert for the future knowledge base system. Past research shows that patients healing time varies dramatically based on various factors. Such as, amount of workout, weight, number of diseases, and number of days on medications, nutrition amount and number of medication the patient is taking, etc. Each of these factors are entered and connected to a patient entity.

8.2.2 Design and implementation of KRT Database

KRT database was implemented in Oracle 9i. Appendix D1 has the database script for KRT database.

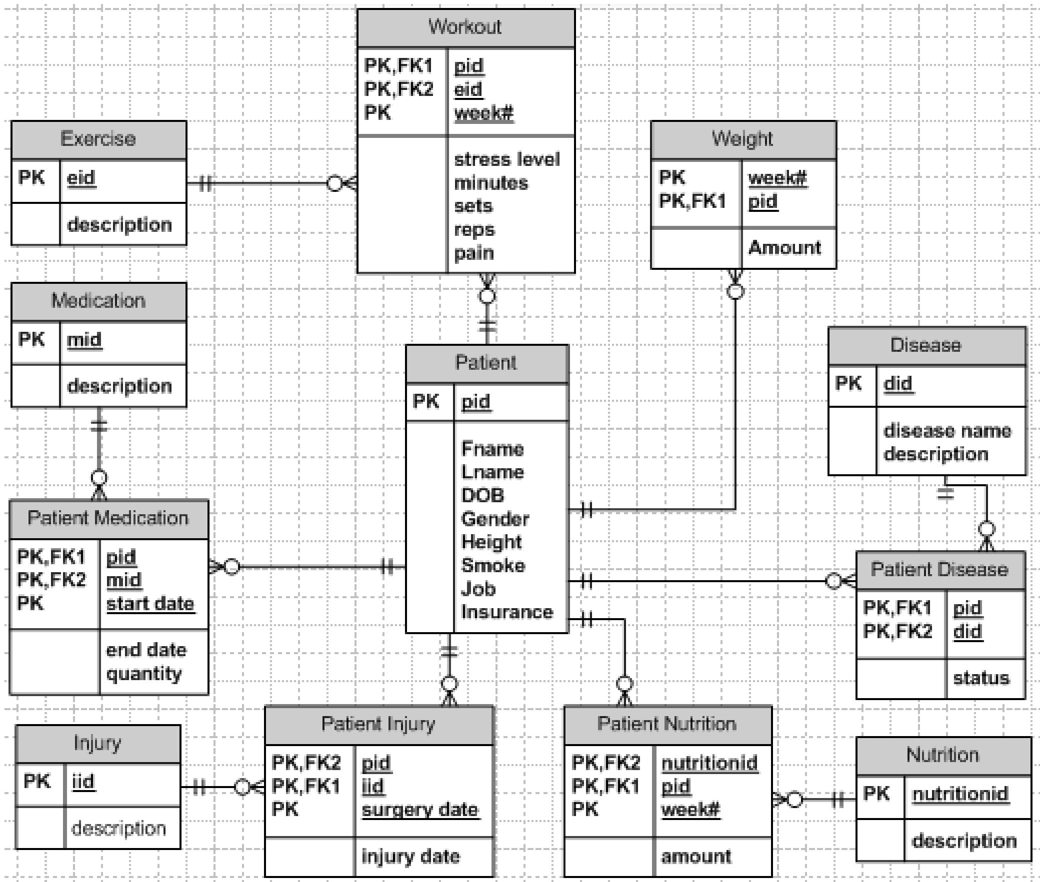


Figure 33: Data Model for KRT Database

KRT Tables
DISEASE
EXERCISE
INJURY
MEDICATION
NUTRITION
PATIENT
PATIENTDISEASE
PATIENTINJURY
PATIENTMEDICATION
PATIENTNUTRITION
WEIGHT
WORKOUT

Figure 34: List of tables in KRT database

8.2.3 Methodology, results and analysis

At first, a patient set is created and data is populated to simulate the process of rehabilitation.

Patient ID	Expected Decision
1	11-12 weeks
2	13-14 weeks
3	13-14 weeks
4	15-16 weeks
5	11-12 weeks
6	13-14 weeks
7	15-16 weeks
8	9-10 weeks
9	11-12 weeks
10	13-14 weeks

Table: Patient ID and Expected Decision

On the second phase, the fuzzy attributes, fuzzy categories, membership values, and rules are determined. While recovering, the expert (a physical therapist) the physical therapy information of a patient each week to determine the amount of healing. Figure 35 shows all the fuzzy attributes of KRT database

FAID	ATTRIBUTE
380	AGE
381	WRKOUT
382	PAIN
383	STRESS
384	DISEAS
385	NO_MED
386	DAYS_MD
387	NUTRIT

Figure 35: Fuzzy Attributes used in Web-FDM

Once the Fuzzy attributes are determined, the fuzzy categories are established.

FUZZYCATID	NAME
9	Low
10	Mid
11	High

Figure 36: Fuzzy Categories used in Web-FDM

For each fuzzy category and fuzzy attributes, membership values are established. Each membership value was calculated for both Triangle and Trapezoid function.

MEMBID	WTID	FAID	ATTRIBUTE	FUZZYCAT	START_POINT	MID1	END_POINT	GRAPH
1342	23	380	AGE	Low	25	39	54	TRI
1344	23	380	AGE	Mid	44	55	67	TRI
1346	23	380	AGE	High	59	72	85	TRI

Figure 37: Fuzzy Attributes and membership values for Triangle function

The last step is to identify the fuzzy rules depending on the category set, membership values.

Existing Rules

DECISION	AGE	WRKOUT	PAIN	STRESS	DISEAS	NO_MED	DAYS_MD	NUTRIT
optimum recovery with 9-10 weeks	Low	High	Low	Low	Low	Low	Low	Low
optimum recovery with 9-10 weeks	Low	High	Low	Low	Low	Low	Low	Mid
optimum recovery with 9-10 weeks	Low	High	Low	Low	Low	Low	Low	High
optimum recovery with 9-10 weeks	Low	High	Low	Low	Low	Low	Mid	Low
Good recovery with 11-12 weeks	Low	High	Low	Low	Mid	Mid	Mid	High
Slow recovery with 13-14 weeks	Low	High	Low	Low	High	High	High	Low
Slow recovery with 13-14 weeks	Low	High	Low	Low	High	Mid	Mid	Low
Good recovery with 11-12 weeks	Low	High	Low	Low	High	Low	Low	Mid
Good recovery with 11-12 weeks	Low	High	Low	Low	High	Low	Mid	High
Good recovery with 11-12 weeks	Low	High	Low	Low	Mid	Low	Low	High

Figure 38: Fuzzy rules for KRT Database

Implementation of fuzzy rules created the following decisions were generated for some specific patients.

ID	FUZZYCATSETID	GRAPH	RULESETID	DEC_TRI	DEC_TRP
1	1	TRI-TRP	2	Poor recovery with 15-16 weeks	Poor recovery with 15-16 weeks
2	1	TRI-TRP	2	no decision	no decision
3	1	TRI-TRP	2	Poor recovery with 15-16 weeks	no decision
4	1	TRI-TRP	2	no decision	no decision
5	1	TRI-TRP	2	Good recovery with 11-12 weeks	Good recovery with 11-12 weeks
6	1	TRI-TRP	2	Slow recovery with 13-14 weeks	Slow recovery with 13-14 weeks
7	1	TRI-TRP	2	Poor recovery with 15-16 weeks	Poor recovery with 15-16 weeks
8	1	TRI-TRP	2	no decision	no decision
9	1	TRI-TRP	2	Slow recovery with 13-14 weeks	no decision
10	1	TRI-TRP	2	Poor recovery with 15-16 weeks	no decision

Figure 39: Decision table of first implementation for KRT database

The WEB-FDM worked as a fuzzy active system that handles patient related data, and based on the applied rule, the system could make good and expert decision regarding patient's healing time. Though the system could not predict when the patient might heal, which is handled in this Thesis project.

8.3 Fuzzy predictor based on a partial data set:

From the KRT Database a data prediction formula has been established that can predict future data set based on a partial set of data for any particular test object. A sample patient has been chosen who has gone through the physical therapy for 5 weeks.

Week #	Total Progress in Healing
Week 1	1.00347222
Week 2	2.0069444
Week 3	3.010416667
Week 4	4.013888889
Week 5	5.017361111

Figure 40: Summary of Sample patient data for 5 weeks.

In this sample, the patient has been healing at a rate which is slightly faster than average rate. If the patient heals at a maximum rate possible, the predictor would decide the healing in 11 weeks. For slowest rate possible the decision would be 14 weeks. Figure 37 shows the graphical presentation of the future data of this particular patient.

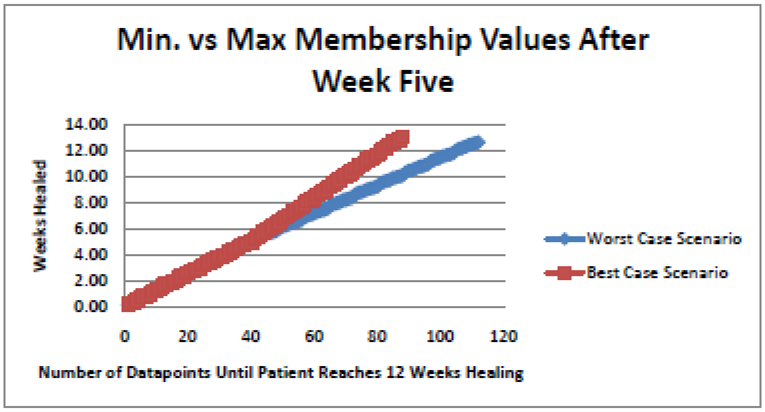


Figure 41: Minimum Vs Maximum Healing trails a patient can follow after week 5

8.4 Mathematical Model to Predict Result:

To create a decision support system for partial decision, all the fuzzy attributes have been quantified to generate the final decision based on partial data set. [Azarbod, 2011]

From the Thesis paper [Azarbod, 2011] the following equation has been derived:

Equation 1: Total Healing Represented by the Summation of Each Week for the Period it Takes for the Patient to Heal

$$TotalHealig = \sum W_n Heal = W_1 Heal + W_2 Heal + \ldots + W_n Heal$$

Where

Healing= Old Healing + New Incremental Healing

8.5 Converting the Fuzzy System into a Mathematical model

To convert the fuzzy system to a mathematical model, the fuzzy components must be parameterized depending on the patient's behavior from previous weeks, then convert the result with already existing fuzzy rules to predict a viable decision.

Step 1: The fuzzy components are considered as parameters in the data prediction phase which are same at fuzzy attributes. For KRT Database

1. Age
2. Workout
3. Pain
4. Stress
5. Number of disease
6. Number of medication
7. Days of medication
8. Total nutrition

All the fuzzy parameters are then assigned a weight, which indicates how much the parameter is contributing in the decision making process.

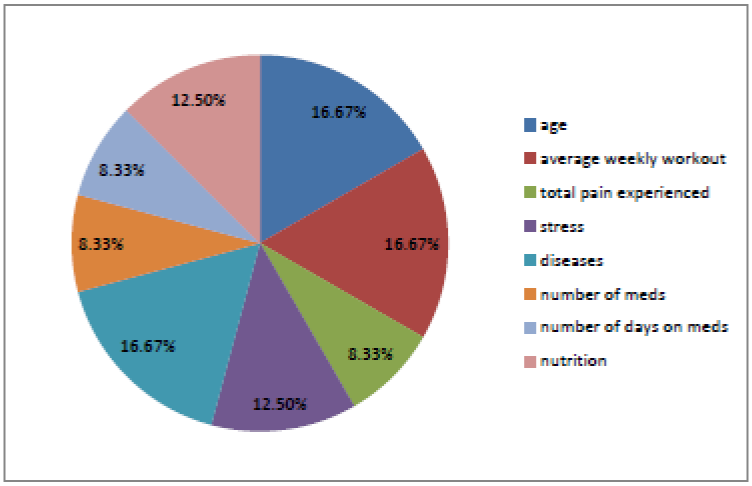


Figure 42: Weight of each fuzzy parameter which indicates the contribution of parameters in the decision process.

Step 2: Then all the fuzzy categories are converted to a numerical constant.

Step 3: Then each parameter is assigned "dynamic" or "static" value, depending how they contribute in the decision making process over the prediction time. Figure 43 shows the summary of parameter identification.

Attribute	Parameter	Parameter Flexibility	Weight	Ideal Fuzzy Category
age	a	static	16.67%	LOW
average weekly workout	w	dynamic	16.67%	HIGH
total pain experienced	p	dynamic	8.33%	LOW
stress	s	dynamic	12.50%	LOW
diseases	d	dynamic	16.67%	LOW
number of meds	m	dynamic	8.33%	LOW
number of days on meds	z	dynamic	8.33%	LOW
nutrition	n	dynamic	12.50%	HIGH
			100.00%	

Figure 43: Parameter identifier Summary

Depending on the equation derived in the Thesis Paper [Azarbod, 2011], the following data (Figure 40) can be calculated for a patient for 1 week activity. The data shows the healing amount after 1 week is 1.04 which is slightly greater than the average rate.

Parameter	Parameter Weight	Fuzzy Category Constant	Fuzzy Parameter Weight	Healing Progress
week 1 a	0.17	1.00	0.17	0.17
week 1 w	0.17	1.33	0.22	0.39
week 1 p	0.08	1.33	0.11	0.50
week 1 s	0.13	1.00	0.13	0.62
week 1 d	0.17	0.75	0.13	0.75
week 1 m	0.08	1.00	0.08	0.83
week 1 z	0.08	1.00	0.08	0.92
week 1 n	0.13	1.00	0.13	1.04

Figure 44: Week 1 of healing for a patient

Depending on how the healing rate is, we can calculate week by week healing progress of a particular patient using this mathematical model.

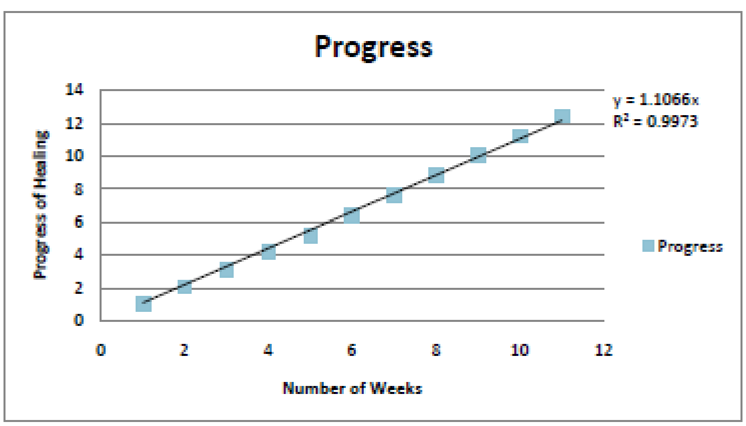


Figure 45: Healing progress of a patient.

Figure 45 show that with e rate of healing slightly greater than 1, the patient will reach the threshold of 12 weeks recovery level before the actual time. Appendix K shows the projected full dataset of a patient for depending on 5 weeks of data (Refer Appendix N1 for partial dataset)

CHAPTER 9

DATA PREDICTION FOR WEB-FDM 2.0

9.1 Steps to apply data prediction in Web-FDM 2.0:

mining phase will take new record(s) similar to full data set that was used to train the database in phase 3 (learning phase) for Web-FDM. Web-FDM can predict if the target data set is similar to data set used in training but cannot handle partial data set which means data is subset of the full data set. For example, in FAOES, 3 years sales data for each employee were used to train database. If the target data is sales data for an employee covering three years then phase 4 can predict how this employee is going to be evaluated but if we only have one year sales data then we cannot predict the employee performance.

To overcome this limitation, we have implemented the mathematical model proposed for data prediction [Azarbod, 2011] in Web-FDM 2.0. The algorithm is followed to generate a full dataset from an input partial dataset for the full time range. The prediction rule has been applied for FAOES_2E database in web-FDM to generate decision for an employee on a future date depending on his current performance. A 9 month partial data for a certain employee has been created as an input. The time length for data in FAOES_2E is 3 years which is equivalent to 36 months. The data prediction approach has been applied to find the performance of the employee after 36 months.

With this prediction module, Web-FDM 2.0 can generate automatically the data for all the fuzzy attributes for a certain time period to convert the partial dataset to a full dataset to match the other entries saved in the database. For FAOES, we can produce the amount of sale, number of orders, and number of products an employee would have after 36 months depending on his performance of 9 months available data. With the data mining module of Web-FDM we can use this full dataset and predict a decision for him after 36 months.

Figure 42 shows the screen where expert will start working on the data prediction phase.

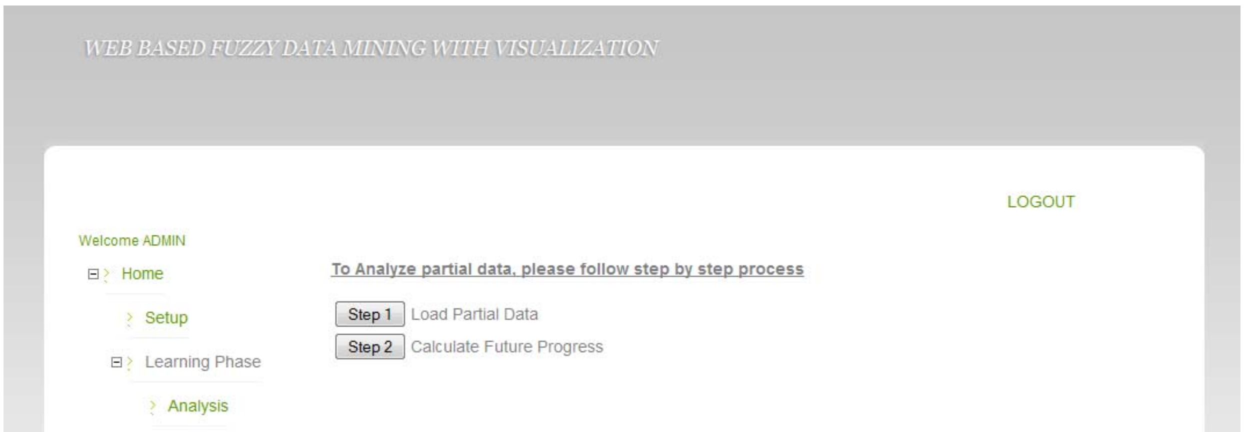


Figure 46: Data Prediction page of Web-FDM 2.0 application

9.1.1 Load Partial Data

9.1.1.1. Identifying fuzzy parameters for prediction:

In this step the fuzzy parameters are figured out. In most cases the fuzzy attributes act as the fuzzy parameters in the prediction process.

9.1.1.2. In this step the fuzzy category constants, rate of progression for each parameter are calculated.

To find this, the Maximum, minimum and average value of each fuzzy parameter is queried from the worktable.

So for FAOES_2E,

Maximum Category constant for net_sales = Maximum net_sales/average net_sales

Minimum category constant for net_sales= Minimum net_sales/average net_sales

9.1.1.3. The rate of attribute for each fuzzy parameter is calculated

The attribute rate has been calculated by the following formula

Attribute rate= Average value of an attribute/ total time length

9.1.1.4 Add Prediction tables in database:

In This step all the tables needed for data prediction is added in the database. 3 tables each for average, maximum and minimum effort are created.

9.1.1.5 Add Data Mine Table

In this table 3 tables each for average, minimum and maximum effort are added in the database for the data mining process at the end of data prediction.

Figure 47 shows the page where user adds the tables.

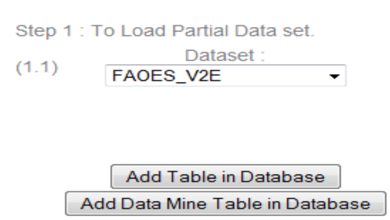


Figure 47: Page where user creates all the tables needed for data prediction and mining

The fuzzy parameters of FAOES_2E are

1. Net_sales

2. Net_orders

3. Net_Products

The application offers a default value for the parameter weight for any database, which is for simplicity, in this version same for all parameter

Parameter weight= 100/Number of parameters

This concludes the identifiers as

Parameter	Parameter weight
Net_sales	33%
Net_orders	33%
Net_products	33%

Table 13: parameter weight for FAOES_2E

9.1.1.6. Add partial data in the database

In this step user will add partial dataset in the database.

