2011

Determining a Patient Recovery from a Total Knee Replacement Using Fuzzy Logic and Active Databases

Robert Azarbod
Minnesota State University, Mankato

Determining a Patient Recovery from a Total Knee Replacement

Using Fuzzy Logic and Active Databases

By

Robert Cameron Azarbod

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

In

Information Technology

Minnesota State University, Mankato

Mankato, Minnesota

May 2011

Determining a Patient Recovery from a Total Knee Replacement Using Fuzzy Logic and Active Databases

Robert Azarbod

This thesis has been examined and approved by the following members of the thesis committee.

Professor at Minnesota State University, Mankato in the Department of Information Technology - Dr. Mahbubur Syed, Advisor

Professor at Minnesota State University, Mankato in the Department of Computer Science - Dr. Hamed Sallam

Professor at Minnesota State University, Mankato in the Department of Information Technology - Dr. Cyrus Azarbod

Acknowledgements

This report is submitted as part of the required work in the course IT 699 (Department of Information System and Technology, 1 credit, and Thesis) at Minnesota State University, Mankato, and has been supervised, examined, and accepted by Professors Dr. Mahbubur Syed (Paper Advisor), Dr. Hamed Sallam (Committee Member), and Dr. Cyrus Azarbod (Committee Member).

I would also like to acknowledge Dr. Mahbubur Syed for his valuable advice and guidance throughout this research process.

I would also like to thank Dr. Hamed Sallam and Dr. Cyrus Azarbod for their continuous support through the research process and all their valuable input along the way.

I would like to acknowledge the Database Research Groups at both Minnesota State University, Mankato and from the American University of Armenia, especially Annie Hovian, Anagha Bankar, Haritha Yanala, and Saro DerVanasian

Under the Thesis option for the Master of Science, this report is offered in lieu of a thesis.

Abstract

The purpose of the knowledge-based system is to predict the rehabilitation timeline of a patient in physical therapy for a total knee replacement. All patients have various attributes that contribute to their rehabilitation rate such as: weight, gender, smoking habit, medications, physical ability, or other medical problems. A combination of any one or several of these attributes will affect the recovery process. The proposed FRTP (Fuzzy Rehabilitation Timeline Predictor) is a fuzzy data mining model that can predict the recovery length of a patient in physical therapy for a total knee replacement and provide feedback to experts for revision of the physical therapy plans to meet the recovery goal. Using the FRTP, an approximate timeline for a patient can be predicted, thus creating more insight into the healing process.

The process of analyzing patient data, predicting the number of weeks for the maximum healing result, adaptation of a different recovery plan based on our research prototype using fuzzy logic in database systems to maximize the recovery period, is a very interesting and important component for the patient, health insurance companies, medical clinics, and physicians. This research paper presents a methodology to analyze and mine the data using a web based application (Web Fuzzy Data Mining) and fuzzy calculus to perform data mining and predicting the best possible plan for a faster recovery.

Table of Contents

List of Tables

List of Figures

List of Equations

Chapter 1

Introduction

1.1 Thesis Introduction

Many real business, government, and education databases deal with a large amount of data. This large volume and complexity make it difficult for management to make a timely decision on some important issues such as employee performance, student assessment, customer/student retention, or patient recovery.

One method to handle this issue is by using fuzzy logic techniques in database applications. Fuzzy logic can help in handling uncertain or incomplete information. Combining databases and fuzzy technologies creates “Fuzzy databases” [ref-Omron]. Fuzzy techniques have been applied in many aspects of relational databases such as: representing and querying fuzzy data [Hsieh 2005, Galindo et. al. 2003], knowledge discovery from databases [Maddouri, Elloumi, & Jaoua, 1998], and modeling uncertainty at the conceptual schema level [Chaudhry et. al., 1999].

There has not been a lot of research done in generating dynamic fuzzy tables. At Minnesota State University, Mankato and American University of Armenia, extensive research has been conducted in the area of fuzzy logic and intelligent databases/active databases.

In this research, we have developed a database based on a patient undergoing knee replacement surgery. This database is used to train the WebFDM [Anagha 2010] fuzzy database engine and predict patient recovery. In this research, we have also proposed a methodology based on fuzzy mathematical modeling to predict a patient's recovery based on incomplete data.

1.2 Thesis Statement

To develop a database based on a patient undergoing knee replacement surgery and use it to train the WebFDM [Anagha 2010] fuzzy database engine and predict patient recovery. Also, a methodology based on fuzzy mathematical modeling to predict patient recovery from incomplete data has been proposed.

1.3 Thesis Outline

Chapter 2 presents the background information for the thesis with a detailed description of fuzzy logic, relational databases and data warehousing, data mining, materialized view, procedures, and triggers. Chapter 3 provides the basis for the thesis by describing research projects including the working of Web-FDM done by a database research group at MSU Mankato and AUA (American University of Armenia). Chapter 4 explains the structure of the KRT (Knee Replacement Therapy) database, thesis components and its architecture. Chapter 5 focuses on implementation and analysis of the KRT database. Chapter 6 shows concepts, methodology and the examples of using fuzzy calculus to predict the patient recovery based on incomplete data and WebFDM learned modules of the KRT database. Chapter 7 concludes the research and Chapter 8 presents suggestions for further research.

Chapter 2

Literature Review

2.1 Total or Partial Knee Replacements

Total or partial knee replacements are becoming a very frequent operation in today's society. As stated by Robyn Stein and Caitlin Hool, "The number of total knee replacements performed in the United States has increased dramatically since 1990; currently 581,000 such procedures are performed every year. This number is expected to increase markedly as Baby Boomers age." [Stein, Hool, 2009] The reason for these knee replacements was "the recipients of total knee replacements experienced significant improvement in function, including a 17.5% increase in mobility, a 39.3% improvement in motor skills; and a 46.9% decrease in limitations in activities of daily living such as bathing and dressing oneself." [Stein, Hool, 2009]

Since there has been an increase in the number of total knee replacements, there has been a consequent increase in the number of rehabilitations from this surgery. As stated from Mayo Clinic, “In the hospital, patients receive physical therapy to help them adjust to the prosthesis. Patients also learn postoperative exercises to perform when they return home.” [Mayo Clinic, 2010] Therefore, seeing how a knee replacement acts over time of rehabilitation (whether physical therapy is used or not) would suggest if there is a better process for patients to follow during the rehabilitation phase after a knee replacement.

Furthermore, there are multiple designs or types of knee replacements used which vary depending on the needs of the patient and the recommendation of the doctor. As stated by Dr. H. D. Huddleston, "There are many designs of knee implants available to the surgeon. There is no universal agreement as to which design is best. Each surgeon selects what he believes is best, or what he was trained to use."[Bonesmart, 2009] In addition, "The most important consideration is that your surgeon should be totally comfortable and familiar with the surgical technique for installation of the implant selected. Each type has unique surgical aspects and considerations which can only be learned by experience with many cases."[Bonesmart, 2009] Therefore, measuring how a knee replacement acts over time of rehabilitation will illustrate a patient's progress during recovery, and assist recipients in their recovery process.

2.2 Fuzzy Logic

The Fuzzy Logic concept was introduced by Zadeh (1973), a professor at the University of California at Berkley, as a way of processing the data in linguistic variables. This concept has been widely used in various fields like control systems, databases, and other related concepts. As the concept of fuzzy logic was evolving, it was used in mechanical systems, electrical systems, or for industrial use. Today there are fuzzy microwaves or fuzzy cars [Naranjo, Gonzalez, Garcia, Pedro, & Sotelo, 2006].

An example of past research [Skarlatos, Karakasis & Trochidis, 2004] has shown the use of fuzzy logic to diagnose the railway wheel defects. This method is based on vibration measurements at different train speeds on healthy wheels versus pre-examined defective wheels. The fuzzy logic model stores the obtained experience of various measurements including vibrations, train speed and frequency in the database and makes a decision on the extent of the damage and the need for preventative maintenance [Skarlatos, Karakasis & Trochidis, 2004].