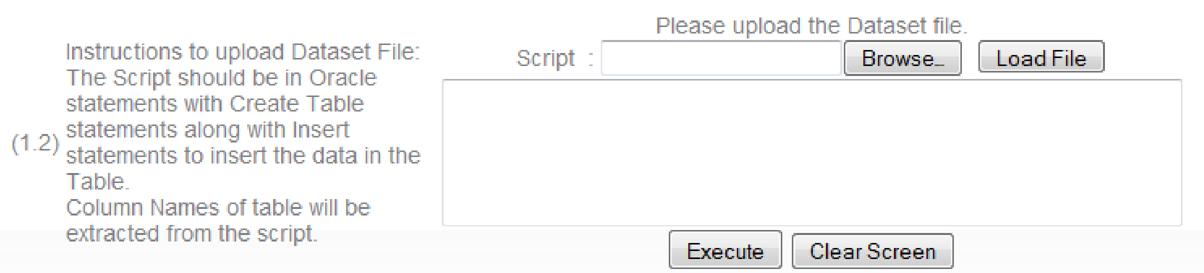
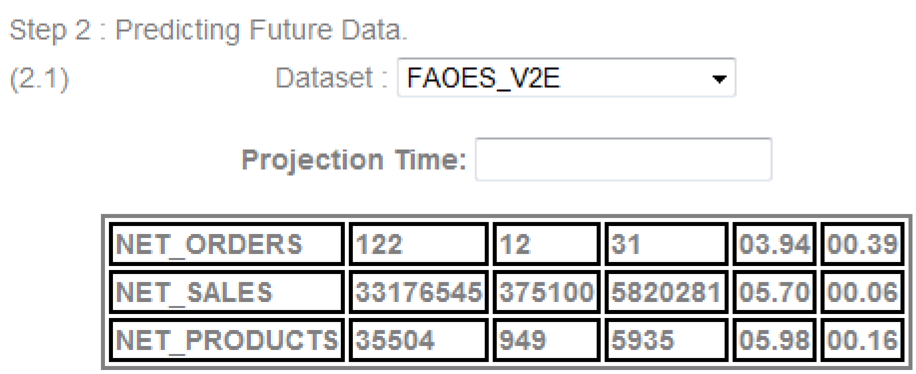


Figure 48: Module to take user input script for partial data insertion

User will provide script to insert partial data for each of the tables where average, maximum and minimum progress will be stored.

9.1.2. Predicting future data:

9.1.2.1. Calculating Max, Min, Average values for all fuzzy parameters:



NET_ORDERS	122	12	31	03.94	00.39
NET_SALES	33176545	375100	5820281	05.70	00.06
NET_PRODUCTS	35504	949	5935	05.98	00.16

Figure 49: Table showing all the related values for FAOES_2E

In this step, the application calculates the maximum, minimum and average value for each of the fuzzy parameters from the worktable. From this values, maximum and minimum category constant are also calculated. Appendix L, M contains the stored procedure that will calculate the future dataset for a given time frame.

9.1.2.2. Calculate decision:

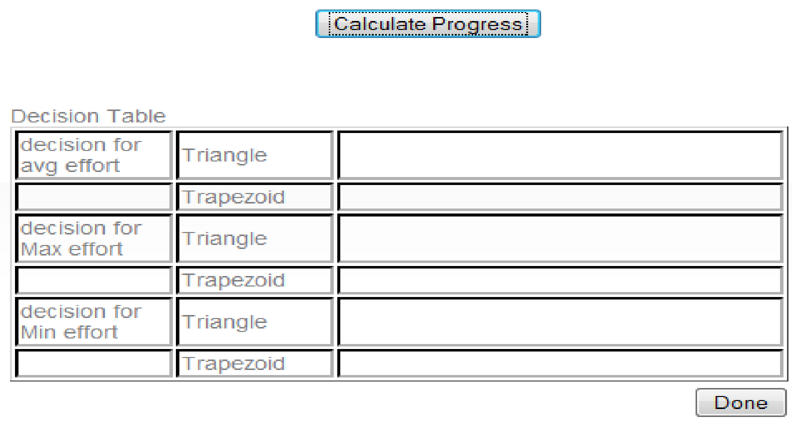


Figure 50: Table where all the decisions will be shown

In this step, the application takes the user input for projection time period, and finds out the decisions after that certain time. In the database, the following steps are performed.

a. Execute CreateProcedureFutureProgress: This procedure will create another procedure for the specific worktable which will eventually calculate the progress amount.

b. Execute _FutureProgress: This procedure will calculate the cumulative progress amount for each dataset.

c. Insert into DataMine tables: in this step, the newly calculated final cumulative values for each attributes are inserted in all the DataMine Tables. Only one row is created in the DataMine table. This table serves as the work table in the decision making process.

d. Execute CreateAllCombs: The CreateAllCombs procedure creates all the necessary tables, procedures, triggers to calculate the final decision for the predicted dataset.

e. Execute _ins_Dec_combs: this procedure matches all the available rules in the database, and finally generates the decisions for the particular test dataset.

CHAPTER 10

TESTING AND VALIDATION

10.1 Applying Combs Inference Method with FAOES_2E

The results produced by Web-FDM 2.0 have been tested using FAOES_2E dataset. As discussed in before, twelve schemas generated in FDM project are used for testing purposes in Web-FDM and the output for schemas – FAOES_v2a, FAOES_v2b and FAOES_v2e are matched with the results of FDM.

When FAOES database evolved, dataset was captured in FAOES_v2a. The sample dataset is shown in Figure 31. Following are some sample queries and its results produced in FDM and Web-FDM project.

ID	LNAME	FNAME	PRODUCTS	ORDERS	SALES	POSITION
1000	Wyatt	Stefan	3525	10	3543500	1
1001	Wright	Donald	1281	14	1833960	2
1002	Worral	Al	1255	13	1158350	3
1003	Wooton	Bruce	4075	25	4492620	4
1004	Widdes	Albert	3477	19	755515	4
1005	Wehland	William C.	8813	22	1172425	4
1006	Thomas	Peter	4190	13	209500	5
1008	Stone	James F.	2131	12	2867955	6
1010	Stansbury	Thomas	1143	12	1520840	8
1011	Stansbury	Steward	5294	36	7207330	9
1012	Somers	Bill	1914	13	125185	2
1013	Simmins	Steven	14	14	19835	3
1014	Ripkin	Jan	3106	32	3308585	4
1015	Reed	Donna	1835	8	2037325	4

Figure 51: Sample FAOES Data

Decisions used in the testing purpose are the followings

Fuzzy Attribute	Fuzzy Category	Decision

Products	Poor	Fire
Products	Below Average	Give Warning
Products	Average	Give Raise
Products	Above Average	Give 2% Gift
Products	Excellent	Give 10% Gift
Orders	Poor	Fire
Orders	Below Average	Give Warning
Orders	Average	Give Warning
Orders	Above Average	Give 2% Raise
Orders	Excellent	Give 10% Gift
Sales	Poor	Fire
Sales	Below Average	Give 2% Raise
Sales	Average	Give 10% Raise
Sales	Above Average	Give 15% Raise
Sales	Excellent	Give Raise and Gift

Query 1: In the decision table, how many employees are in each category? (FAOES_v2e)

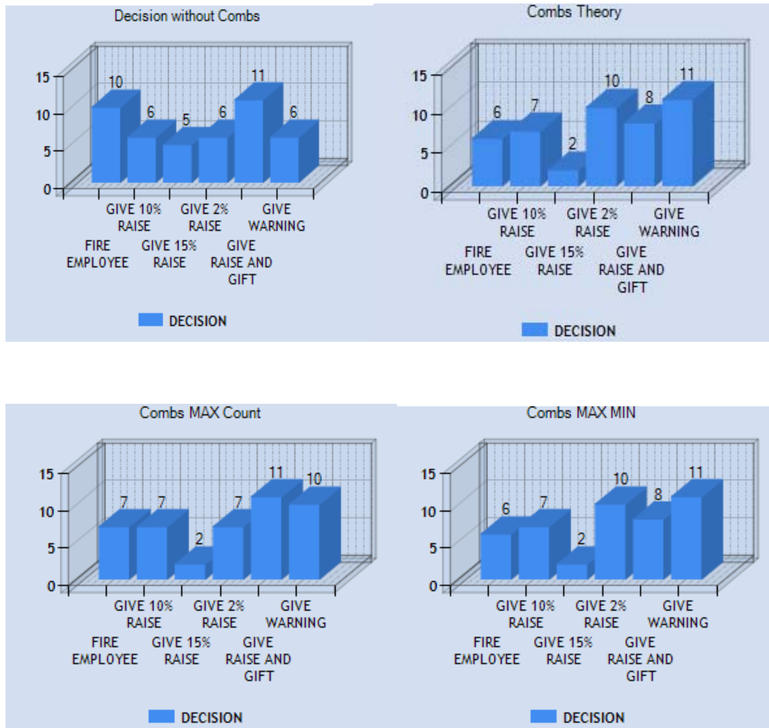


Figure 52: Graphical Analysis of 4 rule sets in Web-FDM 2.0

Detail Data#No Combs

DEC_TRI	COUNT
FIRE EMPLOYEE	10
GIVE 10% RAISE	6
GIVE 15% RAISE	5
GIVE 2% RAISE	6
GIVE RAISE AND GIFT	11
GIVE WARNING	6

Detail Data#Combs

DEC_TRI	COUNT
FIRE EMPLOYEE	6
GIVE 10% RAISE	7
GIVE 15% RAISE	2
GIVE 2% RAISE	10
GIVE RAISE AND GIFT	8
GIVE WARNING	11

Detail Data#Combs Max

DEC_TRI	COUNT
FIRE EMPLOYEE	7
GIVE 10% RAISE	7
GIVE 15% RAISE	2
GIVE 2% RAISE	7
GIVE RAISE AND GIFT	11
GIVE WARNING	10

Detail Data#Combs Min

DEC_TRI	COUNT
FIRE EMPLOYEE	6
GIVE 10% RAISE	7
GIVE 15% RAISE	2
GIVE 2% RAISE	10
GIVE RAISE AND GIFT	8
GIVE WARNING	11

Figure 53: Tabular analysis of 4 rule approaches in Web-FDM 2.0

From the result tables it is shown that decision made without combs inference and three methods for applying combs inference method are somewhat very similar. In fact with combs

Inference method, the decisions are more distributed among the output subsets rather than creating clusters in one single decision area.

10.2 Data Prediction and decision with FAOES_2E

The Data prediction process takes a partial dataset as an input and provides a prediction for full dataset over a period of time. Also in this step the decision after that certain time period is predicted with data mining.

For FAOES_2E in the FDB database, all the employees are evaluated for a 36 months period. For the testing purpose a new employee is created with a partial dataset of 9 months (Refer Appendix N2 for partial dataset).

10.2.1. Select Fuzzy Parameters: For FAOES_2E , the fuzzy parameters are the same as the fuzzy attributes.

a. Orders
b. Products
c. Sales

10.2.2. Calculate Fuzzy Attribute Weight:

For FAOES_2E, the fuzzy attributes are assigned the weight as following

a. Sales: 40%
b. Orders: 40%
c. Products: 20%

For Products the partial dataset for 9 month is the following,

Set	FAID	rate	ATTRIBUTEWEIGHT	FUZZYCATCONSTANT	FUZZYATTRWEIGHT	PROGRES
1	140	164.9	0.4	5.9	389.164	389.16
1	141	164.9	0.4	0.15	9.894	399.05
1	142	164.9	0.2	1	32.98	432.03
2	140	164.9	0.4	0.15	9.894	441.93
2	141	164.9	0.4	1	65.96	507.89
2	142	164.9	0.2	3	98.94	606.83

…

…

Table: Partial dataset for Fuzzy Attribute: Products

10.2.3. Calculate Maximum Category Constant, Minimum Category Constant, Attribute Rate:

Here, the Maximum value of Number of products in the worktable is= 35504

Minimum value of Number of Products is = 949

Average value of Number of Products is= 5935

So, Maximum Category Constant= 35504/949 = 5.98

Minimum Category Constant= 5935/949 = 0.159

Total Number of Months= 36

Attribute rate for each dataset = 5935/36 = 164.86

The Maximum Category Constant, Minimum Category Constant, Attribute Rate for other fuzzy attributes, sales are orders are also calculated.

For Sales,

Maximum Category Constant= 5.7

Minimum category Constant= 0.06

Attribute Rate= 161674.5

For Orders,

Maximum Category Constant= 3.94

Minimum Category Constant= 0.38

Attribute Rate= 0.86

10.2.4. Calculate future cumulative progress amount: After inserting the partial dataset for all 3 Fuzzy Parameters in the FDB Database, the following procedures are called:

CreateFutureProgressProcedure

This procedure will create another procedure for the worktable, which will eventually calculate the cumulative progress amount for FAOES_2E. Here we input the projection time period as 36 months. The application projects the number of net_orders, net_sales, net_products sold by the employee.

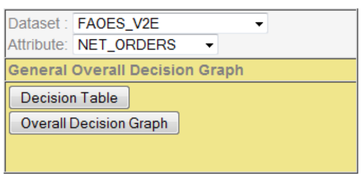
Using these projected values in the data mining process, the application automatically produces the decisions based on the Combs rule inference method.

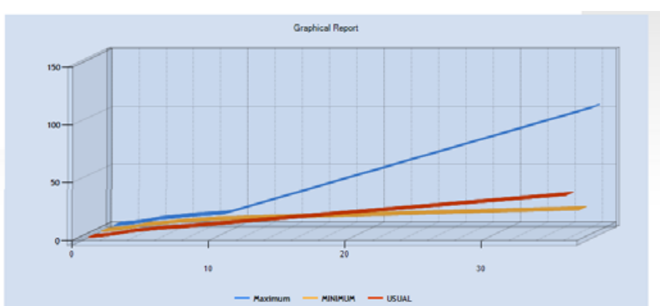
10.2.5. Analysis of the prediction:

In this phase, the application analyzes the predicted result set graphically. The graph has number of months as X-Axis and Cumulative Progress Amount as Y-Axis.

Three lines are plotted based on the following progression:

a. Maximum: The prediction based on the maximum Fuzzy Category Constant for rest of the time period after 9 months. This means, the employee is following the best possible route for the rest of the time period.
b. Minimum: The prediction based on the Minimum Category Constant.
c. Usual: The prediction based on the employee average performance for the 9 month period





Decision

EFFORT	DEC_TRI	DEC_TRP
AVERAGE	GIVE RAISE AND GIFT	GIVE RAISE AND GIFT
MAXIMUM	GIVE RAISE AND GIFT	GIVE RAISE AND GIFT
MINIMUM	GIVE WARNING	GIVE WARNING

Predicted Value

TYPE	ATTR	PROGRESS
AVG	NET_ORDERS	37.79
MAX	NET_ORDERS	102.86
MIN	NET_ORDERS	19.97

Figure 54: Analysis of data prediction of FAOES_2E

From the analysis, it is shown that, if the employee performs with a maximum rate or an average rate, for rest of the 36 months, he will be given raise and gift. In these 2 cases, his net_orders will be 102 and 37 respectively. If he performs in the worst path, he will be warned after 36 months and his net_orders will be 19.

CHAPTER 11

CONCLUSION

In this research, we have successfully implemented the analysis of a large amount of data using fuzzy logic with data mining concept.

In addition to the conventional rule approach applied in Web-FDM 1.0, 3 different approaches of Combs Method of Rapid inference method are implemented in Web-FDM 2.0 which is tested and validated with the previous saved data in FDB Database.

Based on the input database, rule set, a user can get an expert decision; also can compare all the decisions with or without Combs Method, and can chose the appropriate one.

The KRT Database is used as a basis for the data prediction and mining approach. This research successfully combined the data prediction algorithm for partial dataset with data mining, and can generate a decision in a future time period. This will help the user to assist the population to change their behavior to achieve a certain result.

CHAPTER 12

FUTURE RESEARCH

The following improvements can be made in the future research of this project:

- Web-FDM can be extended to calculate the projection time of any partial dataset from already available data in the worktable. Currently it is an input from the user.
- The weight of fuzzy parameters in the data predictor is assumed or by default set by the system. The application can find the appropriate weight of each fuzzy parameter depending how they contribute in the decision making process.
- Improve the input process system of Web-FDM. Currently the system assumes user knows SQL Language. The input data can be taken from an excel sheet.
- Printable report generating system can be added in Web-FDM.
- Web-FDM treats all fuzzy attributes with same set of fuzzy categories. The system should support multiple fuzzy categories for multiple fuzzy attribute.

REFERENCES

Cyrus Azarbod (2010), “Data Warehousing and Data Mining”, Lecture Notes. Retrieved from the website: http://mavdisk.mnsu.edu/cyrus123/444/444-lectures.htm on February 20, 2010.

Azarbod, R (2011). Determining a Patient Recovery from a Total Knee Replacement Using Fuzzy Logic and Active Databases. Thesis. MNSU – Mankato

Bankar, A. (2010): Web Based Fuzzy Data Mining and Visualization. Thesis. MNSU – Mankato. 2010

Alachisoft. (2008).http://www.alachisoft.com/tdev/index.html (accessed January 14, 2009).

Azarbod C., Thompson C. (2007). A Fuzzy Active Relational Database for Employee Performance Evaluation. SEDE - 2007 International Conference on Software Engineering and Data Engineering, Las Vegas, Nevada, 2007.

Azarbod, C. (2005). Fuzzy active database research.

Azarbod, C., Hamed, S., & Ali, J. (2006). An Automated fuzzy active database for employee performance evaluation using oracle. 4th ACS/IEEE International Conference on Computer System Applications, Dubai/Sharjah, UAE.

Bai, Y., Zhuang, H., & Wang, D. (2006). Advanced fuzzy logic technologies in industrial applications. Springer Verlag.

Bankar, A. A. (2008). Fuzzy active order entry system-2. Informally published manuscript, Department of Computer Science, Minnesota State University, Mankato, Minnesota, USA. Retrieved from http://mavdisk.mnsu.edu/cyrus123/FADB/FAOES2/default.htm

Calvert, C. (2009, April 08). Publishing an asp.net web application in IIS. Retrieved from http://blogs.msdn.com/charlie/archive/2009/04/08/publishing-an-asp-net-web-appli9ation-in-iis.aspx

Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.

Zadeh, L.A., "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 1, pp. 28-44, Jan. 1973.

Bonesmart (2009), Internet Society of Orthopedic Surgery and Trauma. "How long does it take to recover from total knee replacement surgery?". 29 June 2009. Bonesmart.org. http://www.orthogate.org/articles/.html

DerVaanesian, Saro (2007); Automatic Extension of Relational Database to Incorporate Fuzzy Logic. APP. American University of Armenia. 2007

Giori, Nicholas J (2001). "Measurement of Perioperative Flexion-Extension Mechanics of the Knee Joint". Mar. 8 2001. The Journal of Arthroplasty. Vol 16. No. 7. 2001. www.sciencedirect.com.

Haynes, Richard A (2009). "Total Knee Replacements - How Long Does it Take to Heal?". Orthopedic Fitness and Rehabilitation Products and Services. January 10, 2009. http://ezinearticles.com/?Total-Knee-Replacements---How-Long-Does-it-Take-to-Heal?&id=1858742.

Annie Hovian (2008), “Extending Fuzzy Active Relational Database to Automatically Incorporate Data Clustering Derived from Statistical Analysis”, Thesis, American University of Armenia, Yerevan, December 2008.

Stephen M. Howell, MD (2009); Stephanie L. Rogers, MPT. "Method for Quantifying Patient Expectations and Early Recovery After Total Knee Arthroplasty". December 2009. OrthoSuperSite. http://www.orthosupersite.com/view.aspx?rid=50758.

Dr. H. D. Huddleston (2005). "Arthritis of the Knee Joint". 2005. The Hip and Knee Institute. http://www.hipsandknees.com/knee/kneeimplants.htm.

A.K. JAIN, M.N. MURTY, AND P.J. FLYNN (2005). “Data Clustering: A Review”, ACM Computing Surveys, Vol. 31, No. 3, September 1999.

Retrieved from the website:

http://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf on February 21, 2010.

Steven D. Kaehler (2010), “Fuzzy Logic- An Introduction”, Newsletter of Seattle Robotics Society, March 1998 Retrieved from the website: http://www.seattlerobotics.org/encoder/mar98/fuz/fl_part1.html on February 20,2010

Sanjay Kaparalmy (1998), “Data Warehousing and Loading”, Alternate Plan Paper, Minnesota State University, Mankato, December 1998.