Fuzzy logic is an important technique for incorporating linguistic variables with numerical data and for interpretation of the user's choice in a qualitative manner. Currently, fuzzy logic is used in database systems, however, fuzzy logic can be used to retrieve the documents from such complex databases based not only on the contents of documents, but also from the idea the user has of their appearance, through queries specified in terms of the user's criteria.

2.3 Relational Databases

A relational database is a collection of data items which is organized by means of related fields. Data items are a set of formally-described tables from which data can be accessed or reassembled in different ways without having to reorganize the database tables. It is based on the relational model wherein tables can be linked to each other. The fields in the table can be of many different types, which may vary with each DBMS, such as: Oracle, MySQL, and SQL. Generally, there are three kinds of fields: character, numeric, and date. These fields may contain NULL values, which means 'not defined' or 'nothing at all'. To avoid the complexity in calculations and to maintain data accuracy, many fields are set to non-null values. The records in the table are accessed using a key that uniquely identifies each record of the table and it is called the primary key (PK). The table can also have a Foreign Key (FK) which is a regular attribute in one table, but a primary key in another table. An index is a physical way to improve the performance of a database. Indexes are strictly parts of a physical structure while keys are parts of a logical structure. [Connolly & Begg, 2005]. Additionally, there are three types of relationships can be defined on database tables – one-to-one, one-to-many and many-to-many.

Data integrity describes accuracy, validity, and consistency of data. The technique used to properly define data integrity is called Database Normalization which helps to reduce inconsistency of data and poor data integrity. Another important concept is 'Referential Integrity' which ensures that the relationships among tables in a database remain consistent. When one table has a foreign key to another table, the concept of referential integrity states that you may not add a record to the table that contains the foreign key if there is a corresponding record in the linked table. [Connolly & Begg, 2005].

2.4 Materialized View

The relational database contains an object called a view or a materialized view. A view is a virtual table created using a subset of actual tables which display dynamically a query on created records by operating one or more relational tables. Updates to a base table get reflected in a view. A materialized view takes a different approach in which the query result is cached as a concrete table that may be updated from the original base table from time to time. Materialized views are created to improve query response time. As the data is pre-computed, materialized views allow fast and accurate retrieval of information. Mostly, views are used to reduce the network loads, to distribute the corporate database to regional sites, to create mass deployment, to enable data sub-setting and, finally, to enable disconnected computing. Materialized views are updated through an efficient batch process from a single master site or master materialized view site. Deployment templates help to create a materialized view environment locally. Materialized views allow replicating data based on column-level and row-level sub-setting. Data sub-setting enables replicating the information that pertains only to a particular site. Materialized views do not require a dedicated network connection. Refreshing the data in materialized views can be done by scheduling the job or by manually refreshing as required. Materialized views commonly created for data warehousing are aggregate views, single-table aggregate views, and join views.[Connolly & Begg, 2005].

For data warehousing purposes, the materialized views commonly created are aggregate views, single-table aggregate views, and join views. Use of materialized views are beneficial when the user runs the same query a multiple of times. When a similar query is given, Query Rewrite mechanism automatically rewrites the query to use the materialized view. In this mechanism, the Database Administrator (DBA) may not know in advance what query needs to run to use the materialized view. Moreover, in data warehouses a new query has to be executed. This is the time when the query rewrite mechanism uses the query even if only part of the query can be satisfied using the materialized view. Query Rewrite takes place when the query exactly matches with the materialized view. Sometimes, the column which the query refers to may not be in the materialized view (Summary Join back). When the query request aggregates at a higher level, summary rollup and aggregation to all occurs. [Connolly & Begg, 2005].

2.5 Active Database

An active database system improves traditional database functionality, where a pattern of data in the database invokes an action (rule). An active database incorporates the event monitoring scheme for detecting when certain data is inserted or updated and automatically executes the actions in response to certain events that happen and when particular conditions are met. This project implements procedures, functions and triggers to incorporate the event monitoring of the fuzzy data.

The active rules are mostly in the form of Even Condition Action (ECA) rules which has the capability to process the rules and to provide a uniform and efficient mechanism for database system applications. Applications include integrity constraints, views, authorization, knowledge-based systems, and workflow management. Events in the ECA model trigger the rules, which include database update operations (Insert, Update, and Delete) that are explicitly applied to the database. Generally, these functions can be temporal or external events. The ECA model condition determines which rule action needs to be executed. When an event occurs, an optional condition might be evaluated and if no condition is specified, the action will be executed immediately after an event occurrence. If a condition is specified, it is first evaluated and if it returns true, then an action will be executed. The ECA model mostly consists of SQL statements but it can also be an external program or a database transaction. [Widom, & Seri, 1996].

2.6 Fuzzy Database Research

Fuzzy logic techniques have been introduced and utilized in database applications. Introducing fuzzy logic technologies to the database fields help in handling uncertain or incomplete information. Combining database and fuzzy technologies create

"Fuzzy databases" [ref-Omron]. Fuzzy techniques have been applied in many aspects of relational databases such as: representing and querying fuzzy data [Hsieh 2005, Galindo et. al. 2003], knowledge discovery from databases [Maddouri, Elloumi, & Jaoua, 1998], and modeling uncertainty at the conceptual schema level. [Chaudhry et. al., 1999]

Not a lot of research has been done in generating dynamic fuzzy tables. At Minnesota State University, Mankato and American University of Armenia, extensive research has been conducted in the area of fuzzy logic and intelligent databases/active databases. Extensive research has also been done at Minnesota State University, Mankato in regard to fuzzy databases. In the following paragraphs, different research projects related to fuzzy databases relevant to this project will be described.

2.6.1 FAOES (Fuzzy Active Oder Entry System)

Many corporations and businesses rely on relational databases for their daily activities and decision-making processes. Employee performance evaluations are considered a very time consuming task faced by many large business mangers especially if it involves a huge amount of transaction sales. In the FAOES project, the intention was to expand an existing relational database to support an active database as well as fuzzy logic features. The extended fuzzy active database was automatically built based on the methodology that is presented in the following paragraph. The database used for this project is OES (Order Entry System) that is developed at MSU-Mankato and the data model can be seen in Appendix A1. The proposed methodology involved building a fuzzy set theory on top of a relational database to automatically generate a series of fuzzy tables. These fuzzy tables were queried for employee performance using SQL (Structured Query Language).

The proposed methodology goes through three phases: Initial phase, Learning phase and Mining phase. The following shows all of the steps for these three phases.

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

Fuzzy attributes in this project were selected as sales, orders and products. Fuzzy categories for each attribute were selected as poor, below average, average, above average and excellent. Membership values were defined for triangular and trapezoidal functions based on the fuzzy attributes and fuzzy categories (Appendix A2). Sample rules were used for this project are shown in Table 1.

Table 1: Sample Rules Table

RULE_SET	RULE_NUM	STATUS	ORDERS	SOLD_PRODUCT	SALES	DECISION
2	89	1	BELOW_AVG	BELOW_AVG	ABOVE_AVG	Give Warning to Employee
2	90	1	AVG	ABOVE_AVG	EXCELLENT	Give Raise to Employee
2	91	1	POOR	BELOW_AVG	ABOVE_AVG	Give Raise to Employee
2	92	1	BELOW_AVG	BELOW_AVG	EXCELLENT	Give Raise to Employee
2	93	1	BELOW_AVG	AVG	EXCELLENT	Give Raise to Employee
2	94	1	BELOW_AVG	POOR	EXCELLENT	Give Warning to Employee
2	95	1	ABOVE_AVG	EXCELLENT	BELOW_AVG	Give Raise and Gift to Employee

FAOES has been implemented in Oracle 9i DBMS using PL/SQL and SQL languages. FAOES uses a series of stored procedures, user defined functions, triggers, and materialized views to extend the OES database to provide a fuzzy analysis. The fuzzy components can be found in appendix A3. The figure 9 shows a simple flowchart for implementing FAOES [Haritha 2010].