Mayo Clinic (2010). "Knee Replacement". 15 May 2010. Mayo Foundation for Medical Education and Research. http://www.mayoclinic.com/health/knee-replacement/MY00091

Mommersteeg, T.J.A (1995). "Characterization of the Mechanical Behavior of Human Knee Ligaments: A Numerical-Experimental Approach". 8 March 1995. Journal of Biomechanics, Vol 29. No. 2. pp 151-160. www.sciencedirect.com.

"X-Plain Knee Replacement-Physical Therapy Reference Summary". 2004. National Library of Medicine - National Institute of Health. http://www.nlm.nih.gov/medlineplus/tutorials/kneereplacementphysicaltherapy/pt049101.pdf

Decision Support Systems. July 2006. Open Clinical. http://www.openclinical.org/dss.html

Riener, Robert (1996). Biomechanical Model of the Human Knee Evaluated by Neuromuscular Stimulation. 1996. Journal of Biomechanics. Vol 29. No 9. pp 1157-1167. www.sciencedirect.com

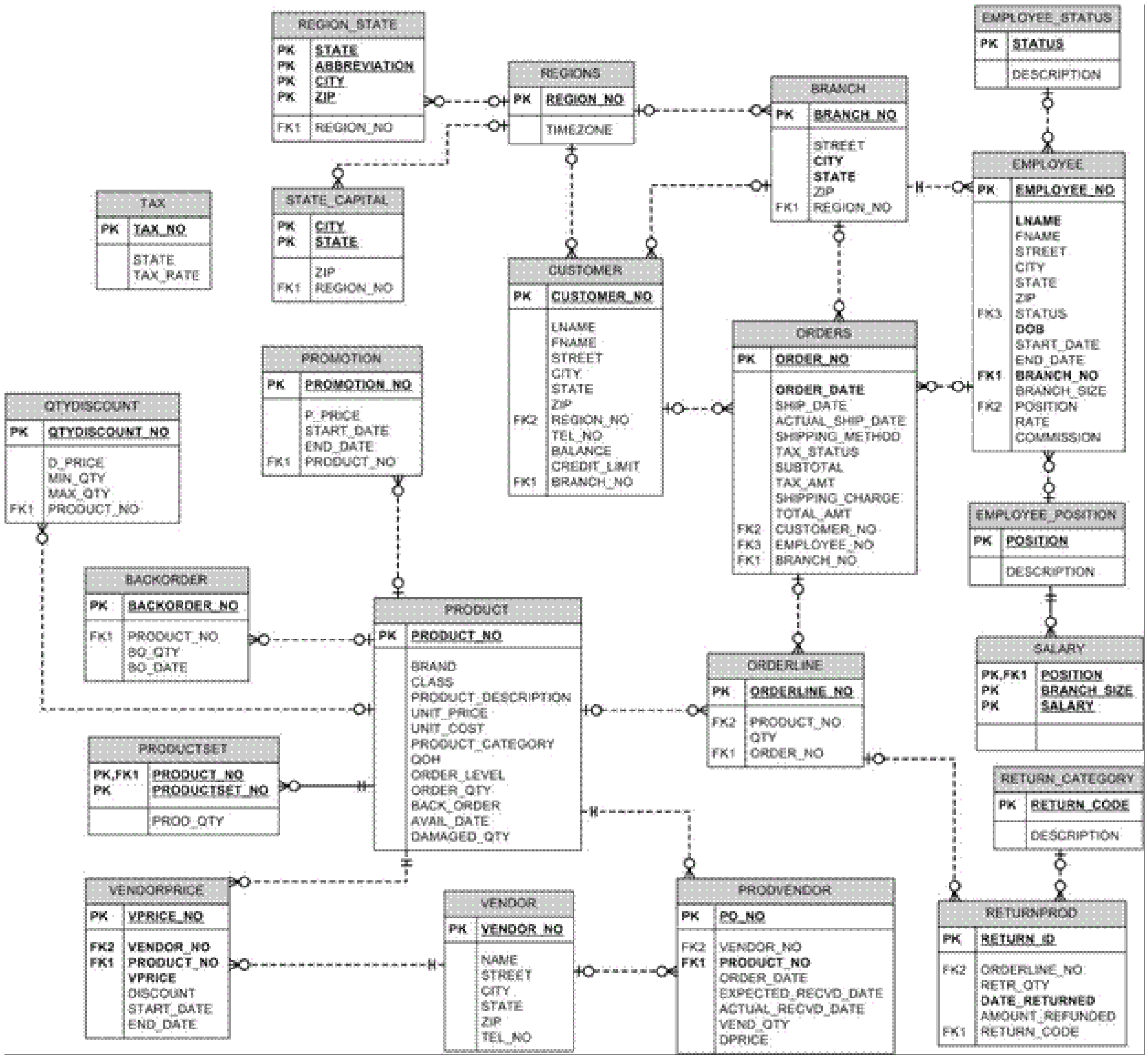
Hanan Sallam (1998), “Fuzzy Logic: Theory and Application”, Alternate Plan Paper, Minnesota State University, Mankato, December 1998.

Semerci, Cetin (2004). Department of Educational Services, Faculty Education. April 2004. Firat University, 23119 Elazig-Turkey. http://www.tojet.net/articles/329.pdf

Snedecor, George W. and Cochran, William G. (1989), “Statistical Methods”, Eighth Edition, Iowa State University Press. 1989

Yanala, Haritha (2010). Fuzzy Data Mining for Evaluating Employee Performance. APP MNSU – Mankato. 2010

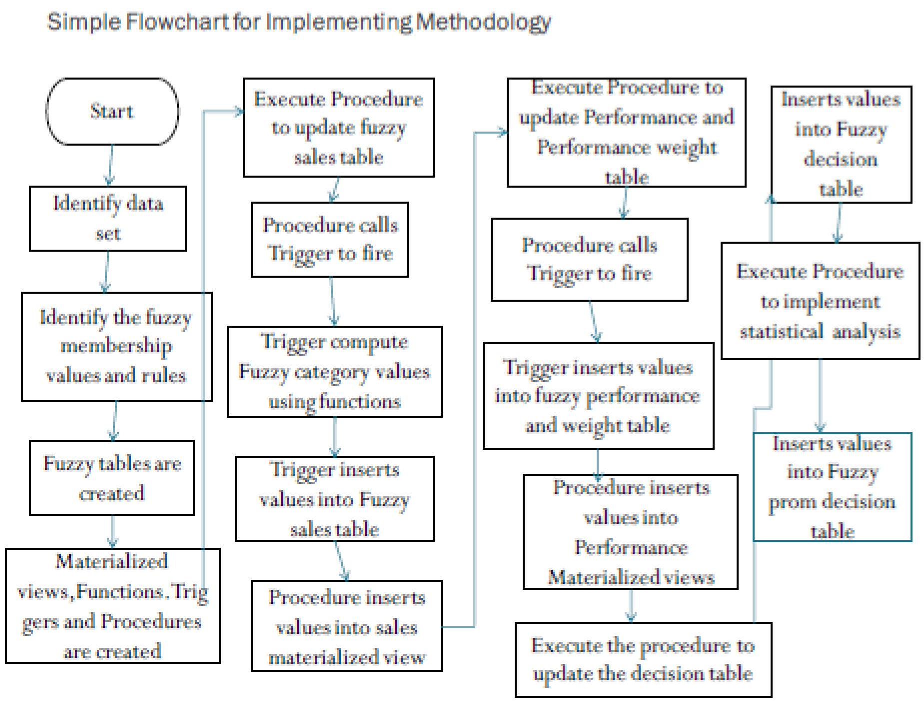
Appendix A1: OES Data Model



Appendix B1: ARDIF-FAOES query comparing both their results

ID	EMPLOYEE_NO	LNAME	GRAPH	FUZZY_DECISION
1002	1002	Worral	TRI	GIVE 2% RAISE
1003	1003	Wooton	TRI	GIVE 10% RAISE
1004	1004	Widdes	TRI	GIVE 2% RAISE
1005	1005	Wehland	TRI	GIVE 10% RAISE
1006	1006	Thomas	TRI	FIRE EMPLOYEE
1008	1008	Stone	TRI	FIRE EMPLOYEE
1013	1013	Simmins	TRI	FIRE EMPLOYEE
1014	1014	Ripkin	TRI	GIVE 2% RAISE
1015	1015	Reed	TRI	GIVE WARNING
1016	1016	Prouty	TRI	FIRE EMPLOYEE
1022	1022	Nabb	TRI	GIVE RAISE AND GIFT
1023	1023	Murthy	TRI	GIVE WARNING
1024	1024	Mudd	TRI	GIVE 2% RAISE
1028	1028	Mayfield	TRI	GIVE 10% RAISE
1029	1029	Martin	TRI	GIVE 15% RAISE
1030	1030	Keting	TRI	FIRE EMPLOYEE
1032	1032	Johnston	TRI	GIVE 15% RAISE
1033	1033	Johnson	TRI	GIVE RAISE AND GIFT
1034	1034	Jenkins	TRI	FIRE EMPLOYEE
1036	1036	Heisler	TRI	GIVE RAISE AND GIFT
1038	1038	Hanzdo	TRI	GIVE RAISE AND GIFT
1039	1039	Halle	TRI	GIVE RAISE AND GIFT
1042	1042	Farmer	TRI	GIVE RAISE AND GIFT
1045	1045	Doering	TRI	GIVE RAISE AND GIFT
1046	1046	Doering	TRI	GIVE 2% RAISE
1047	1047	Constable	TRI	GIVE RAISE AND GIFT
1053	1053	Pregmon	TRI	GIVE 10% RAISE

Appendix B2: FAOES detailed flowchart to implement FAOES



Appendix B3: FAOES Triangle and Trapezoid Function

<u>Triangle Function</u>

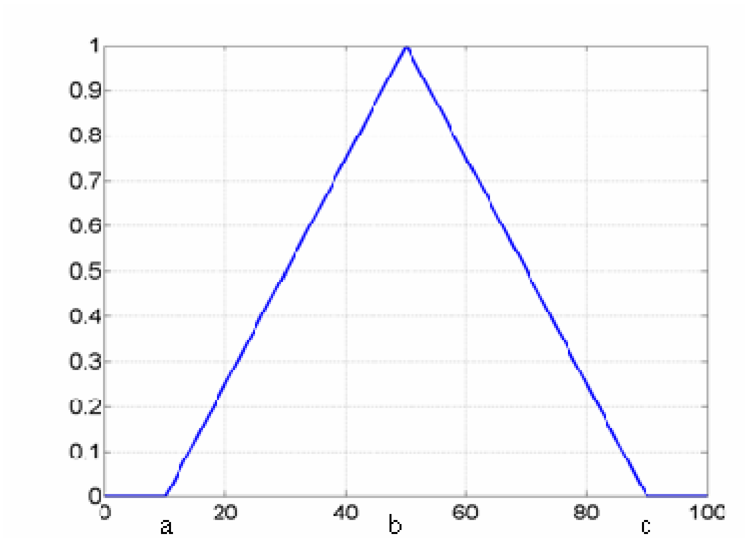
The triangular function is described as:

0 when x <= a

(x-a)/(b-a) when x is between a and b

(c-x)/(c-b) when x is between b and c

0 when x >= c



Triangle Membership function graph

x = sales amount

a = lowest value in membership range

b = membership value of 1

c = greatest value in membership range

Attribute Name	Fuzzy Categories	Fuzzy Range	Membership Value (a,b,c)
Number Orders	Poor	[0,9]	(0,0,9)
	Below Average	[8,20]	(8,15,20)
	Average	[18,30]	(18,25,30)
	Above Average	[28,39]	(28,35,40)
	Excellent	[39,200]	(39,45,1000000)
Number Products	Poor	[0,40]	(0,0,40)
	Below Average	[38,60]	(38,50,60)
	Average	[58,80]	(58,70,80)
	Above Average	[78,100]	(78,90,100)
	Excellent	[99,1000]	-991,001,000,000
Sales (Mega$)	Poor	[0,2.5]	(0,0,2.5)
	Below Average	[2.4,4.5]	(2.4,3.5,4.5)
	Average	[4.4,6.5]	(4.4,5.5,6.5)
	Above Average	[6.3,8.5]	(6.3,7.4,8.5)
	Excellent	[8.4,1000]	(8.4,8.5, 1000000)

<u>Trapezoid Function</u>

The trapezoidal function is described as

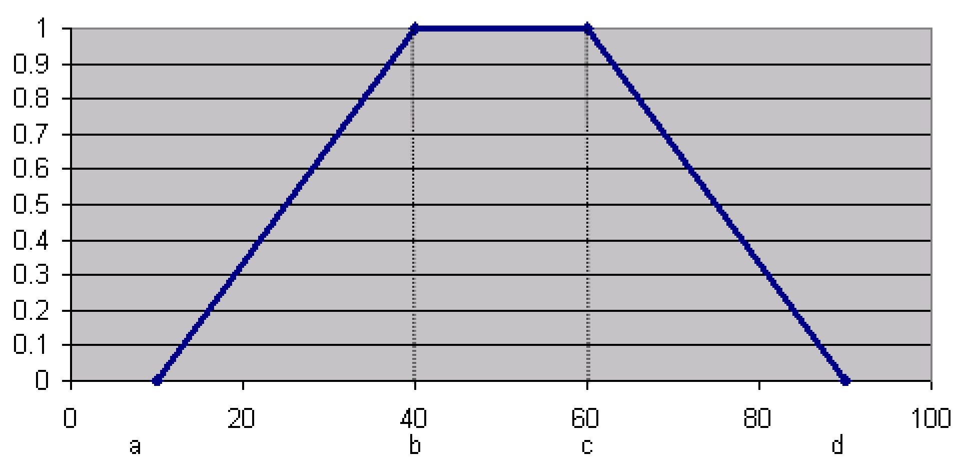
a. 0 when x <= a

b. 1 when x is between b and c (b < x < c)

c. (x-a)/(b-a) when x is between a and b (a < x <= b)

d. (d-x)/(d-c) when x is between c and d (c <= x <d)

e. 0 when x >= d



The category ranges were kept consistent for both functions (Triangle and Tarpezoid)

Attribute Name	Fuzzy Names	Fuzzy Range	Membership Value (a,b,c,d)
Number Orders	Poor	[0,9]	(0,0,5,9)
	Below Average	[8,20]	(8,12,16,20
	Average	[18,30]	(18,22,26,30)
	Above Average	[27,40]	(27,31,36,40)
	Excellent	[39,200]	(39,43+)
Number Products (Thousands)	Poor	[0,40]	(0,0,36,40)
	Below Average	[36,60]	(36,40,56,60)
	Average	[58,80]	(58,62,76,80)
	Above Average	[78,100]	(78,82,96,100)
	Excellent	[99,1000]	(99, 103, 103+)
Sales (Millions)	Poor	[0,2.5]	(0,0,2.1,2.5)
	Below Average	[2.1,4.5]	(2.1,2.5,4.1,4.5)
	Average	[3.7,6.5]	(3.7,4.1,6.1,6.5)
	Above Average	[6.3,8.5]	(6.3,6.7,8.1,8.5)
	Excellent	[8.4,1000]	(8.4, 8.8+)

Appendix B4: FAOES Fuzzy Components

List of procedures used in this project are

- P_update_Emp_fuzzy_sales_mv
- P_update_Emp_fuzzy_orders_mv
- P_update_Emp_fuzzy_products_mv
- P_update_Emp_fuzzy_performance_mv
- P_generate_fuzzy_decision
- P_update_Emp_fuzzy_sales_mv_z
- P_update_Emp_fuzzy_orders_mv_z
- P_update_Emp_fuzzy_products_mv_z
- P_update_Emp_fuzzy_performance_mv_z
- error_cluster
- normal_cluster
- promotion_category
- p_generate_fuzzy_prom_decision

Procedures Used to Populate the Orders Table:

- update_subtotal : populates the subtotal field
- update_shipping_charge : populates the shipping_charge field
- update_tax : populates the tax field
- update_total_amt : populates the total_amt field

Triggers:

- t_update_Emp_fuzzy_sales_mv
- t_update_Emp_fuzzy_orders_mv
- t_update_Emp_fuzzy_products_mv
- t_update_Emp_fuzzy_performance_mv
- t_update_Emp_fuzzy_sales_mv_z
- t_update_Emp_fuzzy_orders_mv_z
- t_update_Emp_fuzzy_products_mv_z
- t_update_Emp_fuzzy_performance_mv_z

User defined functions:

Sales Functions – calculate membership degree values, used by t_emp_fuzzy_sales_mv

- update_sales_poor
- update_sales_below_average
- update_sales_average
- update_sales_above_average
- update_sales_excellent

Orders Functions – calculate membership degree values, used by t_emp_fuzzy_orders_mv

- update_orders_poor
- update_orders_below_average

 update_orders_average

 update_orders_above_average

 update_orders_excellent

Number Products Functions – calculate membership degree values, used by

t_emp_fuzzy_products_mv

 update_products_poor

 update_products_below_average

 update_products_average

 update_products_above_average

 update_products_excellent

List of Materialized views used in this project

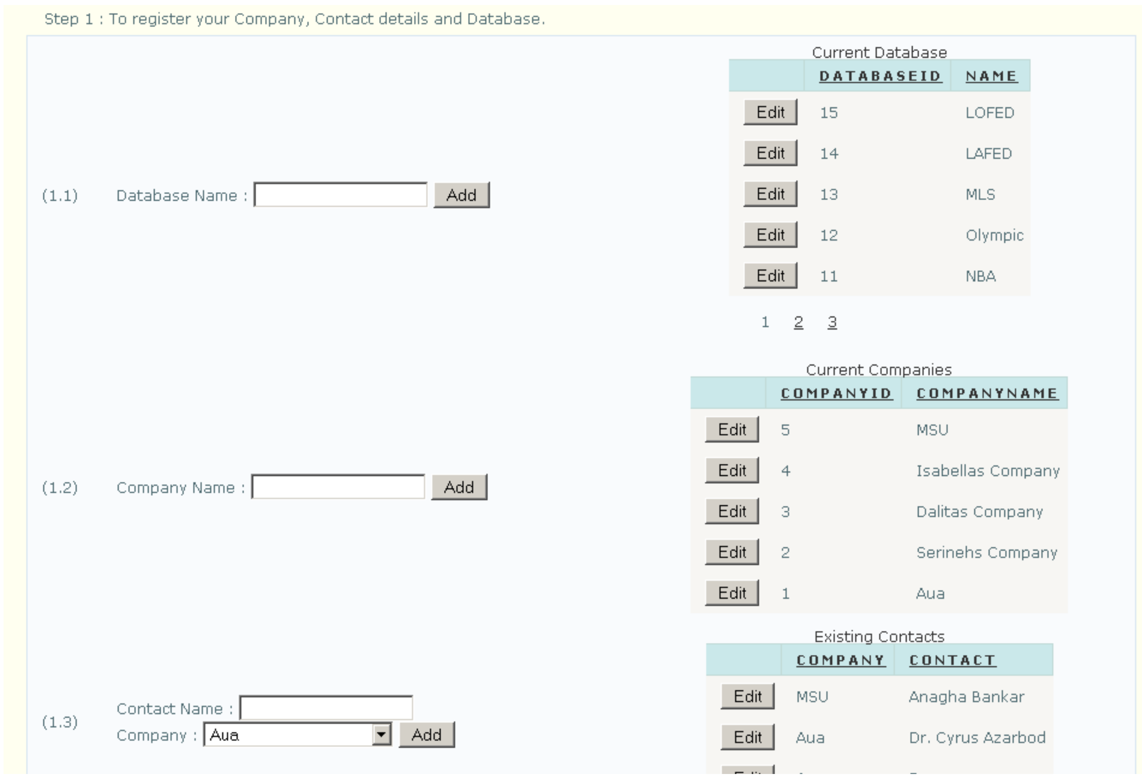
 Emp_fuzzy_sales_mv

 Emp_fuzzy_orders_mv

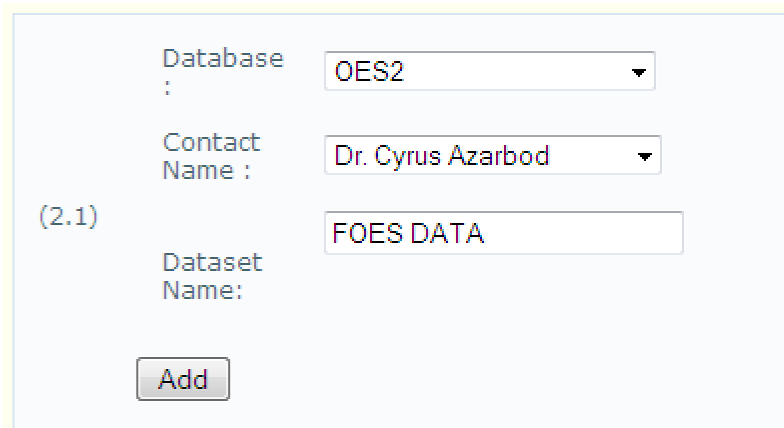
 Emp_fuzzy_products_mv

 Emp_fuzzy_performance_mv

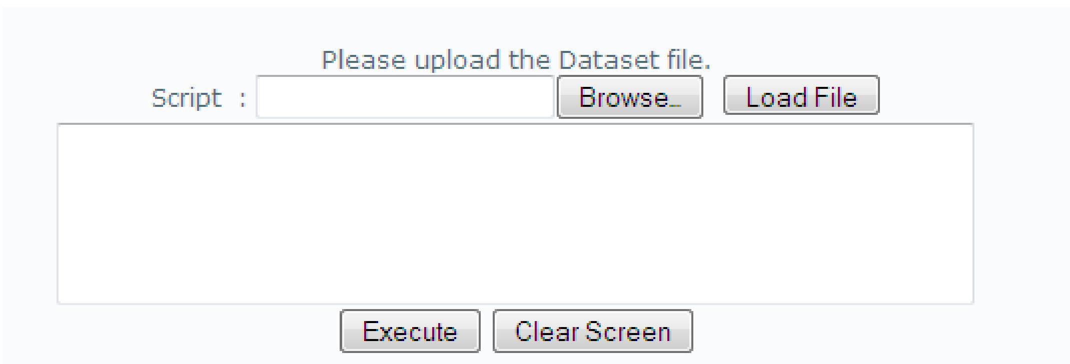
Appendix C1: Initial set-up – Step1 Add Contact name



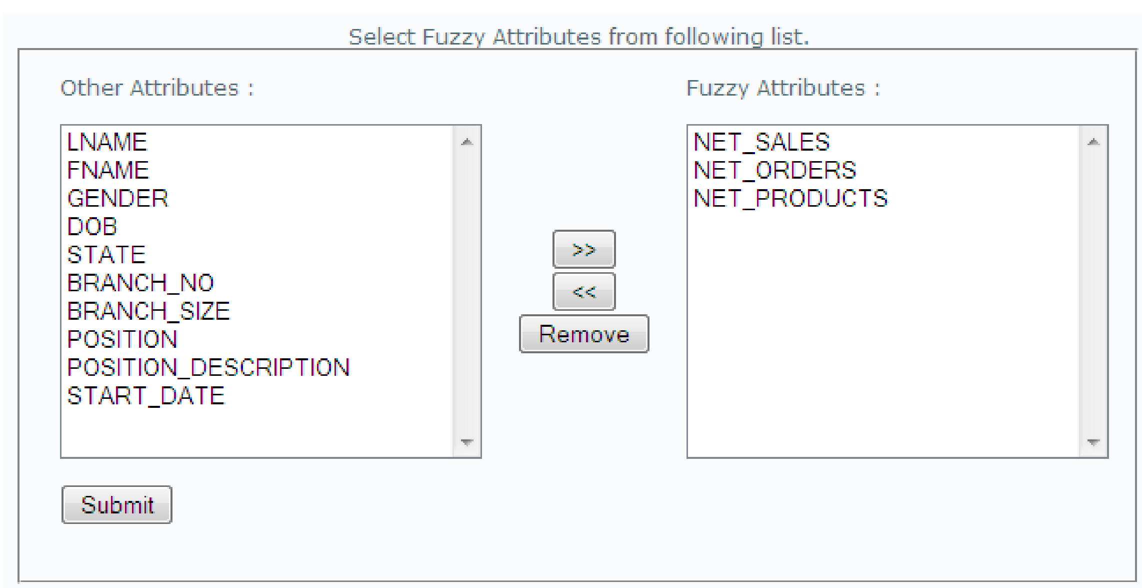
Appendix C2: Initial set-up – Step 2 Add Dataset Name



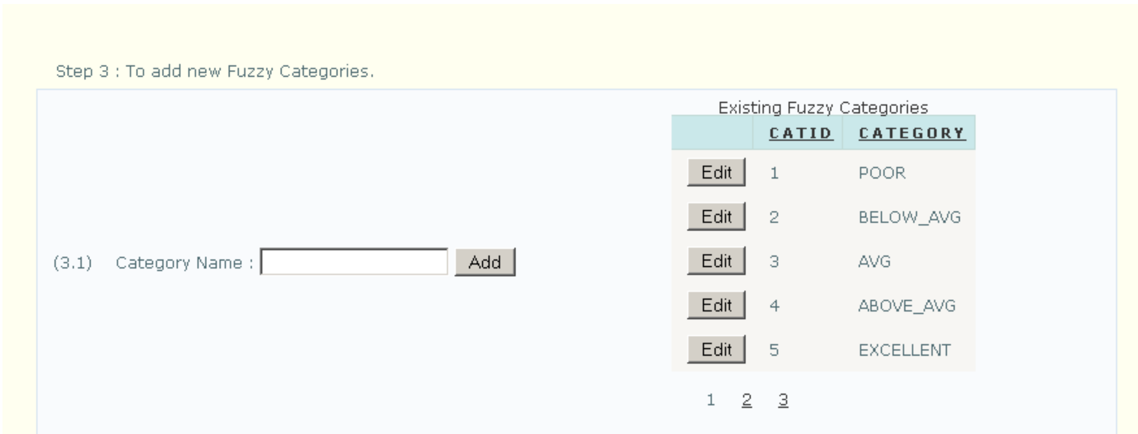
Initial set-up – Step 2 – Dataset entry



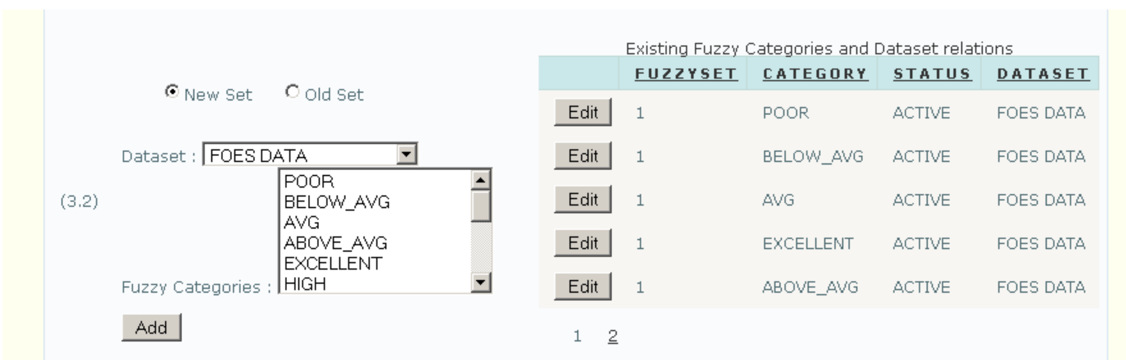
Initial set-up – Step 2 – Selection of Fuzzy and other attributes



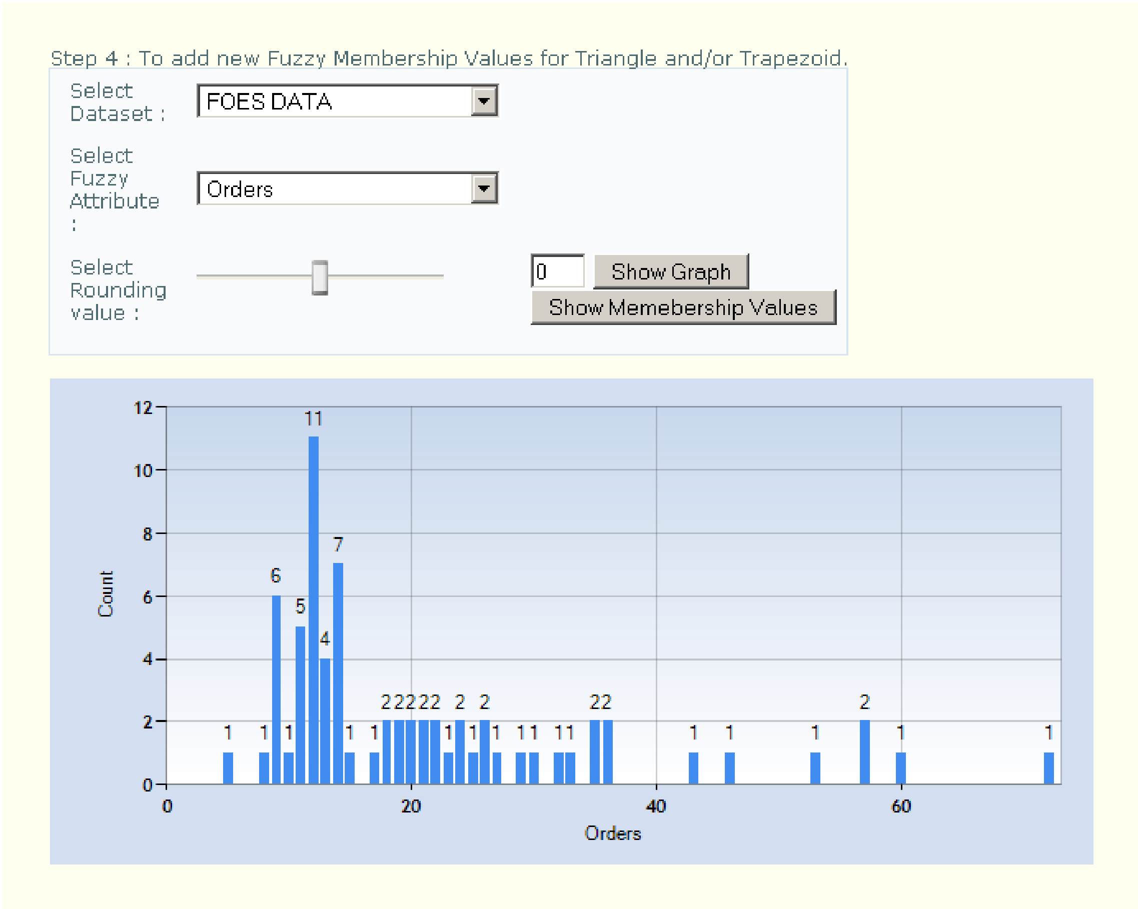
Appendix C3: Initial set-up – Step 3 Add Categories: Fuzzy categories are also called as fuzzy operators



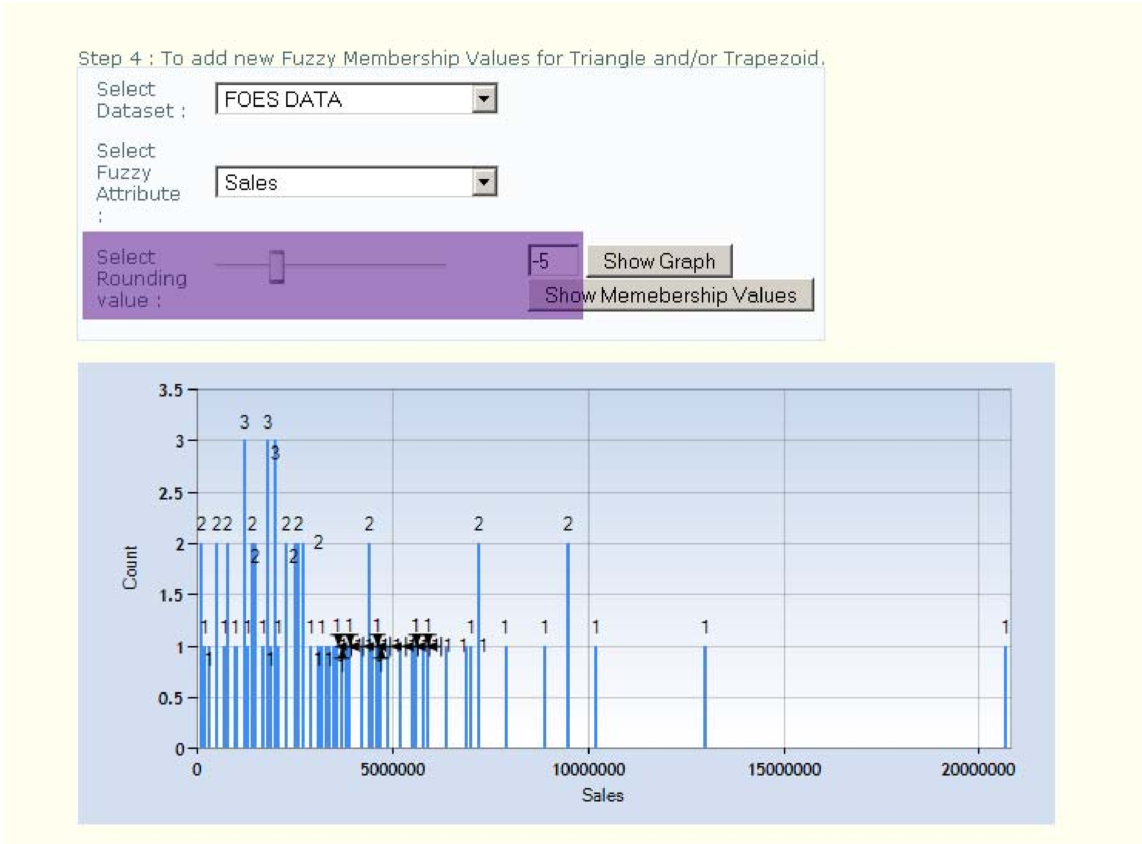
Initial set-up – Step3: Relate categories to the dataset.



Appendix C4: Initial set-up – Step 4 Selection of Dataset, Fuzzy Attribute and Rounding value



Initial set-up – Step 4: Change in graph by applying rounding function to input data



Initial set-up – Step 4: Auto-Generating Membership values

Membership Set - Graph : Triangle - Attribute :Sales

CATEGORY	START_POINT	MID1	MID2	END_POINT
POOR	0	600000	NULL	1200000
BELOW_AVG	1080000	1590000	NULL	2100000
AVG	1998000	2749000	NULL	3500000
ABOVE_AVG	3349800	4474900	NULL	5600000
EXCELLENT	5374980	13037490	NULL	20700000

Triangular Membership function values

Membership Set - Graph : Trapezoid - Attribute :Sales

CATEGORY	START_POINT	MID1	MID2	END_POINT
POOR	0	400000	800000	1200000
BELOW_AVG	1040000	1393333	1746667	2100000
AVG	1958666	2472444	2986222	3500000
ABOVE_AVG	3294488	4062993	4831496	5600000
EXCELLENT	5292598	10428399	15564200	20700000

Trapezoidal Membership function values

Appendix C5: Initial set-up – Step 5 Add decision Types

Step 5 : To add new Decision Types and Rules.

(5.1) Dataset : FOES DATA

Decision : (max 50 characters)

Add

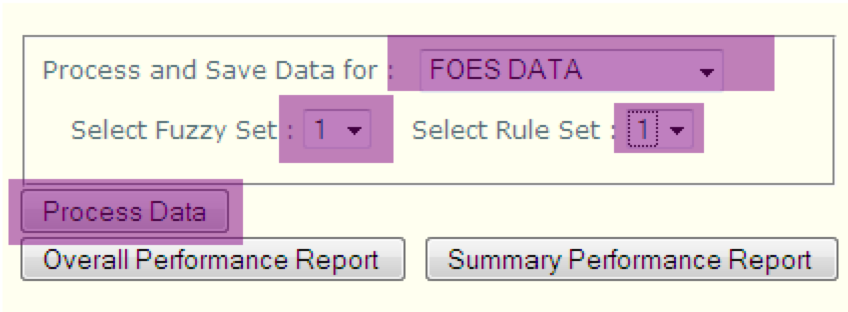
Existing Decision Types :

	DECID	DECISION
Edit	1	fire
Edit	2	give warning
Edit	3	do nothing
Edit	4	give raise
Edit	5	gift and raise
Edit	6	Give Premium

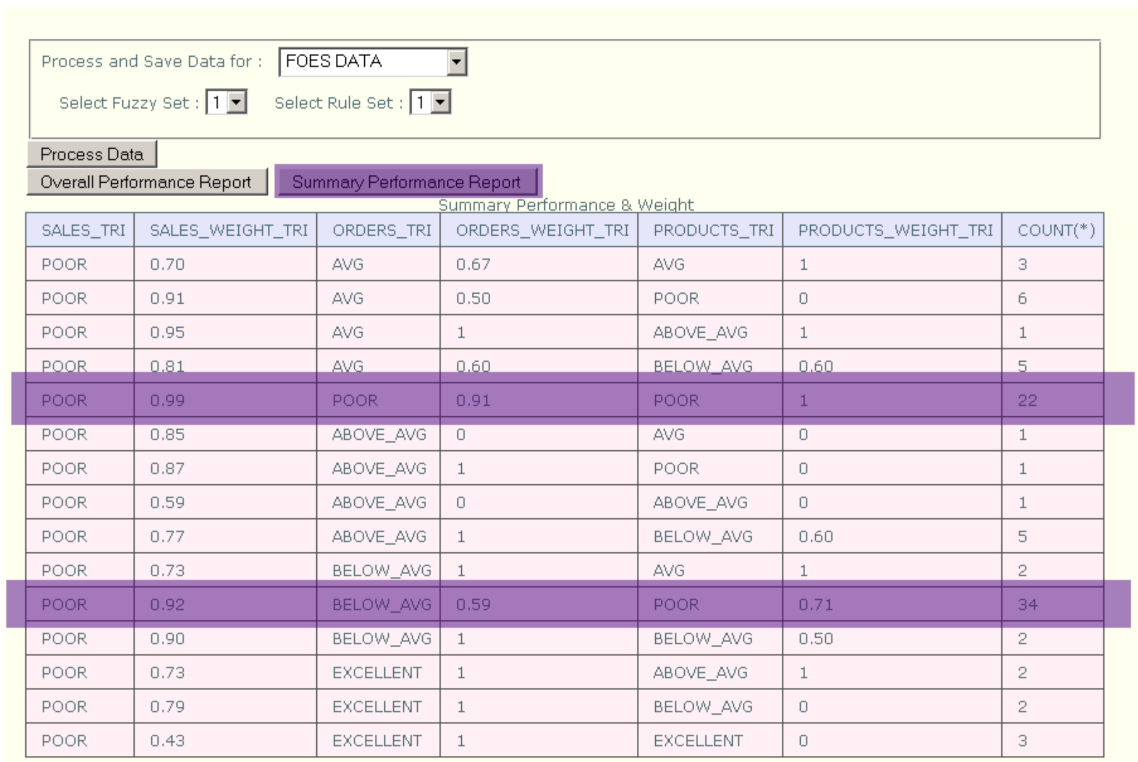
Initial set-up – Step 5: Create Decision Set

SELECT	RULE	SALES	ORDERS	PRODUCTS
☑	fire	1:POOR	1:POOR	1:POOR
☑	fire	1:POOR	1:POOR	2:BELOW_AVG
☑	fire	1:POOR	1:POOR	3:AVG
☑	give warning	1:POOR	1:POOR	4:ABOVE_AVG
☑	give warning	1:POOR	1:POOR	5:EXCELLENT
☑	fire	1:POOR	2:BELOW_AVG	1:POOR
☑	fire	1:POOR	2:BELOW_AVG	2:BELOW_AVG
☑	fire	1:POOR	2:BELOW_AVG	3:AVG

Appendix C6: Initial set-up – Step 6 Execute the system (Processing the dataset using fuzzy logic)



Initial set-up – Step 7: Summary Performance report (twenty two employees have poor performance in sales, orders and products and thirty four employees have poor, below average and poor performance in sales, orders and products respectively