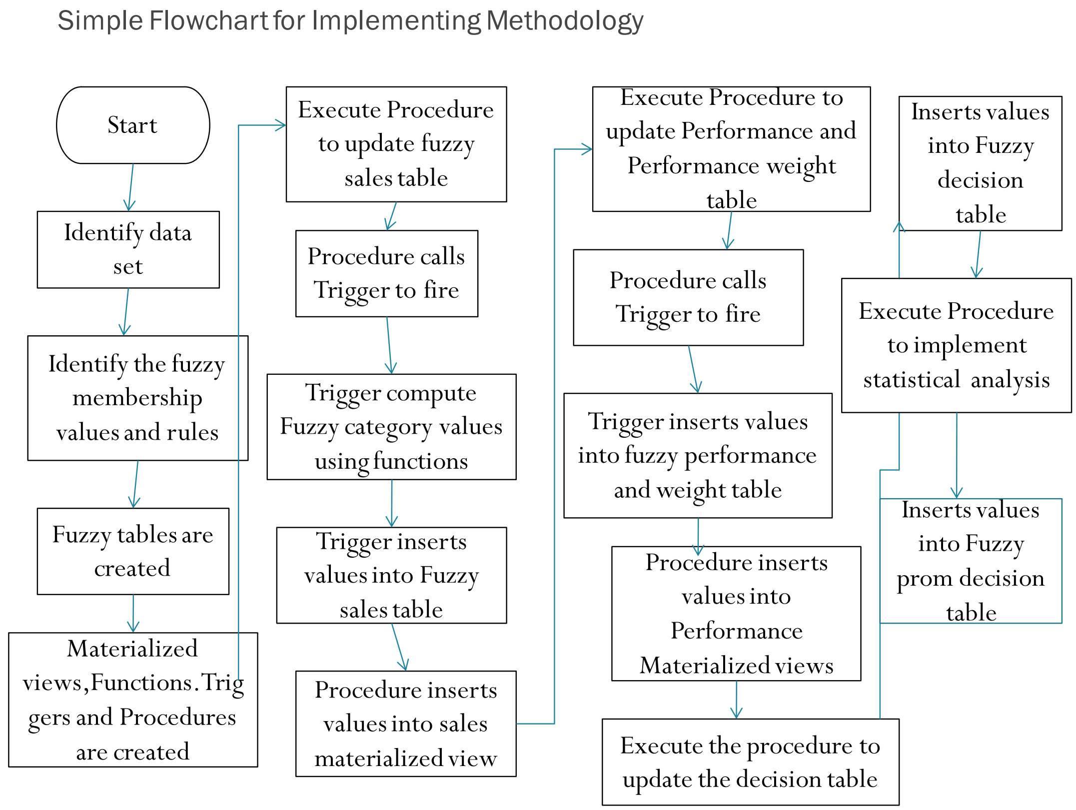


Figure 1: FAOES Flowchart

Successful implementation of the proposed methodology generated a fuzzy table for each fuzzy attribute, two performance tables that combine the fuzzy tables into one, and one final decision table. The sample tables are shown in figures 12, 13, 14, and 15.

Table 2: Employee Fuzzy Sales Table

EMPLOYEE_NO	LNAME	FNAME	REPORT_DA	SALES_TOTAL	POOR	BELOW_AVERAGE	AVERAGE	ABOVE_AVERAGE	EXCELLENT
1006	Thomas	Peter	05-DEC-07	209500	.9162	0	0	0	0
1008	Stone	James F.	05-DEC-07	2867955	0	.6399625	0	0	0
1010	Stansbury	Thomas	05-DEC-07	1520840	.391664	0	0	0	0
1011	Stansbury	Steward	05-DEC-07	7207330	0	0	0	.8248455	0

Table 3: Sample Employee Fuzzy Performance Table

EMPLOYEE_NO	LNAME	FNAME	REPORT_DA	NUMBER_ORDERS	NUMBER_PRODUCT	SALES_TOTAL
1006	Thomas	Peter	13-DEC-09	Below Average	Below Average	poor
1008	Stone	James F.	13-DEC-09	Below Average	poor	Below Average
1010	Stansbury	Thomas	13-DEC-09	Below Average	poor	poor
1011	Stansbury	Steward	13-DEC-09	Above Average	Below Average	Above Average

Table 4: Sample Employee Performance Weight Table

EMPLOYEE_NO	LNAME	FNAME	REPORT_DA	NUMBER_ORDERS	ORDERS_WEIGHT	NUMBER_PRODUCT	PRODUCTS_WEIGHT	SALES_TOTAL	SALES_WEIGHT
1005	Wehland	William C.	13-DEC-09	Average	.5714286	Above Average	.8441667	poor	.53103
1006	Thomas	Peter	13-DEC-09	Below Average	.7142857	Below Average	.4916667	poor	.9162
1008	Stone	James F.	13-DEC-09	Below Average	.5714286	poor	.46725	Below Average	.6399625
1010	Stansbury	Thomas	13-DEC-09	Below Average	.5714286	poor	.71425	poor	.391664
1011	Stansbury	Steward	13-DEC-09	Above Average	.5714286	Below Average	.5883333	Above Average	.8248455

Table 5: Sample Employee Decision Table

EMPLOYEE_NO	FUZZY_MEM_SET	DEC_SET_RULE	LNAME	FNAME	REPORT_DA	DECISION_NET
1088	101	1	Hanzdo	Lee	13-DEC-09	NO DECISION
1088	101	2	Hanzdo	Lee	13-DEC-09	Give Warning to Employee
1088	102	1	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	102	2	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	103	1	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	103	2	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	201	1	Hanzdo	Lee	13-DEC-09	NO DECISION
1088	201	2	Hanzdo	Lee	13-DEC-09	Give Warning to Employee
1088	202	1	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	202	2	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	203	1	Hanzdo	Lee	13-DEC-09	Give Raise to Employee
1088	203	2	Hanzdo	Lee	13-DEC-09	Give Raise to Employee

During the learning phase, 16 different database schemas were developed to store different experiments. These schemas were analyzed to identify which schema represents the best result and that schema was used in the mining phase. The learning process led to altering the programs, fuzzy tables, fuzzy components, data sets and decision rules. Some analysis of different schemas developed during the learning phase is presented [Haritha 2009] in Figure 2.

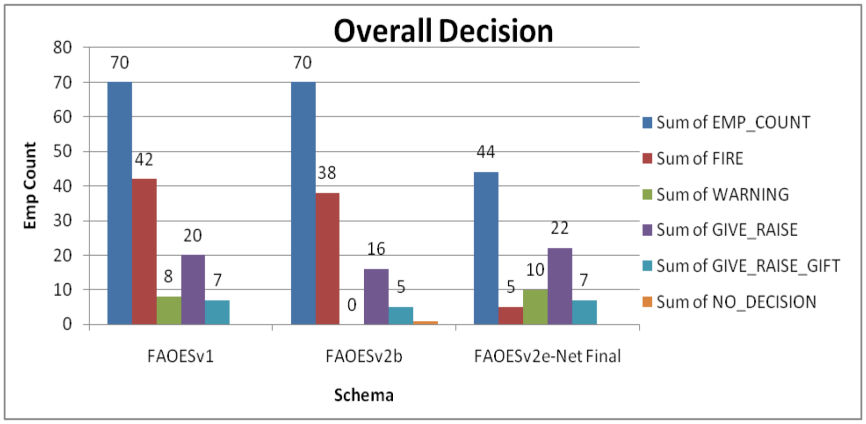


Figure 2: Comparisons of Different Schema in FAOES

In the mining phase, sales information about a new employee was processed against the optimum schema (the learned schema) and the result in comparison to (the other schemas) other schema is presented in Table 6.