SALES_TRI	SALES_WEIGHT_TRI	ORDERS_TRI	ORDERS_WEIGHT_TRI	PRODUCTS_TRI	PRODUCTS_WEIGHT_TRI	COUNT(*)
POOR	0.70	AVG	0.67	AVG	1	3
POOR	0.91	AVG	0.50	POOR	0	6
POOR	0.95	AVG	1	ABOVE_AVG	1	1
POOR	0.81	AVG	0.60	BELOW_AVG	0.60	5
POOR	0.99	POOR	0.91	POOR	1	22
POOR	0.85	ABOVE_AVG	0	AVG	0	1
POOR	0.87	ABOVE_AVG	1	POOR	0	1
POOR	0.59	ABOVE_AVG	0	ABOVE_AVG	0	1
POOR	0.77	ABOVE_AVG	1	BELOW_AVG	0.60	5
POOR	0.73	BELOW_AVG	1	AVG	1	2
POOR	0.92	BELOW_AVG	0.59	POOR	0.71	34
POOR	0.90	BELOW_AVG	1	BELOW_AVG	0.50	2
POOR	0.73	EXCELLENT	1	ABOVE_AVG	1	2
POOR	0.79	EXCELLENT	1	BELOW_AVG	0	2
POOR	0.43	EXCELLENT	1	EXCELLENT	0	3

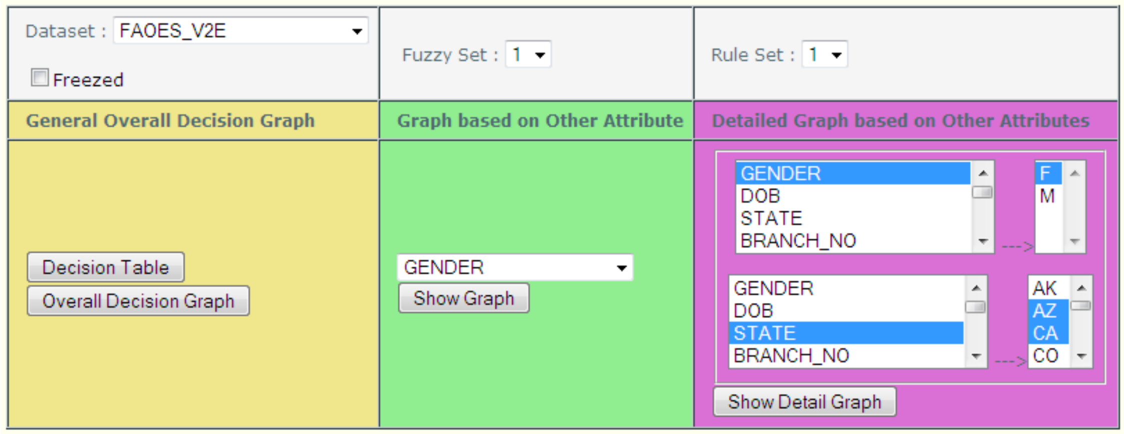
Initial set-up – Step7: Overall Performance report

Overall Performance Report | Summary Performance Report

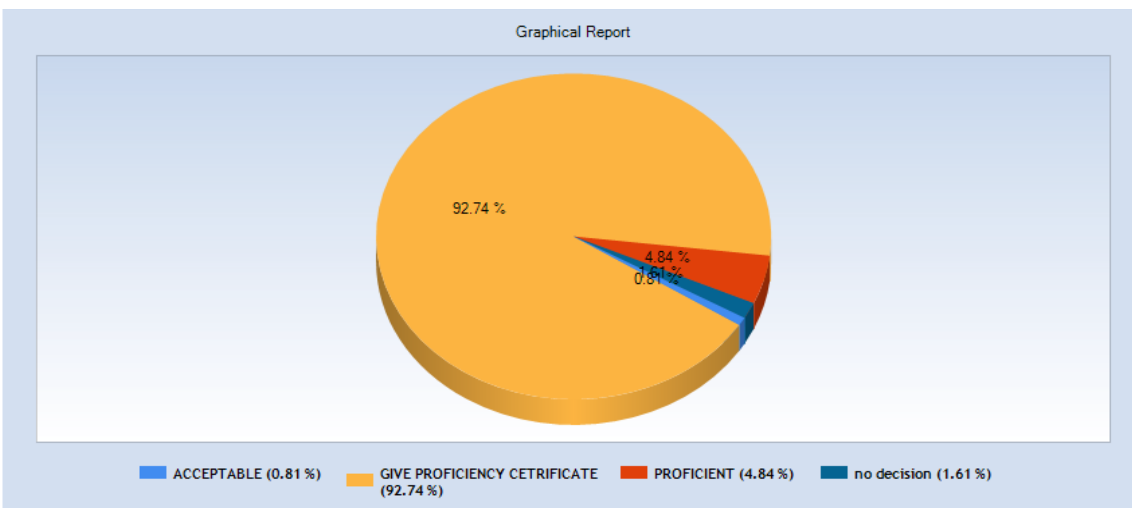
Overall Performance & Weight

ID	SALES_TRI	SALES_WEIGHT_TRI	ORDERS_TRI	ORDERS_WEIGHT_TRI	PRODUCTS_TRI	PRODUCTS_WEIGHT_TRI
1000	POOR	0.858260	BELOW_AVG	0	POOR	0
1001	POOR	0.926642	BELOW_AVG	1	POOR	1
1002	POOR	0.953666	BELOW_AVG	1	POOR	1
1003	POOR	0.820295	AVG	1	BELOW_AVG	1
1004	POOR	0.969779	AVG	0	POOR	0
1005	POOR	0.953103	AVG	1	ABOVE_AVG	1
1006	POOR	0.991620	BELOW_AVG	1	BELOW_AVG	1
1007	POOR	1	POOR	1	POOR	1
1008	POOR	0.885282	BELOW_AVG	1	POOR	0
1009	POOR	1	POOR	1	POOR	1
1010	POOR	0.939166	BELOW_AVG	1	POOR	1
1011	POOR	0.711707	ABOVE_AVG	1	BELOW_AVG	0
1012	POOR	0.994993	BELOW_AVG	1	POOR	1
1013	POOR	0.999207	BELOW_AVG	1	POOR	1
1014	POOR	0.867657	ABOVE_AVG	1	POOR	0
1015	POOR	0.918507	POOR	0	POOR	1

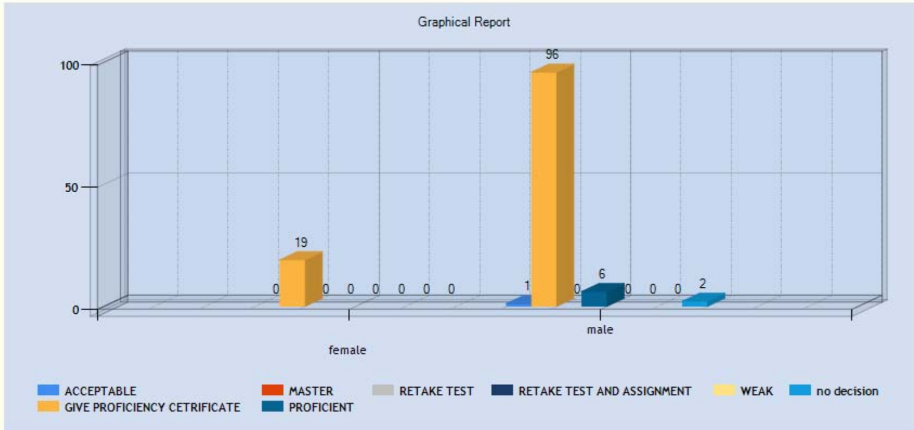
Appendix C7: Learning Phase: Analysis toolbox **(**the customized analysis toolbox helps to identify the optimal results (mark the freeze box to be used in mining)



Learning Phase type 1: Overall analysis for a selected fuzzy set and rule



Learning Phase type 2: Analysis is based on other attributes (only one attribute)

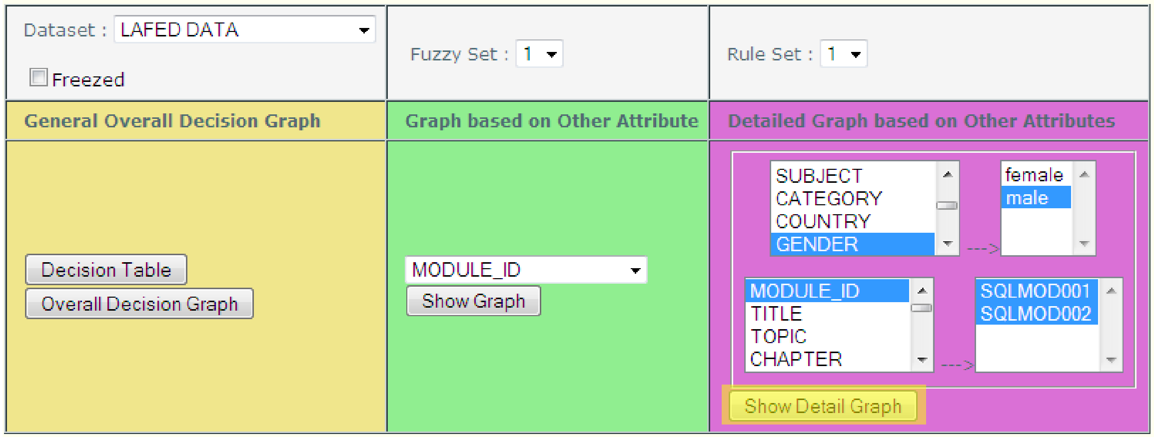


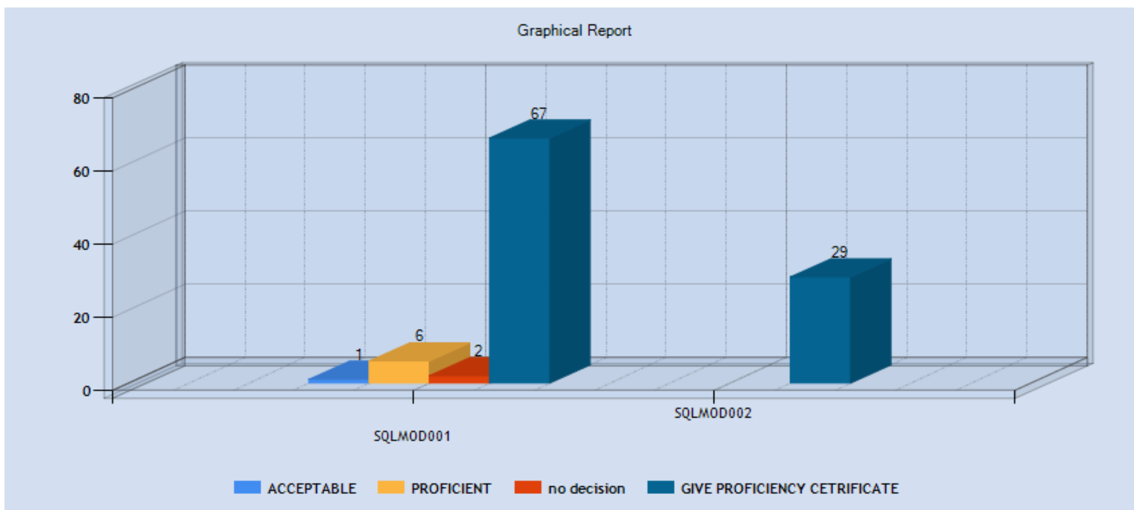
Legend : DECISION Show Point Values ☑ Clustered ☑ 3D ☑ Pie Chart ☐ TRIANGLE

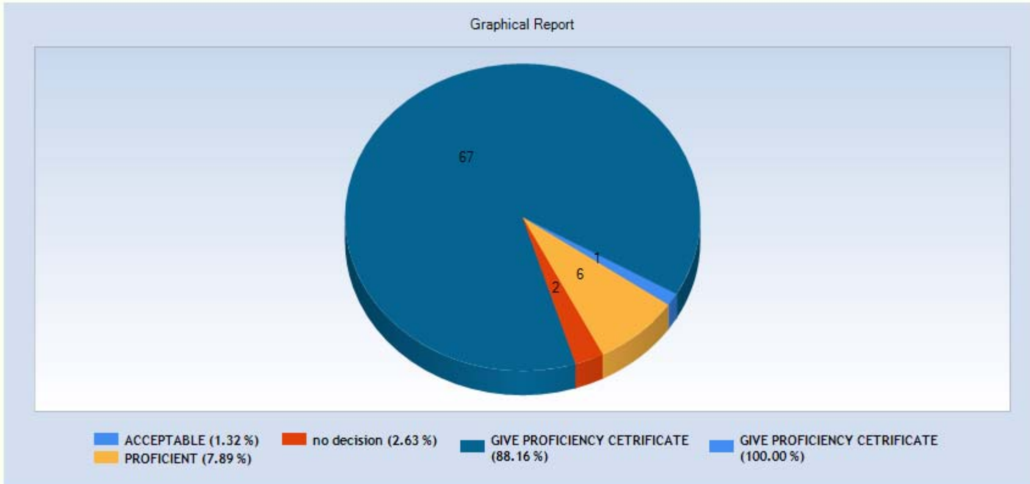
Detail Data#2

ATTRIBUTE	DECISION	COUNT
female	ACCEPTABLE	0
female	GIVE PROFICIENCY CETRIFICATE	19
female	MASTER	0

Appendix C8: Learning Phase type 3: Detail graphical analysis







Legend : ATTRIBUTE Show Point Values ☑ Clustered ☑ 3D ☑ Pie Chart ☑ TRIANGLE

Detail Data#2

DECISION	ATTRIBUTE	COUNT
ACCEPTABLE	SQLMOD001	1
PROFICIENT	SQLMOD001	6
no decision	SQLMOD001	2
GIVE PROFICIENCY CETRIFICATE	SQLMOD001	67
GIVE PROFICIENCY CETRIFICATE	SQLMOD002	29

ID	EMPLOYEE_NO	LNAME	GRAPH	FUZZY_DECISION
1002	1002	Worral	TRI	GIVE 2% RAISE
1003	1003	Wooton	TRI	GIVE 10% RAISE
1004	1004	Widdes	TRI	GIVE 2% RAISE
1005	1005	Wehland	TRI	GIVE 10% RAISE
1006	1006	Thomas	TRI	FIRE EMPLOYEE
1008	1008	Stone	TRI	FIRE EMPLOYEE
1013	1013	Simmins	TRI	FIRE EMPLOYEE
1014	1014	Ripkin	TRI	GIVE 2% RAISE
1015	1015	Reed	TRI	GIVE WARNING
1016	1016	Prouty	TRI	FIRE EMPLOYEE
. . .				

Appendix D1: Table Scripts and Insert Scripts

```
DROP TABLE PATIENT CASCADE CONSTRAINTS;
DROP TABLE EMPLOYEE CASCADE CONSTRAINTS;
DROP TABLE NUTRITION CASCADE CONSTRAINTS;
DROP TABLE PATIENTNUTRITION CASCADE CONSTRAINTS;
DROP TABLE INJURY CASCADE CONSTRAINTS;
DROP TABLE PATIENTINJURY CASCADE CONSTRAINTS;
DROP TABLE MEDICATION CASCADE CONSTRAINTS;
DROP TABLE PATIENTMEDICATION CASCADE CONSTRAINTS;
DROP TABLE EXERCISE CASCADE CONSTRAINTS;
DROP TABLE WEIGHT CASCADE CONSTRAINTS;
DROP TABLE WORKOUT CASCADE CONSTRAINTS;
DROP TABLE DISEASE CASCADE CONSTRAINTS;
DROP TABLE PATIENTDISEASE CASCADE CONSTRAINTS;
CREATE TABLE PATIENT(
PID VARCHAR2(8) NOT NULL,
FNAME VARCHAR2(30) NOT NULL ,
LNAME VARCHAR2(30) ,
DOB DATE NOT NULL ,
GENDER VARCHAR2(1) ,
HEIGHT NUMBER(5,2),
SMOKE VARCHAR(1),
jOB VARCHAR2(30),
INSURANCE VARCHAR(1),
CONSTRAINT PATIENT_PK PRIMARY KEY (PID));
CREATE TABLE EMPLOYEE(
EMPID VARCHAR2(8) NOT NULL,
FNAME VARCHAR2(30) NOT NULL ,
LNAME VARCHAR2(30) ,
GENDER VARCHAR2(1) ,
HEIGHT NUMBER(5,2),
DATE_STARTED DATE NOT NULL,
DOB DATE NOT NULL,
END_DATE DATE,
CONSTRAINT EMPLOYEE_PK PRIMARY KEY (EMPID));
CREATE TABLE NUTRITION(
NID VARCHAR2(8) not null,
NUTRITION_NAME VARCHAR(30) NOT NULL,
DESCRIPTION VARCHAR(100) ,
constraint NUTRITION_PK primary key (NID) ); 84
```

```
CREATE TABLE PATIENTNUTRITION(
PID VARCHAR2(8) NOT NULL ,
NID VARCHAR2(8) not null,
WEEK_NO NUMBER(2) NOT NULL,
AMOUNT NUMBER(5,2),
constraint PARIENT_NUTRITION_PK primary key (PID,NID,WEEK_NO),
CONSTRAINT PATIENT_NUTRITION_FK1 FOREIGN KEY (NID) REFERENCES NUTRITION(NID),
CONSTRAINT PATIENT_NUTRITION_FK2 FOREIGN KEY (PID) REFERENCES PATIENT(PID));
CREATE TABLE INJURY(
IID VARCHAR2(8) not null,
DESCRIPTION VARCHAR(100) ,
constraint INJURY_PK primary key (IID) );
CREATE TABLE PATIENTINJURY(
PID VARCHAR2(8) NOT NULL ,
IID VARCHAR2(8) not null,
SURGERY_DATE DATE NOT NULL,
INJURY_DATE DATE NOT NULL,
constraint PARIENT_INJURY_PK primary key (IID,PID,SURGERY_DATE) ,
CONSTRAINT PATIENT_INJURY_FK1 FOREIGN KEY (IID) REFERENCES INJURY(IID),
CONSTRAINT PATIENT_INUURY_FK2 FOREIGN KEY (PID) REFERENCES PATIENT(PID));
CREATE TABLE MEDICATION(
MID VARCHAR2(8) not null,
MEDICATION_NAME VARCHAR(40) NOT NULL,
DESCRIPTION VARCHAR(100) ,
constraint MEDICATION_PK primary key (MID) );
CREATE TABLE PATIENTMEDICATION(
PID VARCHAR2(8) NOT NULL ,
MID VARCHAR2(8) not null,
START_DATE DATE NOT NULL,
END_DATE DATE ,
constraint PATIENT_MEDICATION_PK primary key (PID,MID,START_DATE) ,
CONSTRAINT PATIENT_MEDICATION_FK1 FOREIGN KEY (MID) REFERENCES MEDICATION(MID),
CONSTRAINT PATIENT_MEDICATION_FK2 FOREIGN KEY (PID) REFERENCES PATIENT(PID));
CREATE TABLE EXERCISE(
EID VARCHAR2(8) not null, 85
```

```
DESCRIPTION VARCHAR(100) ,
constraint EXERCISE_PK primary key (EID) );
CREATE TABLE WEIGHT(
WEEK_NO NUMBER(3,0) not null,
PID VARCHAR2(8) NOT NULL,
AMOUNT NUMBER(5,2),
HOURS_SLEEP NUMBER(2,1),
constraint WEIGHT_PK primary key (WEEK_NO,PID),
CONSTRAINT WEIGHT_FK1 FOREIGN KEY (PID) REFERENCES PATIENT(PID));
CREATE TABLE WORKOUT(
PID VARCHAR2(8) NOT NULL,
WEEK_NO NUMBER(2) NOT NULL,
EID VARCHAR2(8) not null,
MINUTES NUMBER(5,2) NOT NULL,
SETS NUMBER(5,2) NOT NULL,
REPS NUMBER(5,2) NOT NULL,
PAIN NUMBER (2) NOT NULL,
STRESS_LEVEL NUMBER (5,2) NOT NULL,
EMPID VARCHAR2(8) NOT NULL,
constraint WORKOUT_PK primary key (PID,WEEK_NO,EID),
CONSTRAINT PATIENT_EXERCISE_EMPLOYEE_FK1 FOREIGN KEY (EID) REFERENCES
EXERCISE(EID),
CONSTRAINT PATIENT_EXERCISE_EMPLOYEE_FK2 FOREIGN KEY (PID) REFERENCES
PATIENT(PID));
CREATE TABLE DISEASE(
DID VARCHAR2(8) not null,
DISEASE_NAME VARCHAR(30) NOT NULL,
DESCRIPTION VARCHAR(100) ,
constraint DISEASE_PK primary key (DID) );
CREATE TABLE PATIENTDISEASE(
DID VARCHAR2(8) NOT NULL ,
PID VARCHAR2(8) not null,
STATUS VARCHAR(1) NOT NULL,
constraint PARIENT_DISEASE_PK primary key (DID,PID,STATUS) ,
CONSTRAINT PATIENT_DISEASE_FK1 FOREIGN KEY (DID) REFERENCES DISEASE(DID),
CONSTRAINT PATIENT_INJURY_FK2 FOREIGN KEY (PID) REFERENCES PATIENT(PID));
```

Appendix E – KRT Patient Data Excluding First Name and Cause of Injury

ID	LNAME	G	HEIGHT	S	JOB	AGE	WEEKS	WRKOUT	PAIN	STRESS	DISEAS	NO_MED	DAYS_MD	NUTRIT
1	Thompson	M	72	N	CONSTR-UCTION	64	12	17	838	4	2	2	2378	12590
2	Green	M	67	N	MANAGER	69	13	17	937	7	2	3	1870	7750
3	King	M	65	N	SALES	70	11	16	756	5	2	1	2287	11295
4	Berry	M	69	Y	SALES	75	15	18	1127	7	6	4	23438	8205
5	Thompson	F	62	N	TEACHER	35	12	14	861	8	1	2	2142	14660
6	Redish	M	70	N	PLUMBER	54	12	14	833	6	4	3	1221	17695
7	Jenkins	M	76	Y	ENGINEER	59	14	17	1055	8	4	2	10417	16390
8	Jamil	M	65	N	TEACHER	45	10	12	688	2	0	0	0	12810
9	Green	F	66	Y	STUDENT	67	12	14	869	4	1	1	4057	14480
10	Tomcat	M	69	N	FOOTBALL COACH	72	14	17	1042	5	2	1	1585	11990

Appendix F – KRT Fuzzy Attributes

FAID	ATTRIBUTE
380	AGE
381	WRKOUT
382	PAIN
383	STRESS
384	DISEAS
385	NO_MED
386	DAYS_MD
387	NUTRIT

KRT Fuzzy Categories

FUZZYCATID	NAME
9	Low
10	Mid
11	High

Appendix G: Fuzzy Membership Values

MEMBID	WTID	FAID	ATTRIBUTE	FUZZYCAT	START_POINT	MID1	MID2	END_POINT	GRAPH
1342	23	380	AGE	Low	25	39	-999	54	TRI
1344	23	380	AGE	Mid	44	55	-999	67	TRI
1346	23	380	AGE	High	59	72	-999	85	TRI
1348	23	380	AGE	Low	25	34	43	54	TRP
1350	23	380	AGE	Mid	47	53	59	67	TRP
1352	23	380	AGE	High	62	69	76	85	TRP
1354	23	386	DAYS_MD	Low	-10	995	-999	2000	TRI
1356	23	386	DAYS_MD	Mid	1297	1648	-999	2000	TRI
1358	23	386	DAYS_MD	High	1754	12382	-999	23010	TRI
1360	23	386	DAYS_MD	Low	-10	660	1330	2000	TRP
1362	23	386	DAYS_MD	Mid	1531	1687	1843	2000	TRP
1364	23	386	DAYS_MD	High	1891	8930	15969	23010	TRP
1366	23	384	DISEAS	Low	-10	-4	-999	1	TRI
1368	23	384	DISEAS	Mid	-2	0	-999	2	TRI
1370	23	384	DISEAS	High	1	8	-999	16	TRI
1372	23	384	DISEAS	Low	-10	-7	-4	1	TRP
1374	23	384	DISEAS	Mid	-2	-1	0	2	TRP
1376	23	384	DISEAS	High	1	6	11	16	TRP
1378	23	385	NO_MED	Low	-10	-4	-999	1	TRI
1380	23	385	NO_MED	Mid	-2	0	-999	2	TRI
1382	23	385	NO_MED	High	1	7	-999	14	TRI

Appendix H – KRT Partial Fuzzy Rule Set

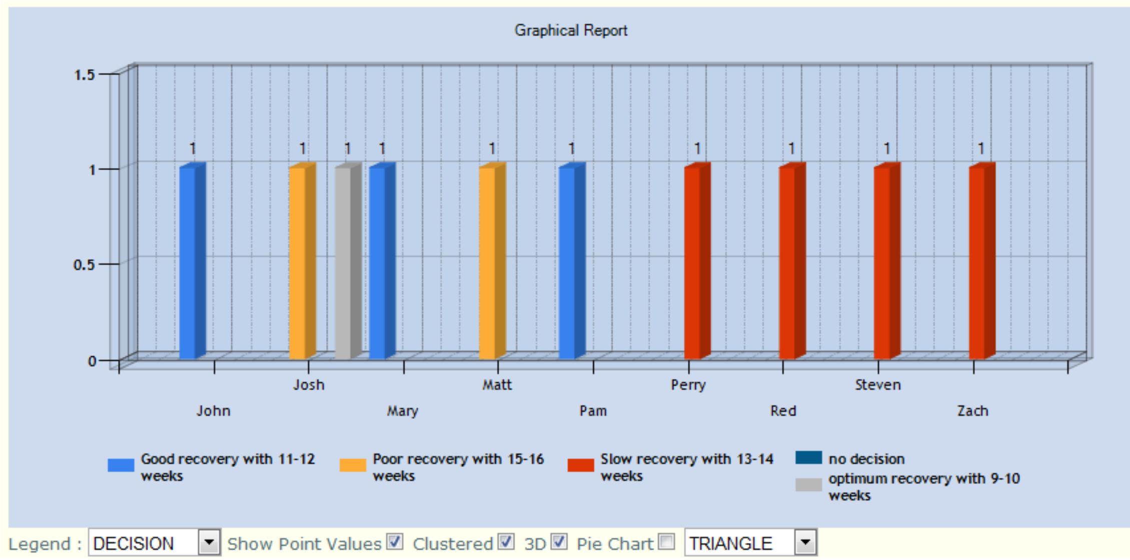
ID	AGE_TRI	WRKOUT_TRI	PAIN_TRI	STRESS_TRI	DISEAS_TRI	NO_MED_TRI	DAYS_MD_TRI	NUTRIT_TRI	AGE_TRP	WRKOUT_TRP	PAIN_TRP	STRESS_TRP	DISEAS_TRP	NO_MED_TRP	DAYS_MD_TRP	NUTRIT_TRP
1	Mid	High	Mid	Mid	Mid	Mid	Low	Mid	Mid	High	Low	Mid	Mid	Mid	Low	Mid
2	High	High	Mid	High	Mid	Mid	Low	Low	High	High	Mid	High	Mid	Mid	Low	Low
3	High	High	Low	Mid	Mid	Low	Low	Mid	High	High	Low	Mid	Mid	Low	Low	Mid
4	High	High	High	High	High	High	High	Low	High	High	High	High	High	High	High	Low
5	Low	Mid	Mid	High	Low	Mid	Low	High	Low	Mid	Mid	High	Mid	Mid	Low	High
6	Mid	Mid	Mid	Mid	High	Mid	Low	High	Mid	Mid	Low	Mid	High	Mid	Low	High
7	Mid	High	High	High	High	Mid	High	High	Mid	High	High	High	High	Mid	Mid	High
8	Low	Mid	Low	Low	Low	Low	Low	Mid	Low	Mid	Low	Low	Low	Low	Low	Mid
9	High	Mid	Mid	Mid	Low	Low	Mid	High	High	Mid	Mid	Mid	Mid	Low	Mid	High
10	High	High	High	Mid	Mid	Low	Low	Mid	High	High	High	Mid	Mid	Low	Low	Mid

Appendix I: KRT Fuzzy Decisions

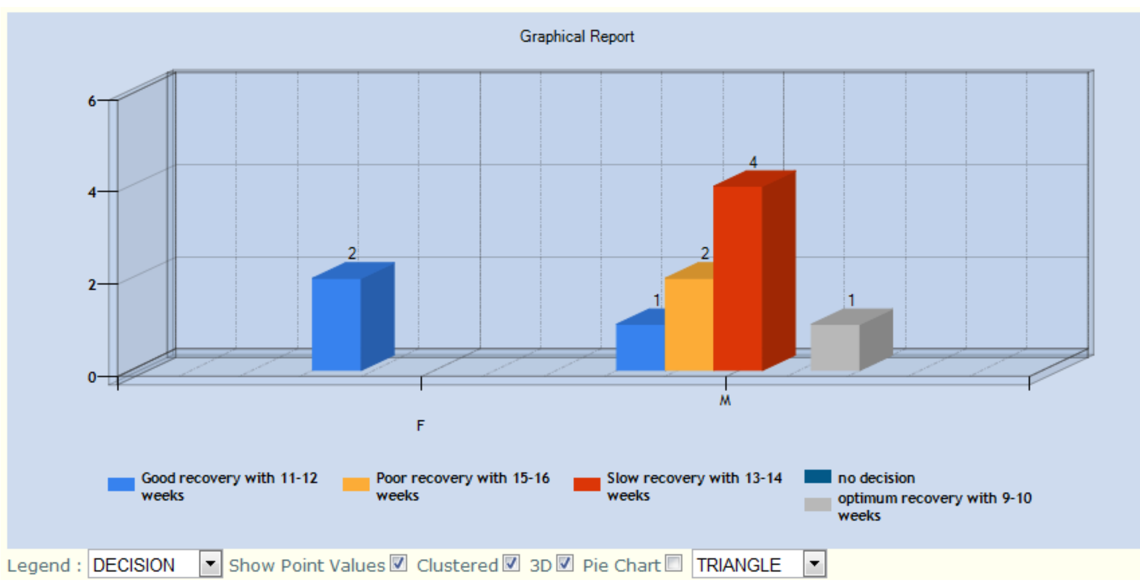
ID	FUZZYCATSETID	GRAPH	DEC_TRI	DEC_TRP
1	1	TRI-TRP	Poor recovery with 15-16 weeks	Poor recovery with 15-16 weeks
2	1	TRI-TRP	no decision	no decision
3	1	TRI-TRP	Poor recovery with 15-16 weeks	no decision
4	1	TRI-TRP	no decision	no decision
5	1	TRI-TRP	Good recovery with 11-12 weeks	Good recovery with 11-12 weeks
6	1	TRI-TRP	Slow recovery with 13-14 weeks	Slow recovery with 13-14 weeks
7	1	TRI-TRP	Poor recovery with 15-16 weeks	Poor recovery with 15-16 weeks
8	1	TRI-TRP	no decision	no decision
9	1	TRI-TRP	Slow recovery with 13-14 weeks	no decision
10	1	TRI-TRP	Poor recovery with 15-16 weeks	no decision
1	2	TRI-TRP	Good recovery with 11-12 weeks	no decision
2	2	TRI-TRP	Slow recovery with 13-14 weeks	Slow recovery with 13-14 weeks
3	2	TRI-TRP	Slow recovery with 13-14 weeks	Slow recovery with 13-14 weeks
4	2	TRI-TRP	Poor recovery with 15-16 weeks	Poor recovery with 15-16 weeks
5	2	TRI-TRP	Good recovery with 11-12 weeks	no decision
6	2	TRI-TRP	Slow recovery with 13-14 weeks	no decision
7	2	TRI-TRP	Poor recovery with 15-16 weeks	no decision
8	2	TRI-TRP	optimum recovery with 9-10 weeks	optimum recovery with 9-10 weeks

Appendix J - Data Mining Examples

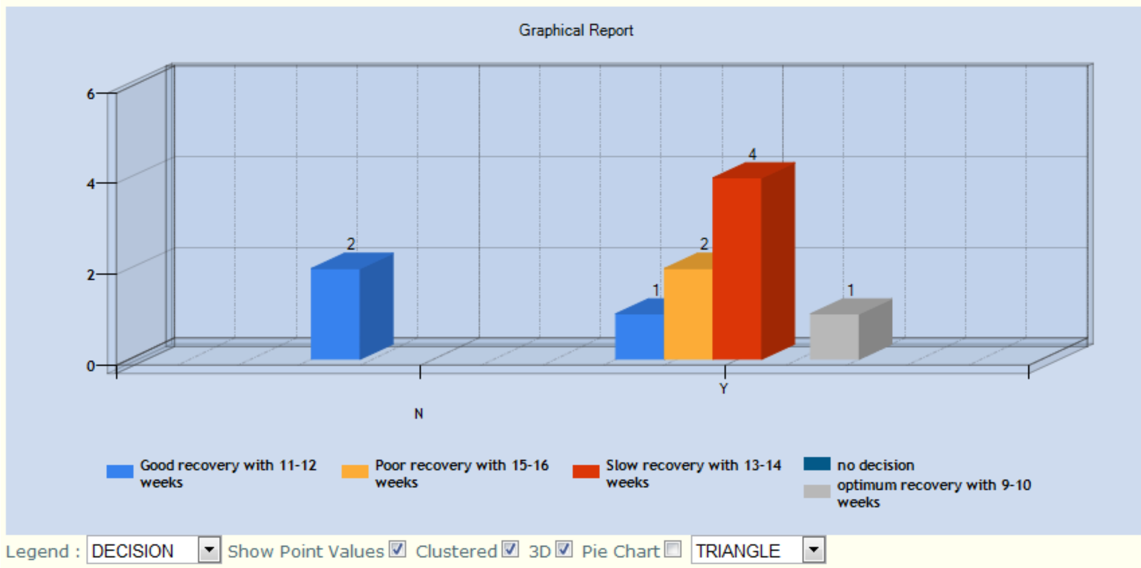
Decision about individual patients through visualization



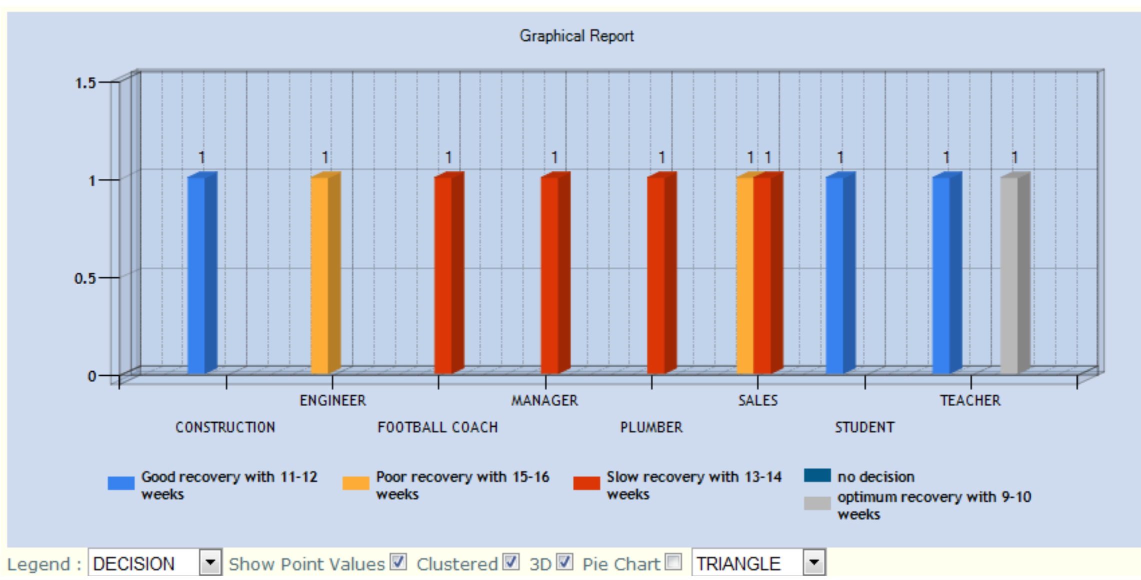
Overall decision based on gender



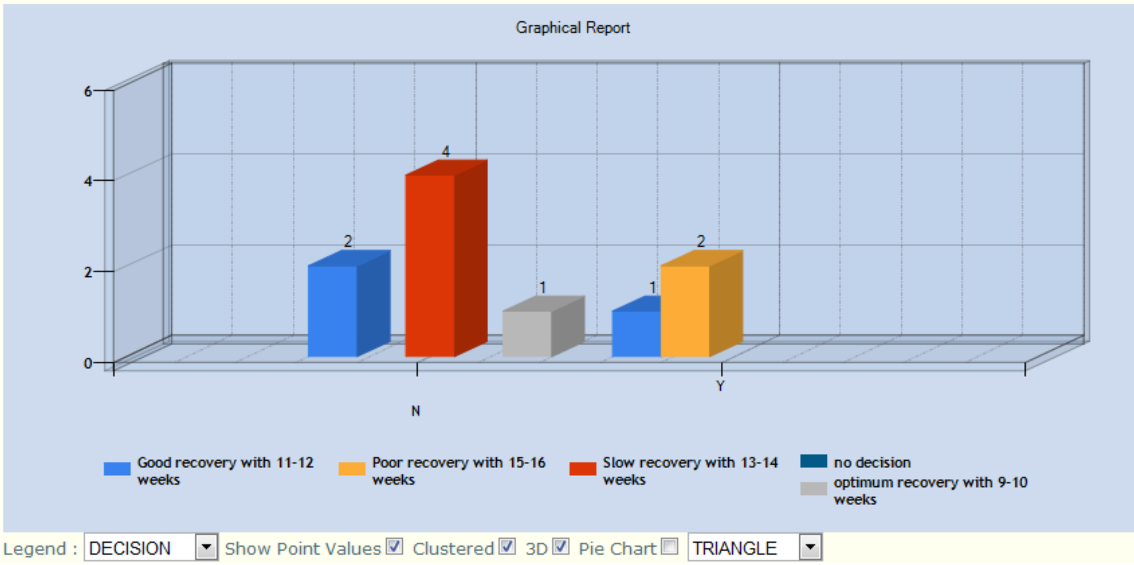
Decision based on patients with insurance



Overall decision based on each patient's occupancy



Overall decision based on each patient's smoking habits


Graphical Report
6
4
2
0
2
4
1
1
2
N
Y
Good recovery with 11-12 weeks
Poor recovery with 15-16 weeks
Slow recovery with 13-14 weeks
no decision
optimum recovery with 9-10 weeks
Legend : DECISION
Show Point Values
Clustered
3D
Pie Chart
TRIANGLE