Table 6: Comparisons of Different Schema in FAOES

EMPLOYEE_NO	LNAME	FNAME	DECISION_NET	SCHEMA
1002	Worral	Al	Fire Employee	FAOES_v2b
1002	Worral	Al	Give Raise to Employee	FAOES_v2e
1002	Worral	Al	Give Raise and Gift to Employee	FAOES_v2l
1002	Worral	Al	Fire	FAOES_v3a
1002	Worral	Al	5% raise	FAOES_v3b
1002	Worral	Al	7% raise with $500 gift certificate	FAOES_v3c

2.6.2 ARDIF (Automatic Extension of Relational Database to Incorporate Fuzzy Logic)

One of the main limitations of FAOES was that all fuzzy components were specifically custom built for the OES database so, it could not be used for other databases. Therefore, applying the FAOES solution to any new database required a substantial amount of time and programming. ARDIF was developed to overcome this limitation [Deravanesian 2007]. ARDIF accepts any table or view (collection of joined tables) and generates a dynamic series of fuzzy tables, fuzzy performance tables, and fuzzy decision tables. ARDIF was developed and implemented using Oracle 9i DBMS and PL/SQL and SQL programming languages. ARDIF accepts a series of meta-data information about a database from a user. These meta-data are a work table (data sets to be analyzed which is represented in the user production database as a view), fuzzy attributes, fuzzy categories, membership values for triangle and trapezoidal graphs, and decision rules. After collecting all this information, the engine, fuzzy tables, materialized views, triggers, stored procedures, and functions are created by querying these meta-data inserted by the user of the application. [Deravanesian 2007]. For every database, ARDIF creates a subprogram that calculates the fuzzy category weights for each record of the worktable and fills each fuzzy table. From all of the fuzzy tables, the subprogram creates and populates two performance tables (one with weights and one without weights) and then according to rules specified by the user creates the decision table. At the end, for each database project, the decision table is kept and all other temporary tables (fuzzy and performance tables) are removed. ARDIF meta data is presented in figure 4.

The components of ARDIF project:

The figure 3 shows the components that are in the form of stored procedures, triggers, functions, and materialized views.

- 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 3: ARDIF Program Components

Metadata Components

The metadata developed for this project is presented in Figure 4.

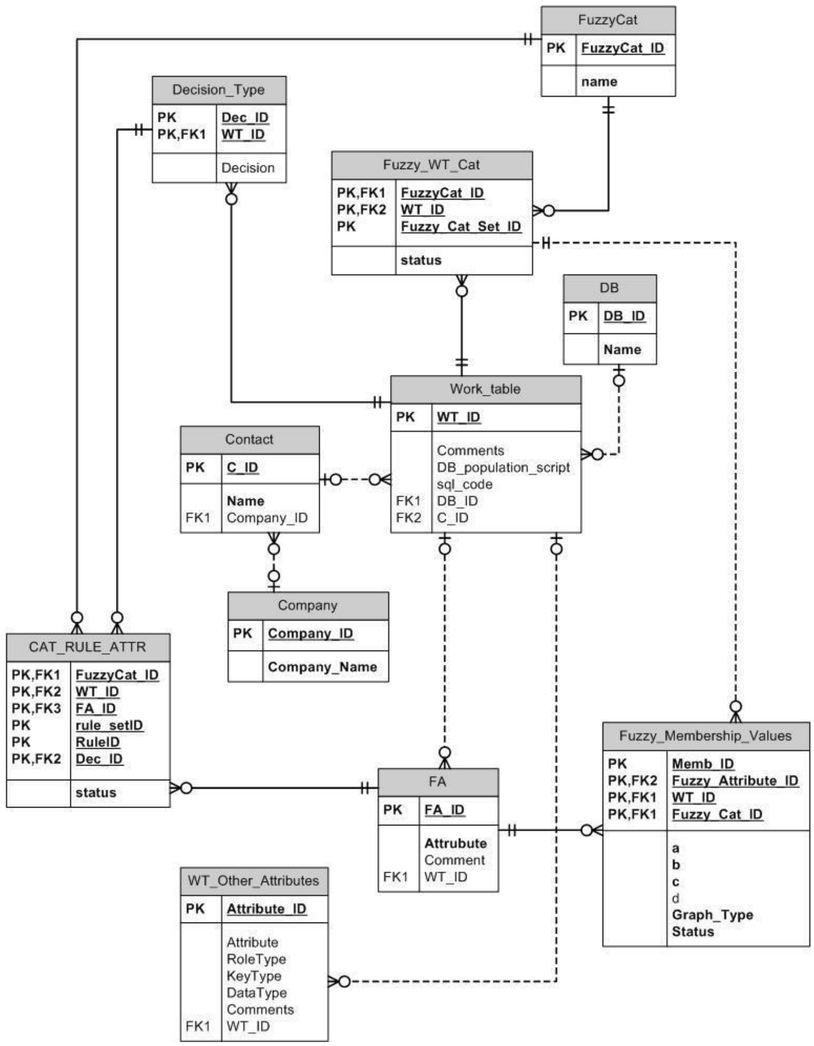


Figure 4: ARDIF Metadata

ARDIF's proposed methodology:

ARDIF's proposed methodology is similar to the initial phase of the FAOES project. It consists of defining fuzzy attributes, identifying fuzzy categories, identifying membership values for fuzzy categories, identifying fuzzy rules, creating and populating each fuzzy table, creating and populating performance tables, creating and populating fuzzy decision tables. In addition to the above components, it also collects information about the database, database owner, company, and contact. The methodology [Bankar 2010] is presented in Figure 5.

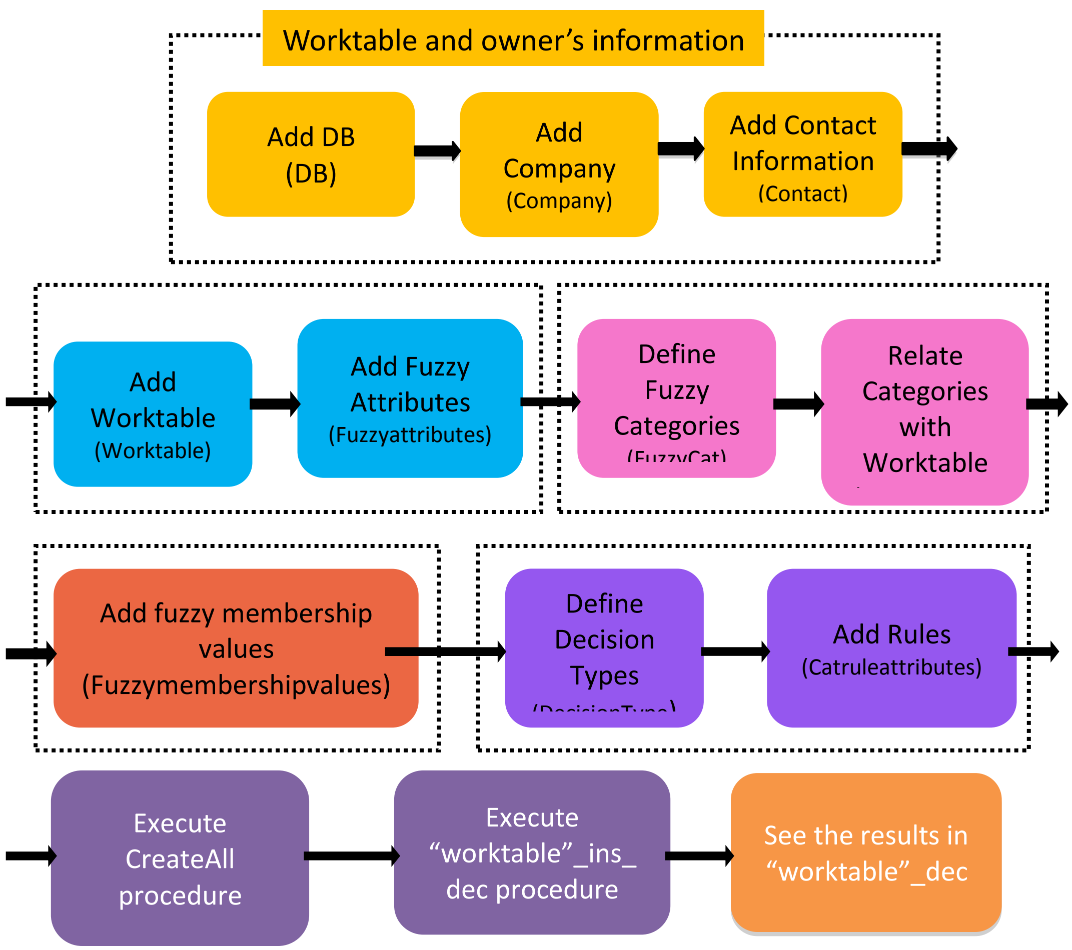


Figure 5: ARDIF Methodology

Implementation of OES using ARDIF:

ARDIF was successful in generating similar results that the FAOES version one had created. Table 7 shows tables that ARDIF generated:

Table 7: Tables Generated by ARDIF for the FAOES

Name	***Use***
WT1_ID4_C1	The main work table for OES database, contains the detail about each employee
WT1_ID4_C1_DEC_RUL	Decision rules provided by the manager
WT1_ID4_C1_SALES_TRI WT1_ID4_C1_SALES_TRP	Category of total sales calculated for each employee using 2 functions
WT1_ID4_C1_PRODUCTS_TRI WT1_ID4_C1_PRODUCTS_TRP	Category of total sold products calculated for each employee
WT1_ID4_C1_ORDERS_TRI WT1_ID4_C1_ORDERS_TRP	Category of total orders calculated for each employee
WT1_ID4_C1_WEIGHT WT1_ID4_C1_PERF WT1_ID4_C1_DEC	Weight, performance and decisions calculated for each employee according to each fuzzy table

The following SQL code will generate a result that is shown in Table 8. This result is a partial list of decision data produced by both FAOES-V1 and ARDIF-FAOES. For complete tables, refer to appendix B1.

```
select distinct id, employee_no, lname,graph, Dec_tri Fuzzy_Decision
from wt1_decall w join faoes_v1.emp_fuzzy_decision f
            on w.id=f.employee_no
where graph='TRI';
```

Table 8: Query Demonstrating Partial Decisions Data Produced by both FAOES-V1 and ARDIF-FAOES

ID	EMPLOYEE_NO	LNAME	GRAPH	FUZZY_DECISION
1000	1000	Wyatt	TRI	fire
1001	1001	Wright	TRI	fire
1002	1002	Worral	TRI	fire
1003	1003	Wooton	TRI	give warning
1004	1004	Widdes	TRI	fire
1005	1005	Wehland	TRI	gift and raise
1005	1005	Wehland	TRI	no decision
1006	1006	Thomas	TRI	fire
1006	1006	Thomas	TRI	no decision
. . .				

ARDIF has the following limitations:

- ARDIF only supports the initial phase of FDM (Fuzzy Database Mining) methodology proposed by FAOES
- ARDIF removes most of the fuzzy tables and only holds the decision table. This eliminates the possibility of comparing different strategies during the learning phase.
- Users of ARDIF should be knowledgeable database programmers in order to successfully interact with ARDIF
- ARDIF does not support user interface to simplify the user access
- ARDIF does not utilize any visualization or OLAP to further learn about the database

2.6.3 ARDIF$_2$ – An Extension of ARDIF

ARDIF$_2$ is the enhanced version of ARDIF which saves the decisions generated for multiple fuzzy sets as well as rule sets so, that the user will be able to compare the results with different membership and rule sets in order to determine the best among it and finalize the decision [Bankar 2010].

ARDIF$_2$ saves different fuzzy decisions results from multiple rule set and different membership values for triangle and trapezoidal graphs, and new or modified data sets. ARDIF$_2$ took care of some of the limitations of ARDIF but not all [Bankar 2010]. The remaining limitations are:

- Difficult to maintain the database and add new Users of ARDIF$_2$
- Should be a knowledgeable database programmer
- No support for different types of users
- No support for visualization or OLAP access
- Lack of user interface to interact with ARDIF$_2$

2.6.4 WebFDM (Web Based Fuzzy Data Mining)

The major motivation behind development of WebFDM was to overcome the limitation of ARDF$_I$ and its successor ARDIF$_2$ projects. WebFDM provided a solution to the following issues [Bankar 2010]:

- Supported the methodology discussed in FDM/FAOES versions: Initial Phase, Learning Phase and Mining Phase

- Through repetitive structure of learning phase, WebFDM can identify the best settings of fuzzy components (data set, membership values for triangle and trapezoidal graph, and rule sets)
- Web user interface for easy interaction
- Web interface enabling all types of users to effectively define the initial setting, learn from it and teach the fuzzy engine (learning phase) and then mine the optimal results
- WebFDM provides Visualization as a complement to textual and numerical results to provide for better understanding of the data pattern
- WebFDM provides support for different types of users

WebFDM Architecture:

WebFDM uses $ARDIF_2$ database implemented in Oracle9i. The web interface programs uses C# programming as the main development tool and it also uses Data Access Layer (DAL) as a middle-ware to add, update, and select data from database. In case of complex retrieval queries, it bypasses DAL and directly interacts with the database. TierDeveloper 6.1 software is used to generate Data Access Layer. TierDeveloper from Alchisoft was used because it is easily available and works well with the Oracle database. It maps Oracle tables into objects, procedures into functions, and speeds-up the .NET development process. The visualization component uses the Microsoft's charting control (asp.net charting control). For publishing the website on the internet, WebFDM uses Microsoft Internet Information Services 6. IIS6 is part of a group of Internet servers (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft's Windows operating systems. WebFDM develops a server-side internet application in C# which is supported by IIS. IIS applications allow reusing the components and separates the code from HTML, and also helps to streamline the process (no need to create the HTML template files that applications send to the browser) [Bankar 2010]. Figure 6 represents the WebFDM architecture.

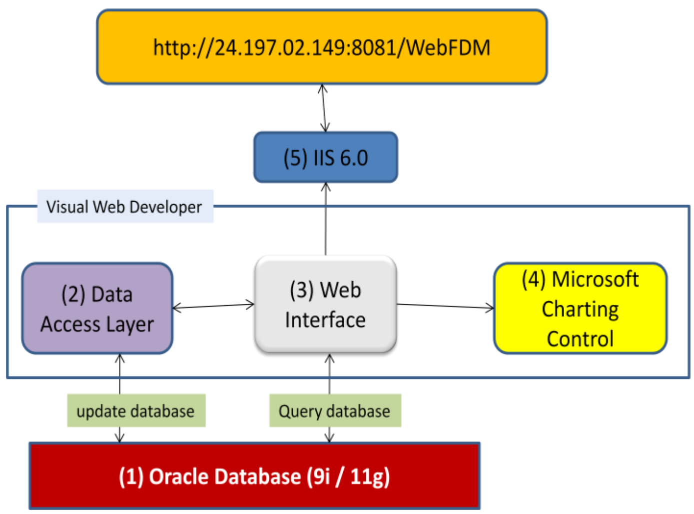


Figure 6: WebFDM Architecture

WebFDM is similar to FAOES because both processes follow a similar model: initial implementation, learning phase, and mining phase. WebFDM expanded $ARDIF_2$ programs, which supports all three phases and uses that as the backend for the web interface. Figure 7 shows the detailed steps involved in each phase.

Initial phase Step 1: **Database info page** (Set-up screens are available in appendix C1)
- Add database
- Add company name
- Add contact name

Initial phase Step 2: **Data set and attributes page** (Set-up screens are available in appendix C2)
- Add dataset name
- Add dataset
- Add Other Attributes and Fuzzy Attributes

Initial phase Step 3: **Fuzzy categories page** (Set-up screens are available in appendix C3)
- Add fuzzy categories
- Relate categories to the dataset

Initial phase Step 4: **Fuzzy category membership page** (Set-up screens are available in appendix C4)
- Selection of dataset, fuzzy attribute and rounding value
- Auto-generating membership values

Initial phase Step 5: **Decision set up page** (Set-up screens are available in appendix C5)
- Add decision types
- Create decision set

Initial phase Step 6: **Generate and report pages** (Set-up screens are available in appendix C6)
- Pre-analysis stage
- Execute the system
- Summary performance report
- Overall performance report

Learning phase - Step 7: **Analysis page** (Set-up screens are available in appendix C7)
- Customized toolbox to analyze the datasets
- Overall analysis
- Analysis based on other attributes
- Detail graphical analysis

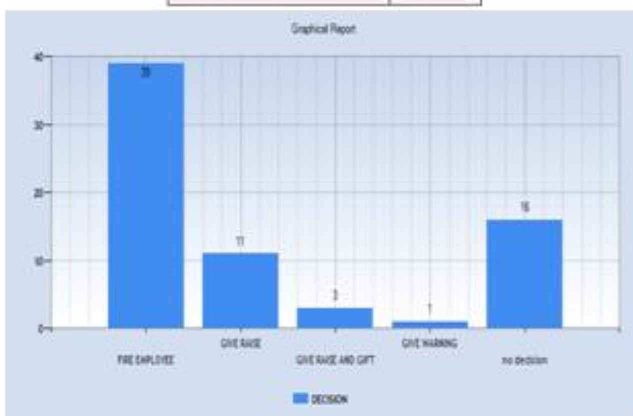
Learning phase Step 8: **Revision page** (Set-up screens are available in appendix C8)
- Modify decision rules
- Modify membership values
- Modify Fuzzy categories
- Modify Fuzzy attributes
- Modify dataset
- Select the most optimum data set and fuzzy setting

Mining phase Step 9: **Mining pages** (Set-up screens are available in appendix C8)
- Select the Dataset (allows system to retrieve the latest optimum fuzzy settings)
- Add new data to be analyzed
- Process data
- Evaluate result

Figure 7: Detailed Processing Step in WebFDM

WebFDM has consistently produced similar results as FDM/FAOES. Figures 7 and 8 shows these results.

DEC_TRI	COUNT
FIRE EMPLOYEE	39
GIVE RAISE	11
GIVE RAISE AND GIFT	3
GIVE WARNING	1
no decision	16



Query 1 Result using Web-FDM

Result generated by FDM/FAOES:

REPORT_DA	FUZZY_MEM_SET	DEC_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 8: Tabular result and graphical result with WebFDM and tabular result with FDM/FAOES

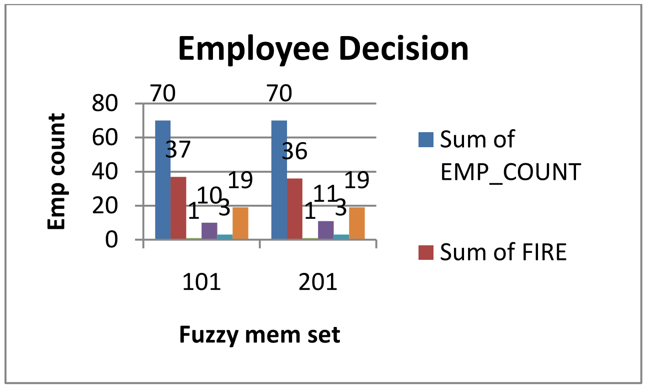
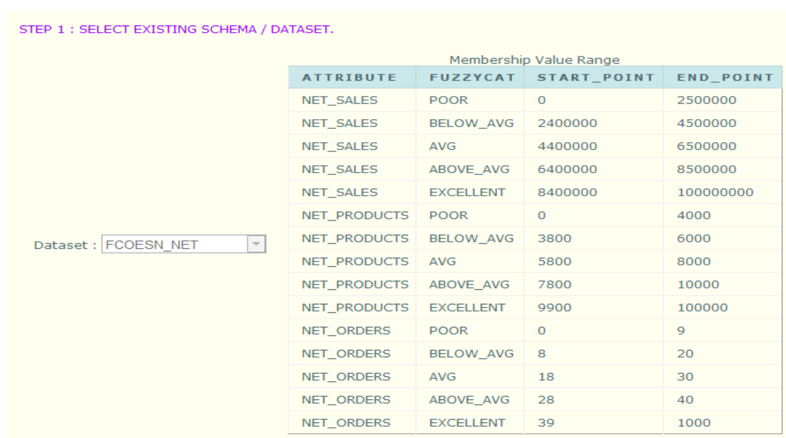


Figure 9: Query 1 Result in FDM/FAOES Using Excel Pivot Charts

The result shows that both FDM/FAOES and WebFDM are producing similar results. For additional queries, refer to appendix C10. Data mining begins with selecting the desired dataset that WebFDM has frozen as the most optimum learned dataset during the learning phase (Figure 10).



ATTRIBUTE	FUZZYCAT	START_POINT	END_POINT
NET_SALES	POOR	0	2500000
NET_SALES	BELOW_AVG	2400000	4500000
NET_SALES	AVG	4400000	6500000
NET_SALES	ABOVE_AVG	6400000	8500000
NET_SALES	EXCELLENT	8400000	100000000
NET_PRODUCTS	POOR	0	4000
NET_PRODUCTS	BELOW_AVG	3800	6000
NET_PRODUCTS	AVG	5800	8000
NET_PRODUCTS	ABOVE_AVG	7800	10000
NET_PRODUCTS	EXCELLENT	9900	100000
NET_ORDERS	POOR	0	9
NET_ORDERS	BELOW_AVG	8	20
NET_ORDERS	AVG	18	30
NET_ORDERS	ABOVE_AVG	28	40
NET_ORDERS	EXCELLENT	39	1000

Figure 10: Selecting the optimum dataset frozen by WebFDM during learning phase

In the next step, new data was added.

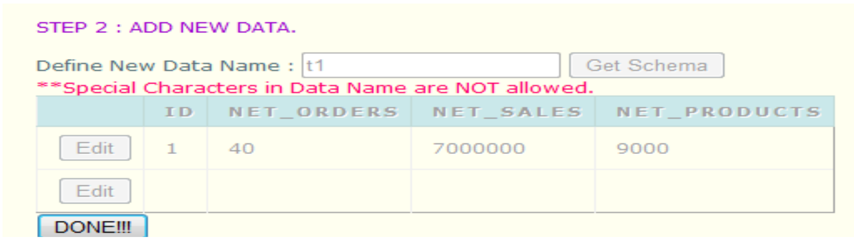


Figure 11: Add new data

In the last step, process the data and see the result. As figure 12 shows, both triangle and trapezoidal graphs recommend that this employee should receive a raise and a gift.

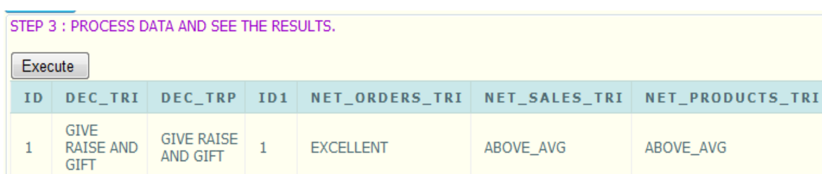


Figure 12: Process data and receive result

Limitations of WebFDM:

- Improve the input process to the system. Currently, the software assumes that users know PL/SQL and SQL languages to insert the data in a table of the Oracle database. The feature to import data from an Excel file to the database will help users to manage data in Excel files and to add data in the database easily without knowing the database language.

- Expand the software to reduce entering a number of possible rules that can be entered into the system by implementing the Combs method.
- WebFDM supports data mining based on Complete Dataset Maturity (CDM) but not on Partial Dataset Maturity (PDM).
- WebFDM, like its predecessors, treats all the fuzzy attributes with the same set of fuzzy categories however, it is very useful to have different fuzzy categories for different fuzzy attributes.

Chapter 3

KRT Database

3.1 Background for KRT (Knee Replacement Therapy Database)

The KRT database was created to simulate a series of patients that could act as a test case for the Clinical Decision Support System software WebFDM. The KRT database creation and implementation process had several processes contributing to its purpose. First, the KRT database acted as the expert for the future knowledge based system thus, representing important relationships between attributes that relate with a patient and his/her recovery. Finally, the patient has real time data that is simulated based on a set of scenarios that are used to test the system for accuracy.