Appendix K: Prediction Results After 5 Weeks

Appendix K: Prediction Results After 5 Weeks Parameter	Weekly Program Weight	Fuzzy Logic Membership Weight	Weighted Weekly Value	Weekly Progress
week 1 a	0.17	1.33	0.22	0.22
week 1 w	0.17	1.33	0.22	0.44
week 1 p	0.08	1.33	0.11	0.56
week 1 s	0.13	1.33	0.17	0.72
week 1 d	0.17	1.33	0.22	0.94
week 1 m	0.08	1.33	0.11	1.06
week 1 z	0.08	1.33	0.11	1.17
week 1 n	0.13	1.33	0.17	1.33
week 2 a	0.17	1.33	0.22	1.56
week 2 w	0.17	1.33	0.22	1.78
week 2 p	0.08	1.33	0.11	1.89
week 2 s	0.13	1.33	0.17	2.06
week 2 d	0.17	1.33	0.22	2.28
week 2 m	0.08	1.33	0.11	2.39
week 2 z	0.08	1.33	0.11	2.50
week 2 n	0.13	1.33	0.17	2.67
week 3 a	0.17	1.33	0.22	2.89

week 3 w	0.17	1.33	0.22	3.11
week 3 p	0.08	1.33	0.11	3.22
week 3 s	0.13	1.33	0.17	3.39
week 3 d	0.17	1.33	0.22	3.61
week 3 m	0.08	1.33	0.11	3.72
week 3 z	0.08	1.33	0.11	3.83
week 3 n	0.13	1.33	0.17	4.00
week 4 a	0.17	1.33	0.22	4.22
week 4 w	0.17	1.33	0.22	4.44
week 4 p	0.08	1.33	0.11	4.56
week 4 s	0.13	1.33	0.17	4.72
week 4 d	0.17	1.33	0.22	4.94
week 4 m	0.08	1.33	0.11	5.06
week 4 z	0.08	1.33	0.11	5.17
week 4 n	0.13	1.33	0.17	5.33
week 5 a	0.17	1.33	0.22	5.56
week 5 w	0.17	1.33	0.22	5.78
week 5 p	0.08	1.33	0.11	5.89
week 5 s	0.13	1.33	0.17	6.06
week 5 d	0.17	1.33	0.22	6.28
week 5 m	0.08	1.33	0.11	6.39
week 5 z	0.08	1.33	0.11	6.50
week 5 n	0.13	1.33	0.17	6.67
week 6 a	0.17	1.33	0.22	6.89

week 6 w	0.17	1.33	0.22	7.11
week 6 p	0.08	1.33	0.11	7.22
week 6 s	0.13	1.33	0.17	7.39
week 6 d	0.17	1.33	0.22	7.61
week 6 m	0.08	1.33	0.11	7.72
week 6 z	0.08	1.33	0.11	7.83
week 6 n	0.13	1.33	0.17	8.00
week 7 a	0.17	1.33	0.22	8.22
week 7 w	0.17	1.33	0.22	8.44
week 7 p	0.08	1.33	0.11	8.56
week 7 s	0.13	1.33	0.17	8.72
week 7 d	0.17	1.33	0.22	8.94
week 7 m	0.08	1.33	0.11	9.06
week 7 z	0.08	1.33	0.11	9.17
week 7 n	0.13	1.33	0.17	9.33
week 8 a	0.17	1.33	0.22	9.56
week 8 w	0.17	1.33	0.22	9.78
week 8 p	0.08	1.33	0.11	9.89
week 8 s	0.13	1.33	0.17	10.06
week 8 d	0.17	1.33	0.22	10.28
week 8 m	0.08	1.33	0.11	10.39
week 8 z	0.08	1.33	0.11	10.50
week 8 n	0.13	1.33	0.17	10.67
week 9 a	0.17	1.33	0.22	10.89

week 9 w	0.17	1.33	0.22	11.11
week 9 p	0.08	1.33	0.11	11.22
week 9 s	0.13	1.33	0.17	11.39
week 9 d	0.17	1.33	0.22	11.61
week 9 m	0.08	1.33	0.11	11.72
week 9 z	0.08	1.33	0.11	11.83
week 9 n	0.13	1.33	0.17	12.00
week 10 a	0.17	1.33	0.22	12.22
week 10 w	0.17	1.33	0.22	12.44
week 10 p	0.08	1.33	0.11	12.56
week 10 s	0.13	1.33	0.17	12.72
week 10 d	0.17	1.33	0.22	12.94
week 10 m	0.08	1.33	0.11	13.06
week 10 z	0.08	1.33	0.11	13.17
week 10 n	0.13	1.33	0.17	13.33
week 11 a	0.17	1.33	0.22	13.56
week 11 w	0.17	1.33	0.22	13.78
week 11 p	0.08	1.33	0.11	13.89
week 11 s	0.13	1.33	0.17	14.06
week 11 d	0.17	1.33	0.22	14.28
week 11 m	0.08	1.33	0.11	14.39
week 11 z	0.08	1.33	0.11	14.50
week 11 n	0.13	1.33	0.17	14.67
week 12 a	0.17	1.33	0.22	14.89

week 12 w	0.17	1.33	0.22	15.11
week 12 p	0.08	1.33	0.11	15.22
week 12 s	0.13	1.33	0.17	15.39
week 12 d	0.17	1.33	0.22	15.61
week 12 m	0.08	1.33	0.11	15.72
week 12 z	0.08	1.33	0.11	15.83
week 12 n	0.13	1.33	0.17	16.00
week 13 a	0.17	1.33	0.22	16.22
week 13 w	0.17	1.33	0.22	16.44
week 13 p	0.08	1.33	0.11	16.56
week 13 s	0.13	1.33	0.17	16.72
week 13 d	0.17	1.33	0.22	16.94
week 13 m	0.08	1.33	0.11	17.06
week 13 z	0.08	1.33	0.11	17.17
week 13 n	0.13	1.33	0.17	17.33
week 14 a	0.17	1.33	0.22	17.56
week 14 w	0.17	1.33	0.22	17.78
week 14 p	0.08	1.33	0.11	17.89
week 14 s	0.13	1.33	0.17	18.06
week 14 d	0.17	1.33	0.22	18.28
week 14 m	0.08	1.33	0.11	18.39
week 14 z	0.08	1.33	0.11	18.50
week 14 n	0.13	1.33	0.17	18.67
week 15 a	0.17	1.33	0.22	18.89

week 15 w	0.17	1.33	0.22	19.11
week 15 p	0.08	1.33	0.11	19.22
week 15 s	0.13	1.33	0.17	19.39
week 15 d	0.17	1.33	0.22	19.61
week 15 m	0.08	1.33	0.11	19.72
week 15 z	0.08	1.33	0.11	19.83
week 15 n	0.13	1.33	0.17	20.00
week 15 a	0.17	1.33	0.22	20.22
week 15 w	0.17	1.33	0.22	20.44
week 15 p	0.08	1.33	0.11	20.56
week 15 s	0.13	1.33	0.17	20.72
week 15 d	0.17	1.33	0.22	20.94
week 15 m	0.08	1.33	0.11	21.06
week 15 z	0.08	1.33	0.11	21.17
week 15 n	0.13	1.33	0.17	21.33

Appendix K2: Weighted Fuzzy Values for each Week, for each Attribute in a Worst Case Scenario

Parameter	Weekly Program Weight	Fuzzy Logic Membership Weight	Weighted Weekly Value	Weekly Progress
week 1 a	0.17	0.75	0.13	0.13
week 1 w	0.17	0.75	0.13	0.25
week 1 p	0.08	0.75	0.06	0.31
week 1 s	0.13	0.75	0.09	0.41
week 1 d	0.17	0.75	0.13	0.53
week 1 m	0.08	0.75	0.06	0.59
week 1 z	0.08	0.75	0.06	0.66
week 1 n	0.13	0.75	0.09	0.75
week 2 a	0.17	0.75	0.13	0.88
week 2 w	0.17	0.75	0.13	1.00
week 2 p	0.08	0.75	0.06	1.06
week 2 s	0.13	0.75	0.09	1.16
week 2 d	0.17	0.75	0.13	1.28
week 2 m	0.08	0.75	0.06	1.34
week 2 z	0.08	0.75	0.06	1.41
week 2 n	0.13	0.75	0.09	1.50
week 3 a	0.17	0.75	0.13	1.63
week 3 w	0.17	0.75	0.13	1.75
week 3 p	0.08	0.75	0.06	1.81

week 3 s	0.13	0.75	0.09	1.91
week 3 d	0.17	0.75	0.13	2.03
week 3 m	0.08	0.75	0.06	2.09
week 3 z	0.08	0.75	0.06	2.16
week 3 n	0.13	0.75	0.09	2.25
week 4 a	0.17	0.75	0.13	2.38
week 4 w	0.17	0.75	0.13	2.50
week 4 p	0.08	0.75	0.06	2.56
week 4 s	0.13	0.75	0.09	2.66
week 4 d	0.17	0.75	0.13	2.78
week 4 m	0.08	0.75	0.06	2.84
week 4 z	0.08	0.75	0.06	2.91
week 4 n	0.13	0.75	0.09	3.00
week 5 a	0.17	0.75	0.13	3.13
week 5 w	0.17	0.75	0.13	3.25
week 5 p	0.08	0.75	0.06	3.31
week 5 s	0.13	0.75	0.09	3.41
week 5 d	0.17	0.75	0.13	3.53
week 5 m	0.08	0.75	0.06	3.59
week 5 z	0.08	0.75	0.06	3.66
week 5 n	0.13	0.75	0.09	3.75
week 6 a	0.17	0.75	0.13	3.88
week 6 w	0.17	0.75	0.13	4.00
week 6 p	0.08	0.75	0.06	4.06

week 6 s	0.13	0.75	0.09	4.16
week 6 d	0.17	0.75	0.13	4.28
week 6 m	0.08	0.75	0.06	4.34
week 6 z	0.08	0.75	0.06	4.41
week 6 n	0.13	0.75	0.09	4.50
week 7 a	0.17	0.75	0.13	4.63
week 7 w	0.17	0.75	0.13	4.75
week 7 p	0.08	0.75	0.06	4.81
week 7 s	0.13	0.75	0.09	4.91
week 7 d	0.17	0.75	0.13	5.03
week 7 m	0.08	0.75	0.06	5.09
week 7 z	0.08	0.75	0.06	5.16
week 7 n	0.13	0.75	0.09	5.25
week 8 a	0.17	0.75	0.13	5.38
week 8 w	0.17	0.75	0.13	5.50
week 8 p	0.08	0.75	0.06	5.56
week 8 s	0.13	0.75	0.09	5.66
week 8 d	0.17	0.75	0.13	5.78
week 8 m	0.08	0.75	0.06	5.84
week 8 z	0.08	0.75	0.06	5.91
week 8 n	0.13	0.75	0.09	6.00
week 9 a	0.17	0.75	0.13	6.13
week 9 w	0.17	0.75	0.13	6.25
week 9 p	0.08	0.75	0.06	6.31

week 9 s	0.13	0.75	0.09	6.41
week 9 d	0.17	0.75	0.13	6.53
week 9 m	0.08	0.75	0.06	6.59
week 9 z	0.08	0.75	0.06	6.66
week 9 n	0.13	0.75	0.09	6.75
week 10 a	0.17	0.75	0.13	6.88
week 10 w	0.17	0.75	0.13	7.00
week 10 p	0.08	0.75	0.06	7.06
week 10 s	0.13	0.75	0.09	7.16
week 10 d	0.17	0.75	0.13	7.28
week 10 m	0.08	0.75	0.06	7.34
week 10 z	0.08	0.75	0.06	7.41
week 10 n	0.13	0.75	0.09	7.50
week 11 a	0.17	0.75	0.13	7.63
week 11 w	0.17	0.75	0.13	7.75
week 11 p	0.08	0.75	0.06	7.81
week 11 s	0.13	0.75	0.09	7.91
week 11 d	0.17	0.75	0.13	8.03
week 11 m	0.08	0.75	0.06	8.09
week 11 z	0.08	0.75	0.06	8.16
week 11 n	0.13	0.75	0.09	8.25
week 12 a	0.17	0.75	0.13	8.38
week 12 w	0.17	0.75	0.13	8.50
week 12 p	0.08	0.75	0.06	8.56

week 12 s	0.13	0.75	0.09	8.66
week 12 d	0.17	0.75	0.13	8.78
week 12 m	0.08	0.75	0.06	8.84
week 12 z	0.08	0.75	0.06	8.91
week 12 n	0.13	0.75	0.09	9.00
week 13 a	0.17	0.75	0.13	9.13
week 13 w	0.17	0.75	0.13	9.25
week 13 p	0.08	0.75	0.06	9.31
week 13 s	0.13	0.75	0.09	9.41
week 13 d	0.17	0.75	0.13	9.53
week 13 m	0.08	0.75	0.06	9.59
week 13 z	0.08	0.75	0.06	9.66
week 13 n	0.13	0.75	0.09	9.75
week 14 a	0.17	0.75	0.13	9.88
week 14 w	0.17	0.75	0.13	10.00
week 14 p	0.08	0.75	0.06	10.06
week 14 s	0.13	0.75	0.09	10.16
week 14 d	0.17	0.75	0.13	10.28
week 14 m	0.08	0.75	0.06	10.34
week 14 z	0.08	0.75	0.06	10.41
week 14 n	0.13	0.75	0.09	10.50
week 15 a	0.17	0.75	0.13	10.63
week 15 w	0.17	0.75	0.13	10.75
week 15 p	0.08	0.75	0.06	10.81

week 15 s	0.13	0.75	0.09	10.91
week 15 d	0.17	0.75	0.13	11.03
week 15 m	0.08	0.75	0.06	11.09
week 15 z	0.08	0.75	0.06	11.16
week 15 n	0.13	0.75	0.09	11.25
week 16 a	0.17	0.75	0.13	11.38
week 16 w	0.17	0.75	0.13	11.50
week 16 p	0.08	0.75	0.06	11.56
week 16 s	0.13	0.75	0.09	11.66
week 16 d	0.17	0.75	0.13	11.78
week 16 m	0.08	0.75	0.06	11.84
week 16 z	0.08	0.75	0.06	11.91
week 16 n	0.13	0.75	0.09	12.00

Appendix K3: Prediction Results After 5 Weeks in an Optimal Setting

Parameter	Weekly Program Weight	Fuzzy Logic Membership Weight	Weighted Weekly Value	Weekly Progress
week 1 a	0.17	1.33	0.22	0.22
week 1 w	0.17	0.75	0.13	0.35
week 1 p	0.08	1.00	0.08	0.43
week 1 s	0.13	1.33	0.17	0.60
week 1 d	0.17	1.00	0.17	0.76
week 1 m	0.08	1.00	0.08	0.85
week 1 z	0.08	0.75	0.06	0.91
week 1 n	0.13	0.75	0.09	1.00
week 2 a	0.17	1.33	0.22	1.23
week 2 w	0.17	0.75	0.13	1.35
week 2 p	0.08	1.00	0.08	1.43
week 2 s	0.13	1.33	0.17	1.60
week 2 d	0.17	1.00	0.17	1.77
week 2 m	0.08	1.00	0.08	1.85
week 2 z	0.08	0.75	0.06	1.91
week 2 n	0.13	0.75	0.09	2.01
week 3 a	0.17	1.33	0.22	2.23
week 3 w	0.17	0.75	0.13	2.35
week 3 p	0.08	1.00	0.08	2.44

week 3 s	0.13	1.33	0.17	2.60
week 3 d	0.17	1.00	0.17	2.77
week 3 m	0.08	1.00	0.08	2.85
week 3 z	0.08	0.75	0.06	2.92
week 3 n	0.13	0.75	0.09	3.01
week 4 a	0.17	1.33	0.22	3.23
week 4 w	0.17	0.75	0.13	3.36
week 4 p	0.08	1.00	0.08	3.44
week 4 s	0.13	1.33	0.17	3.61
week 4 d	0.17	1.00	0.17	3.77
week 4 m	0.08	1.00	0.08	3.86
week 4 z	0.08	0.75	0.06	3.92
week 4 n	0.13	0.75	0.09	4.01
week 5 a	0.17	1.33	0.22	4.24
week 5 w	0.17	0.75	0.13	4.36
week 5 p	0.08	1.00	0.08	4.44
week 5 s	0.13	1.33	0.17	4.61
week 5 d	0.17	1.00	0.17	4.78
week 5 m	0.08	1.00	0.08	4.86
week 5 z	0.08	0.75	0.06	4.92
week 5 n	0.13	0.75	0.09	5.02
week 6 a	0.17	1.33	0.22	5.24
week 6 w	0.17	1.33	0.22	5.46
week 6 p	0.08	1.33	0.11	5.57

week 6 s	0.13	1.33	0.17	5.74
week 6 d	0.17	1.33	0.22	5.96
week 6 m	0.08	1.33	0.11	6.07
week 6 z	0.08	1.33	0.11	6.18
week 6 n	0.13	1.33	0.17	6.35
week 7 a	0.17	1.33	0.22	6.57
week 7 w	0.17	1.33	0.22	6.80
week 7 p	0.08	1.33	0.11	6.91
week 7 s	0.13	1.33	0.17	7.07
week 7 d	0.17	1.33	0.22	7.30
week 7 m	0.08	1.33	0.11	7.41
week 7 z	0.08	1.33	0.11	7.52
week 7 n	0.13	1.33	0.17	7.68
week 8 a	0.17	1.33	0.22	7.91
week 8 w	0.17	1.33	0.22	8.13
week 8 p	0.08	1.33	0.11	8.24
week 8 s	0.13	1.33	0.17	8.41
week 8 d	0.17	1.33	0.22	8.63
week 8 m	0.08	1.33	0.11	8.74
week 8 z	0.08	1.33	0.11	8.85
week 8 n	0.13	1.33	0.17	9.02
week 9 a	0.17	1.33	0.22	9.24
week 9 w	0.17	1.33	0.22	9.46
week 9 p	0.08	1.33	0.11	9.57

week 9 s	0.13	1.33	0.17	9.74
week 9 d	0.17	1.33	0.22	9.96
week 9 m	0.08	1.33	0.11	10.07
week 9 z	0.08	1.33	0.11	10.18
week 9 n	0.13	1.33	0.17	10.35
week 10 a	0.17	1.33	0.22	10.57
week 10 w	0.17	1.33	0.22	10.80
week 10 p	0.08	1.33	0.11	10.91
week 10 s	0.13	1.33	0.17	11.07
week 10 d	0.17	1.33	0.22	11.30
week 10 m	0.08	1.33	0.11	11.41
week 10 z	0.08	1.33	0.11	11.52
week 10 n	0.13	1.33	0.17	11.68
week 11 a	0.17	1.33	0.22	11.91
week 11 w	0.17	1.33	0.22	12.13
week 11 p	0.08	1.33	0.11	12.24
week 11 s	0.13	1.33	0.17	12.41
week 11 d	0.17	1.33	0.22	12.63
week 11 m	0.08	1.33	0.11	12.74
week 11 z	0.08	1.33	0.11	12.85
week 11 n	0.13	1.33	0.17	13.02

Appendix K4: Prediction Results After 5 Weeks in a Worst Case Scenario

Parameter	Weekly Program Weight	Fuzzy Logic Membership Weight	Weighted Weekly Value	Weekly Progress
week 1 a	0.17	1.33	0.22	0.22
week 1 w	0.17	0.75	0.13	0.35
week 1 p	0.08	1.00	0.08	0.43
week 1 s	0.13	1.33	0.17	0.60
week 1 d	0.17	1.00	0.17	0.76
week 1 m	0.08	1.00	0.08	0.85
week 1 z	0.08	0.75	0.06	0.91
week 1 n	0.13	0.75	0.09	1.00
week 2 a	0.17	1.33	0.22	1.23
week 2 w	0.17	0.75	0.13	1.35
week 2 p	0.08	1.00	0.08	1.43
week 2 s	0.13	1.33	0.17	1.60
week 2 d	0.17	1.00	0.17	1.77
week 2 m	0.08	1.00	0.08	1.85
week 2 z	0.08	0.75	0.06	1.91
week 2 n	0.13	0.75	0.09	2.01
week 3 a	0.17	1.33	0.22	2.23
week 3 w	0.17	0.75	0.13	2.35
week 3 p	0.08	1.00	0.08	2.44

week 3 s	0.13	1.33	0.17	2.60
week 3 d	0.17	1.00	0.17	2.77
week 3 m	0.08	1.00	0.08	2.85
week 3 z	0.08	0.75	0.06	2.92
week 3 n	0.13	0.75	0.09	3.01
week 4 a	0.17	1.33	0.22	3.23
week 4 w	0.17	0.75	0.13	3.36
week 4 p	0.08	1.00	0.08	3.44
week 4 s	0.13	1.33	0.17	3.61
week 4 d	0.17	1.00	0.17	3.77
week 4 m	0.08	1.00	0.08	3.86
week 4 z	0.08	0.75	0.06	3.92
week 4 n	0.13	0.75	0.09	4.01
week 5 a	0.17	1.33	0.22	4.24
week 5 w	0.17	0.75	0.13	4.36
week 5 p	0.08	1.00	0.08	4.44
week 5 s	0.13	1.33	0.17	4.61
week 5 d	0.17	1.00	0.17	4.78
week 5 m	0.08	1.00	0.08	4.86
week 5 z	0.08	0.75	0.06	4.92
week 5 n	0.13	0.75	0.09	5.02
week 6 a	0.17	1.33	0.22	5.24
week 6 w	0.17	0.75	0.13	5.36
week 6 p	0.08	0.75	0.06	5.43

week 6 s	0.13	0.75	0.09	5.52
week 6 d	0.17	0.75	0.13	5.65
week 6 m	0.08	0.75	0.06	5.71
week 6 z	0.08	0.75	0.06	5.77
week 6 n	0.13	0.75	0.09	5.86
week 7 a	0.17	1.33	0.22	6.09
week 7 w	0.17	0.75	0.13	6.21
week 7 p	0.08	0.75	0.06	6.27
week 7 s	0.13	0.75	0.09	6.37
week 7 d	0.17	0.75	0.13	6.49
week 7 m	0.08	0.75	0.06	6.56
week 7 z	0.08	0.75	0.06	6.62
week 7 n	0.13	0.75	0.09	6.71
week 8 a	0.17	1.33	0.22	6.93
week 8 w	0.17	0.75	0.13	7.06
week 8 p	0.08	0.75	0.06	7.12
week 8 s	0.13	0.75	0.09	7.22
week 8 d	0.17	0.75	0.13	7.34
week 8 m	0.08	0.75	0.06	7.40
week 8 z	0.08	0.75	0.06	7.47
week 8 n	0.13	0.75	0.09	7.56
week 9 a	0.17	1.33	0.22	7.78
week 9 w	0.17	0.75	0.13	7.91
week 9 p	0.08	0.75	0.06	7.97

week 9 s	0.13	0.75	0.09	8.06
week 9 d	0.17	0.75	0.13	8.19
week 9 m	0.08	0.75	0.06	8.25
week 9 z	0.08	0.75	0.06	8.31
week 9 n	0.13	0.75	0.09	8.41
week 10 a	0.17	1.33	0.22	8.63
week 10 w	0.17	0.75	0.13	8.75
week 10 p	0.08	0.75	0.06	8.82
week 10 s	0.13	0.75	0.09	8.91
week 10 d	0.17	0.75	0.13	9.03
week 10 m	0.08	0.75	0.06	9.10
week 10 z	0.08	0.75	0.06	9.16
week 10 n	0.13	0.75	0.09	9.25
week 11 a	0.17	1.33	0.22	9.48
week 11 w	0.17	0.75	0.13	9.60
week 11 p	0.08	0.75	0.06	9.66
week 11 s	0.13	0.75	0.09	9.76
week 11 d	0.17	0.75	0.13	9.88
week 11 m	0.08	0.75	0.06	9.94
week 11 z	0.08	0.75	0.06	10.01
week 11 n	0.13	0.75	0.09	10.10
week 12 a	0.17	1.33	0.22	10.32
week 12 w	0.17	0.75	0.13	10.45
week 12 p	0.08	0.75	0.06	10.51

week 12 s	0.13	0.75	0.09	10.60
week 12 d	0.17	0.75	0.13	10.73
week 12 m	0.08	0.75	0.06	10.79
week 12 z	0.08	0.75	0.06	10.85
week 12 n	0.13	0.75	0.09	10.95
week 13 a	0.17	1.33	0.22	11.17
week 13 w	0.17	0.75	0.13	11.30
week 13 p	0.08	0.75	0.06	11.36
week 13 s	0.13	0.75	0.09	11.45
week 13 d	0.17	0.75	0.13	11.58
week 13 m	0.08	0.75	0.06	11.64
week 13 z	0.08	0.75	0.06	11.70
week 13 n	0.13	0.75	0.09	11.80
week 14 a	0.17	1.33	0.22	12.02
week 14 w	0.17	0.75	0.13	12.14
week 14 p	0.08	0.75	0.06	12.20
week 14 s	0.13	0.75	0.09	12.30
week 14 d	0.17	0.75	0.13	12.42
week 14 m	0.08	0.75	0.06	12.49
week 14 z	0.08	0.75	0.06	12.55
week 14 n	0.13	0.75	0.09	12.64