According to past research and studies, the healing time of a patient can vary dramatically depending on a number of different factors. For example, one patient might work out more than another patient or one patient might have a greater tolerance of pain than another patient. All of these factors play a role in the recovery time of a patient [3]. For purposes of this study, several patient recovery factors were taken into account as shown in the data model on Figure 13. Each of these various factors are linked to the Patient entity thus, the Patient entity is acting as the central entity of the database. The branches that stem from the Patient entity are exercises/workout, weight, number of diseases, number of medications, number of days on medications, reason for injury/surgery, and nutrition/diet habits. The patient data was generated by having a variety of patients that heal in varying lengths of time and based on different scenarios.

The purpose of this was to create a variety of cases that new patients could be compared for testing.

Even though the factors affecting a patient can be broad, a patient's recovery can still be quantified. By quantifying a patient's recovery using qualitative results, an expert could determine when a patient will heal under various circumstances based on past experiences. Based on this setting, a machine learning technique could be used to simulate the expert.

3.2 Design and Implementation of the KRT Database

After performing an extensive search to make an accurate relationship for the KRT system [Mayo Clinic, 2010; Bonesmart, 2009; Huddleston, 2005; Haynes, 2009; NIH, 2004], the data model in Figure # was developed. This data model was designed to provide information about the patient's weight during recovery, different possible diseases they could have, nutrition intake, possible injuries, and patient workout/exercises. The appropriate attributes were selected for these entities. These attributes describe kind of information is needed to be maintained for the KRT database. KRT databases in a normalized database and support fifth normal forms. It also supports full referential integrity. The create tables script in Appendix D1 demonstrates the needed codes to support full referential integrity.

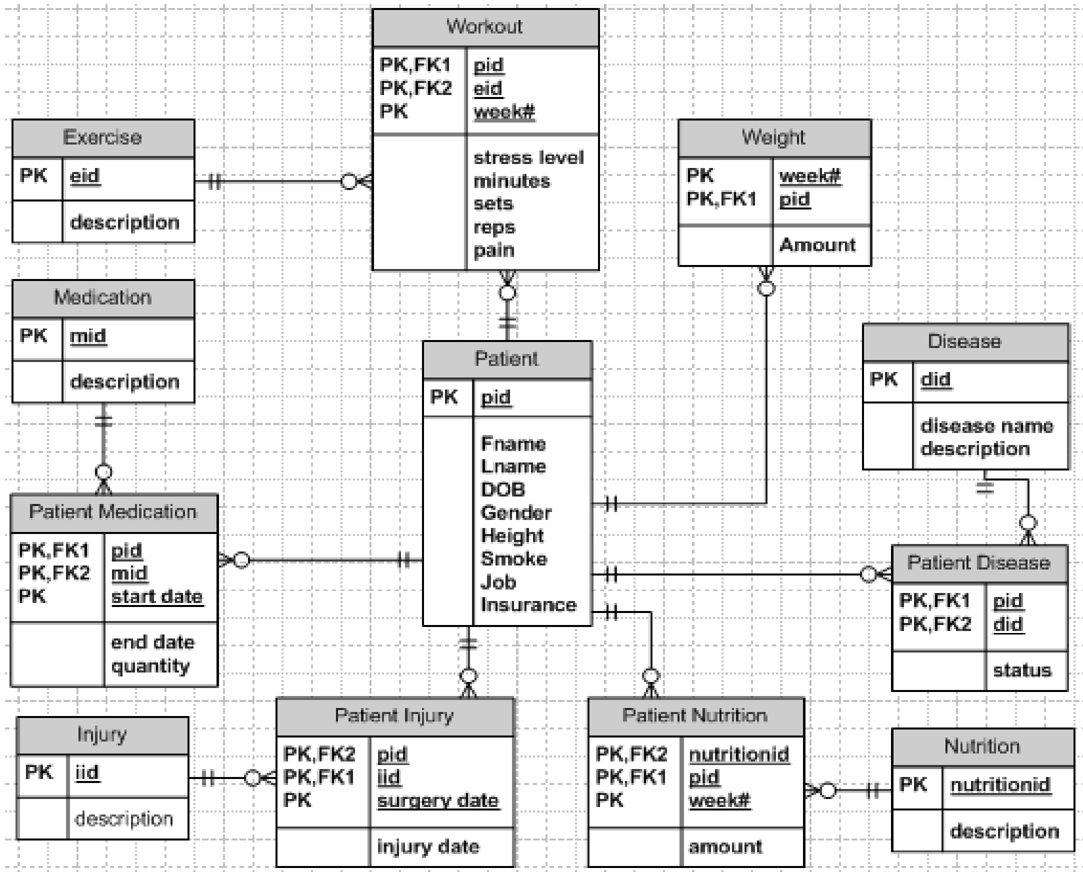


Figure 13: KRT Data Model

KRT database was implemented in Oracle 9i and the script to create and populate the database is in appendix A1. The list of tables in KRT is listed in the Table 9.

Table 9: List of Tables in the KRT Database Stored in the Oracle 9i Database

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

Chapter 4

Methodology, Results, and Analysis

4.1 KRT Revisited

Initially, a set of patient data was created as a "simulated" case study for the system to be implemented and tested. Using this simulated data, this allowed a set key where the results could be compared for accuracy. This key is shown in the table below and is based upon the reference below.

Table 10: Patient ID and Expected Decision

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

The next portion involved the identification of fuzzy components. The fuzzy components involved: identifying the fuzzy attributes, establishing the fuzzy categories, describing the fuzzy category memberships, and implementing the fuzzy rules.

4.2 Initial Phase

Initial phase would be involved to define data set, fuzzy attributes, fuzzy categories, membership values, and setting up initial rules.

4.2.1 Identification of Fuzzy Attributes

In order to identify the fuzzy attributes, a relationship between all factors that affect a human body must be developed. This led to the design and implementation of the KRT database. The KRT database relates criteria about physical therapy that can potentially affect someone as they are recovering. While recovering, the following data will be recorded by an expert (a physical therapist) each week and placed into this system to determine when they have healed from physical therapy. Based on this data model, the contributing fuzzy attributes that have an impact on the patient are shown in Table 11.

Table 11: Fuzzy Attributes Used in WebFDM

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

4.2.2 Identification of Fuzzy Categories

Once the fuzzy attributes were identified, the fuzzy categories were established as shown in the table below.

Table 12: Fuzzy Categories used in the Fuzzy System

FUZZYCATID	NAME
9	Low
10	Mid
11	High

4.2.3 Identification of Fuzzy Membership Values

In addition, the fuzzy category membership values had to be established. These membership values varied depending on the attribute being measured and how this value was calculated, fuzzified, and then interpreted.

Each membership value was specified for each attribute in the following two tables depending on the membership function being used. For all attributes, view Appendix A2

Table 13: Fuzzy Attributes and Membership Values for the Triangle 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

Table 14: Fuzzy Attributes and Membership Values for the Trapezoidal Function

MEMBID	WTID	FAID	ATTRIBUTE	FUZZYCAT	START_POINT	MID1	MID2	END_POINT	GRAPH
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

4.2.4 Identification of Fuzzy Rules

After the fuzzy attributes were identified, the fuzzy categories set, and the fuzzy membership values placed, the fuzzy rules were then identified. Depending on patient progresses, the fuzzy rules can determine when a patient has healed from physical therapy. Examples of fuzzy rules are show in Table 15.

Table 15: Fuzzy Rules Used in WebFDM

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

4.2.5 Decision Tables

Once the fuzzy rules were implemented, the decision values were checked on a specific patient depending on how they were progressing. As shown in Table 16, we can see the implementation of the initial phase.

Table 16: Decision Table of the First Implementation

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

4.3 Learning Phase

After the initial implementation was completed, the learning phase began. In this phase, the decision table was evaluated to ensure the accuracy of the system. In other words, the decisions made by the system should be correct based on the values placed for each patient (IDs 1-10 respectively).

As can be seen in Table 16, there were several no decision values that were expected to have an outcome based on the key described in Table 17. Based on these results, the fuzzy category memberships were re-established and implemented into a new worktable as can be seen in Table 17.

Table 17: The First System and the Second System Successfully Implemented

WTID	COMMENTS	DB_POPULATION_SCRIPT	SQL_CODE	DATABASEID	CID	WTNAME
22	KRT-Data1	-	-	18	2	WT22
23	KRT-Data2	-	-	18	2	WT23

The main change made between the first and second system implementation was the number of fuzzy rules that were implemented (View *Appendix H* for a list of the fuzzy rules in work table 23 - the new KRT implementation).