Appendix L: Procedure to calculate progress amount for a specific partial dataset

```
create or replace

procedure CreateProcedureProgress(TableName in varchar2, wt_id in number,calcType in varchar2) is

   txt varchar2(32767) := 'create or replace procedure '||TableName||'_calculate_progress is ';

        cursor cur_cols is select column_name from user_tab_columns where upper(table_name)=upper(TableName||'_p_data'||calcType);

        counter number(10) := 0;

begin

for val_1 in cur_cols

        loop

                txt:=txt||'

                '||val_1.column_name||'_ '||TableName||'_p_data'||calcType||'.'||val_1.column_name||'%type;';

        end loop;

txt:=txt||'

'||'Progress_amt '||TableName||'_p_data'||calcType||'.Progress%type;

max_progress '||TableName||'_p_data'||calcType||'.Progress%type;

prev_id '||TableName||'_p_data'||calcType||'.ID%type;

max_id '||TableName||'_p_data'||calcType||'.ID%type;

prev_datasetID '||TableName||'_p_data'||calcType||'.datasetid%type; ';

        txt:=txt||'

        cursor cur is select ';

        for val_3 in cur_cols

        loop

                if counter > 0 then

                        txt:=txt||',';

                end if;

                txt:=txt||val_3.column_name;
```

```
                counter := counter + 1;
        end loop;
        txt:=txt||'
        from '||TableName||'_p_data'||calcType||' order by id;
  cursor cur_max_progress is select id, progress  from '||TableName||'_p_data'||calcType||'
where id =(select max(id) from '||TableName||'_p_data'||calcType||');
 ';
 txt:=txt||'begin
  progress_amt:=0;
   prev_datasetID:=0;
  prev_id:=0;
    open cur; loop
        ';
        txt:=txt||' fetch cur into
        ';
counter := 0;
        for val_1 in cur_cols
        loop
                if counter != 0 then
                        txt:=txt||',';
                end if;
                txt:=txt||'
                '||val_1.column_name||'_';
                counter := counter + 1;
        end loop;
        txt:=txt||';
        exit when cur%notfound;';
```

```
     txt:=txt||'fuzzyattrweight_ :=
ATTRIBUTEWEIGHT_*FUZZYCATCONSTANT_*ATTRIBUTERATE_;';

              txt:=txt||'progress_amt:=progress_amt+FUZZYATTRWEIGHT_;';

     txt:=txt||'update  '||TableName||'_p_data'||calcType||' set fuzzyattrweight=
fuzzyattrweight_,PROGRESS=progress_amt

    where faid=faid_ and datasetid=datasetid_;

    if (DATASETID_>prev_datasetID) then

       if(prev_id>0) then

       update '||TableName||'_p_data'||calcType||' set CUMULATIVEPROGRESS=progress_amt-
FUZZYATTRWEIGHT_

       where id=prev_id;

       prev_datasetid:=DATASETID_;

       end if;

    end if;

     prev_id:=ID_;';

           txt:=txt||' end loop; close cur;

   open cur_max_progress;

  loop

  fetch cur_max_progress into max_id,max_progress;

     exit when cur_max_progress%notfound;

   end loop;

   close cur_max_progress;

    update '||TableName||'_p_data'||calcType||' set cumulativeprogress=max_progress

       where id=max_id;


  end;';

           execute immediate txt;

end;
```

Appendix M: Procedure to calculate future progress amount and predicted data for a partial dataset

```
create or replace

procedure CreateProcedureFutureProgress(TableName in varchar2, wt_id in number,dataset in number,cat_constant in number, calcType in varchar2) is

   txt varchar2(32767) := 'create or replace procedure '||TableName||'_future_progress is ';

         cursor cur_cols is select column_name from user_tab_columns where upper(table_name)=upper(TableName||'_P_data'||calcType);

         counter number(10) := 0;

begin

for val_1 in cur_cols

         loop

                  txt:=txt||'

                  '||val_1.column_name||'_
'||TableName||'_p_data'||calcType||'.'||val_1.column_name||'%type;';

         end loop;

txt:=txt||'

'||'Progress_amount '||TableName||'_p_data'||calcType||'.Progress%type;';

txt:=txt||'

'||'max_dataset '||TableName||'_p_data'||calcType||'.datasetid%type;';

txt:=txt||'

'||'max_progress '||TableName||'_p_data'||calcType||'.progress%type;';

txt:=txt||'

'||'max_id '||TableName||'_p_data'||calcType||'.ID%type;

prev_id '||TableName||'_p_data'||calcType||'.ID%type;

prev_datasetID '||TableName||'_p_data'||calcType||'.datasetid%type; ';

         txt:=txt||'

         cursor cur is select ';

         for val_3 in cur_cols
```

```
            loop
                        if counter > 0 then
                                    txt:=txt||',';
                        end if;
                        txt:=txt||val_3.column_name;
                        counter := counter + 1;
            end loop;
            txt:=txt||'
            from '||TableName||'_p_data'||calcType||' order by id;';
            txt:=txt||'
            cursor cur_max_dataset is select max(datasetid) from
'||TableName||'_p_data'||calcType||';';
 txt:=txt||'
            cursor cur_max_progress is select progress  from
'||TableName||'_p_data'||calcType||' where id =(select max(id) from
'||TableName||'_p_data'||calcType||');';
            txt:=txt||'
            cursor cur_max_id is select max(id) from '||TableName||'_p_data'||calcType||';';
 txt:=txt||'begin
  progress_amount:=0;max_progress:=0;
  open cur_max_dataset;
 loop
 fetch cur_max_dataset into max_dataset;
    exit when cur_max_dataset%notfound;
  end loop;
  close cur_max_dataset;
   open cur_max_progress;
 loop
 fetch cur_max_progress into max_progress;
```

```
    exit when cur_max_progress%notfound;
  end loop;
  close cur_max_progress;
open cur_max_id;
 loop
 fetch cur_max_id into max_id;
    exit when cur_max_id%notfound;
  end loop;
  close cur_max_id;';
            txt:=txt||'
   while max_dataset<'||dataset||' loop
 open cur;
 loop
 fetch cur into';
counter := 0;
        for val_1 in cur_cols
        loop
                if counter != 0 then
                        txt:=txt||',';
                end if;
                txt:=txt||'
                '||val_1.column_name||'_';
                counter := counter + 1;
        end loop;
                txt:=txt||';
        exit when cur%notfound;';
   txt:=txt||'
 if (datasetid_ =max_dataset) then';
```

```
if (cat_constant=0) then

            txt:=txt||'

   insert into
'||TableName||'_p_data'||calcType||'(id,faid,datasetid,attributeweight,attributerate,fuzzycatc
onstant,wtid)
values(max_id+1,FAID_,max_dataset+1,attributeweight_,ATTRIBUTERATE_,fuzzycatconstant_,W
TID_);';

 else

   txt:=txt||'

   insert into
'||TableName||'_p_data'||calcType||'(id,faid,datasetid,attributeweight,attributerate,fuzzycatc
onstant,wtid)
values(max_id+1,FAID_,max_dataset+1,attributeweight_,ATTRIBUTERATE_,'||cat_constant||',
WTID_);';

   end if;


if (cat_constant=0) then

            txt:=txt||'

   fuzzyattrweight_ := attributeweight_*fuzzycatconstant_*ATTRIBUTERATE_;';

 else

   txt:=txt||'

   fuzzyattrweight_ := attributeweight_*'||cat_constant||'*ATTRIBUTERATE_;';

   end if;

   txt:=txt||'

      max_progress:=max_progress+fuzzyattrweight_;

   update  '||TableName||'_p_data'||calcType||' set fuzzyattrweight=
fuzzyattrweight_,progress=max_progress

   where faid=faid_ and datasetid=datasetid_+1;

    if (DATASETID_>prev_datasetID) then

      if(prev_id>0) then

      update '||TableName||'_p_data'||calcType||' set CUMULATIVEPROGRESS=max_progress-
FUZZYATTRWEIGHT_
```

```
        where id=prev_id;
        prev_datasetid:=DATASETID_;
        end if;
    end if;
      prev_id:=ID_;
      max_id:=max_id+1;
     end if;
    end loop;
  close cur;
   max_dataset:= max_dataset+1;
  end loop;
  end;';
            execute immediate txt;
end;
```

Appendix N: Procedure to create all active database components for Combs Inference Method

```
create or replace

procedure createAllCombs(TableName in varchar2, wt_id in number) is

        cursor cur_fa is select faid, attribute, col_type from fuzzyattributes where wtid =
wt_id;

        cursor c_graph is select graph from fuzzymembershipvalues

        where wtid=wt_id and upper(status)=upper('active')

        group by graph;

    begin

        -- in this loop it creates the fuzzy tables, their materialized views, triggers and
updates them.

        for val1 in c_graph

        loop

            for val in cur_fa

            loop

                createFuzzyTables(val.col_type,
TableName||'_'||val.attribute||'_'||val1.graph, wt_id);

                createTrigger(val1.graph, TableName||'_'||val.attribute, wt_id,
val.faid);

                general_update_FT(val1.graph, val.attribute, TableName);

            end loop;

        end loop;

        createFuzzyPerformanceCombs(TableName, wt_id);

        createFuzzyWeight(TableName, wt_id);

        createFuzzyPerfTriggerCombs(TableName, wt_id);

        p_update_fuzzy_perf_Combs(TableName);

        createDecisionTableCombs(TableName, wt_id);

        createDecisionRuleTableCombs(TableName, wt_id);
```

```
        generate_insertDecision_Combs(TableName, wt_id);
end;
```

Appendix N1: Partial dataset for 5 weeks for KRT database

```
insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,361,0.17,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,363,0.17,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,364,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,365,0.13,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,366,0.17,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,367,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,368,0.08,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,369,0.13,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,361,0.17,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,363,0.17,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,364,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,365,0.13,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,366,0.17,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,367,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,368,0.08,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,369,0.13,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,361,0.17,1.33,22,1);
```

```
insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,363,0.17,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,364,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,365,0.13,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,366,0.17,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,367,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,368,0.08,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,369,0.13,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,361,0.17,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,363,0.17,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,364,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,365,0.13,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,366,0.17,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,367,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,368,0.08,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,369,0.13,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,361,0.17,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,363,0.17,0.75,22,1);
```

```
insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,364,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,365,0.13,1.33,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,366,0.17,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,367,0.08,1,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,368,0.08,0.75,22,1);

insert into WT22_p_data(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,369,0.13,0.75,22,1);



insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,361,0.17,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,363,0.17,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,364,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,365,0.13,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,366,0.17,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,367,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,368,0.08,0.75,22,1);
```

```
insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,369,0.13,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,361,0.17,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,363,0.17,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,364,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,365,0.13,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,366,0.17,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,367,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,368,0.08,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,369,0.13,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,361,0.17,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,363,0.17,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,364,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,365,0.13,1.33,22,1);
```

```
insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,366,0.17,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,367,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,368,0.08,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,369,0.13,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,361,0.17,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,363,0.17,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,364,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,365,0.13,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,366,0.17,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,367,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,368,0.08,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,369,0.13,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,361,0.17,1.33,22,1);
```

```
insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,363,0.17,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,364,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,365,0.13,1.33,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,366,0.17,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,367,0.08,1,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,368,0.08,0.75,22,1);

insert into
WT22_p_data_MAX(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,369,0.13,0.75,22,1);



insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,361,0.17,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,363,0.17,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,364,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,365,0.13,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,366,0.17,1,22,1);
```

```
insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,367,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,368,0.08,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,1,369,0.13,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,361,0.17,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,363,0.17,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,364,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,365,0.13,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,366,0.17,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,367,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,368,0.08,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,2,369,0.13,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,361,0.17,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,363,0.17,0.75,22,1);
```

```
insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,364,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,365,0.13,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,366,0.17,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,367,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,368,0.08,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,3,369,0.13,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,361,0.17,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,363,0.17,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,364,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,365,0.13,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,366,0.17,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,367,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,368,0.08,0.75,22,1);
```

```
insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,4,369,0.13,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,361,0.17,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,363,0.17,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,364,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,365,0.13,1.33,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,366,0.17,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,367,0.08,1,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,368,0.08,0.75,22,1);

insert into
WT22_p_data_MIN(id,datasetID,faid,attributeweight,fuzzycatconstant,wtid,attributerate)
values(wt22_predictor_seq.nextval,5,369,0.13,0.75,22,1);
```

Appendix N2: partial dataset for FAOES_V2E for 9 months

```
insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,1,140,0.4,161674,3.2,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,1,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,1,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,2,140,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,2,141,0.4,161674,1,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,2,142,0.2,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,3,140,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,3,141,0.4,161674,3.5,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,3,142,0.2,161674,5.7,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,4,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,4,141,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,4,142,0.2,161674,1,16);
```

```
insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,5,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,5,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,5,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,6,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,6,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,6,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,7,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,7,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,7,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,8,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,8,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,8,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,9,140,0.4,161674,3,16);
```

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,9,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,9,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,1,140,0.4,161674,3.2,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,1,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,1,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,2,140,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,2,141,0.4,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,2,142,0.2,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,3,140,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,3,141,0.4,161674,3.5,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_sales_seq.nextval,3,142,0.2,161674,5.7,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,4,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,4,141,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,4,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,5,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,5,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,5,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,6,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,6,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,6,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,7,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,7,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,7,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,8,140,0.4,161674,3,16);

```
insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,8,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,8,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,9,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,9,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,
wtid) values(wt16_p_NET_sales_seq.nextval,9,142,0.2,161674,1,16);



insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,1,140,0.4,161674,3.2,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,1,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,1,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,2,140,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,2,141,0.4,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,2,142,0.2,161674,3,16);
```

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,3,140,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,3,141,0.4,161674,3.5,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,3,142,0.2,161674,5.7,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,4,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,4,141,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,4,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,5,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,5,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,5,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,6,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,6,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,6,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,7,140,0.4,161674,3,16);

```
insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,7,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,7,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,8,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,8,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,8,142,0.2,161674,1,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,9,140,0.4,161674,3,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,9,141,0.4,161674,0.06,16);

insert into
WT16_p_data_Net_sales(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid)
values(wt16_p_NET_sales_seq.nextval,9,142,0.2,161674,1,16);




insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,1,140,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,1,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,1,142,0.2,0.86,1,16);
```

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,2,140,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,2,141,0.4,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,2,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,3,140,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,3,141,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,3,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,4,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,4,141,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,4,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,5,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,5,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,5,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,6,140,0.4,0.86,3,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,6,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,6,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,7,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,7,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,7,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,8,140,0.4,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,8,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,8,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,9,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,9,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,9,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_orders_seq.nextval,1,140,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,1,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,1,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,2,140,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,2,141,0.4,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,2,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,3,140,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,3,141,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,3,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,4,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,4,141,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,4,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,5,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,5,141,0.4,0.86,0.38,16);

```
insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,5,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,6,140,0.4,0.86,3,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,6,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,6,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,7,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,7,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,7,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,8,140,0.4,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,8,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,8,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,9,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,9,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstan
t,wtid) values(wt16_p_NET_orders_seq.nextval,9,142,0.2,0.86,1,16);
```

```
insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,1,140,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,1,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,1,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,2,140,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,2,141,0.4,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,2,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,3,140,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,3,141,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,3,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,4,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,4,141,0.4,0.86,3.9,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,4,142,0.2,0.86,1,16);
```

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,5,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,5,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,5,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,6,140,0.4,0.86,3,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,6,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,6,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,7,140,0.4,0.86,2,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,7,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,7,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,8,140,0.4,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,8,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,8,142,0.2,0.86,1,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,9,140,0.4,0.86,2,16);

```
insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,9,141,0.4,0.86,0.38,16);

insert into
WT16_p_data_Net_orders(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid
) values(wt16_p_NET_orders_seq.nextval,9,142,0.2,0.86,1,16);




insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,1,140,0.4,164.9,5.9,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,1,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,1,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,2,140,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,2,141,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,2,142,0.2,164.9,3,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,3,140,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,3,141,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,3,142,0.2,164.9,1,16);
```

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,4,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,4,141,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,4,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,5,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,5,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,5,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,6,140,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,6,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,6,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,7,140,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,7,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,7,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,8,140,0.4,164.9,5.9,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,8,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,8,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,9,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,9,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MAX(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,9,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,1,140,0.4,164.9,5.9,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,1,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,1,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,2,140,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,2,141,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,2,142,0.2,164.9,3,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,3,140,0.4,164.9,0.15,16);

```
insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,3,141,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,3,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,4,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,4,141,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,4,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,5,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,5,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,5,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,6,140,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,6,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,6,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,7,140,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconst
ant,wtid) values(wt16_p_NET_products_seq.nextval,7,141,0.4,164.9,0.15,16);
```

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,7,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,8,140,0.4,164.9,5.9,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,8,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,8,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,9,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,9,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products_MIN(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,9,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,1,140,0.4,164.9,5.9,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,1,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,1,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,2,140,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,wtid) values(wt16_p_NET_products_seq.nextval,2,141,0.4,164.9,1,16);

```
insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,2,142,0.2,164.9,3,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,3,140,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,3,141,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,3,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,4,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,4,141,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,4,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,5,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,5,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,5,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,6,140,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,6,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,6,142,0.2,164.9,1,16);
```

```
insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,7,140,0.4,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,7,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,7,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,8,140,0.4,164.9,5.9,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,8,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,8,142,0.2,164.9,1,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,9,140,0.4,164.9,3,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,9,141,0.4,164.9,0.15,16);

insert into
WT16_p_data_Net_products(id,datasetID,faid,attributeweight,attributerate,fuzzycatconstant,w
tid) values(wt16_p_NET_products_seq.nextval,9,142,0.2,164.9,1,16);
```