Based on the results, the DEC_TRI (decision for the triangle membership function) successfully made a decision for all the different patients (labeled by their ID number). The DEC_TRP (decision for the trapezoidal membership function) was less consistent than the DEC_TRI. The DEC_TRP agreed with the DEC_TRI function, however the DEC_TRP failed to correctly assess a patient as healed.

Table 18: Decision Table for the Final Fuzzy System

Detail Data#2

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

4.4 Mining Phase

Once the learning phase was completed, the mining phase began. The mining phase involved looking at the other attributes in the KRT data model and attempting to look for correlations between the healing time of a patient and that factor being measured. For this study, the analysis performed used the triangle function since this model most accurately defined the patient key established before the study began. In the following figure, an example is given to show how data mining was used to draw other possible conclusions from the study. For more examples, please view *Appendix J*.

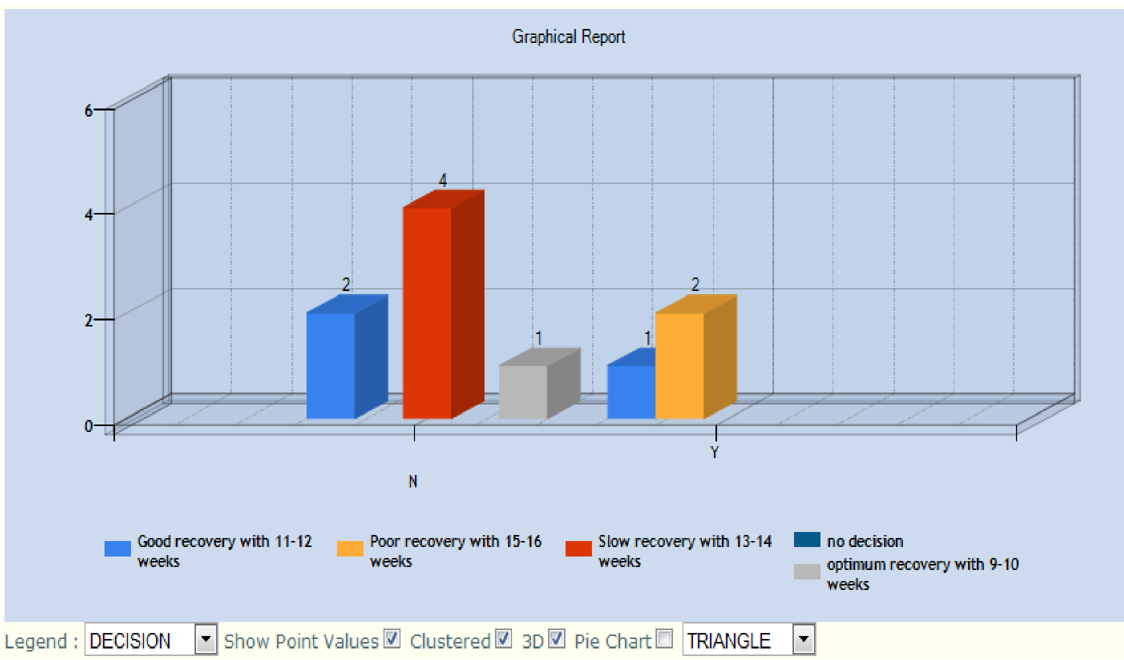


Figure 14: Time Required for a Patient to Heal Based on their Smoking Habits

4.5 Analysis and Discussion of Results

The system was able to take various fuzzy components and use those components to make decisions about the patients. Table 19 and Figure 15 represent the outcome of the second implementation. This matched with the original plan of where the patient's data should fall in the system.

Table 19: Final Decision Table for the Second Implementation

Detail Data#2

DEC_TRI	COUNT
Good recovery with 11-12 weeks	3
Poor recovery with 15-16 weeks	2
Slow recovery with 13-14 weeks	4
optimum recovery with 9-10 weeks	1

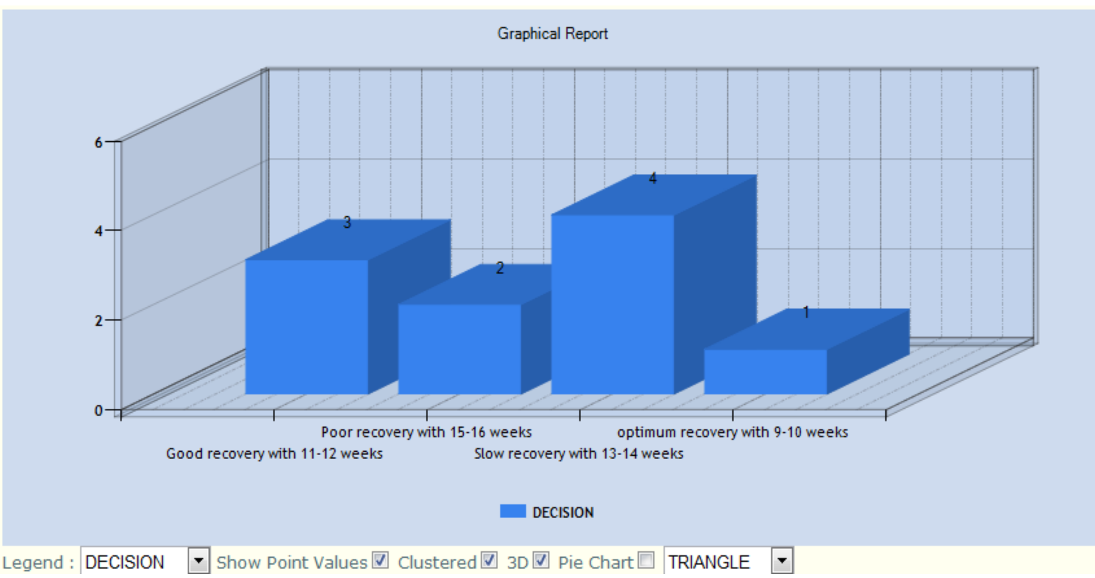


Figure 15: Final Decision Graph for Second Implementation in Work Table 23

The system acted as a fuzzy active database that could handle patient data that the user identified as important, and based on those identified rules, the system could make educated decisions about when a patient is healed. However, the system could not predict when a patient will be healed. A hypothesis was derived to test whether this could be a feasible study.

Chapter 5

Data Mining

5.1 Fuzzy Decision Maker Based on a Full Data Set

Through the learning phase, WebFDM was able to tune the membership values and modify the fuzzy rules to identify the optimum settings that generates the most accurate result. The optimum settings were saved as the learned database and all mining cases were processed against the learned database.

To test the fuzzy decision maker, a test subject named "Lorry Miller" was placed into the database with a full set of data. As can be seen in Table 20, the fuzzy decision maker determined that Lorry had a "good recovery within 11-12 weeks". It is important to note that this patient was not involved in the learning phase.

Table 20: Test Data for Patient Lorry Miller

New patient	Data
id	5
Name	Miller, Lorry
Gender	F
Height	69
Smoke	N
Job	Homemaker
Age	31
workout	14
Pain	901
Stress	7
Disease	0
number of medications	1
Days_MD	2601
Nutrition	13390

5.2 Fuzzy Predictor Based on a Partial Data Set

To test the fuzzy predictor based on only having partial data from the patient, a sample patient that has been through the process for 5 weeks has performed the following results as shown in Table 21. (More detailed information can be seen in Appendix K)

Table 21: Summary of Sample Patient Data Through Week 5

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

As can be seen in this brief table, the patient has been healing at a rate slightly faster than the average rate. At this point in time, if the patient were to heal at a maximum rate of healing, the fuzzy predictor would predict 11 weeks to be fully healed. Whereas, if the patient were to heal at the slowest possible rate after week 5, the patient would heal in 14 weeks. (Note that the patient's age membership category remained constant for this simulation). Refer to Figure 16 for a graphical illustration of how the two trends differ.

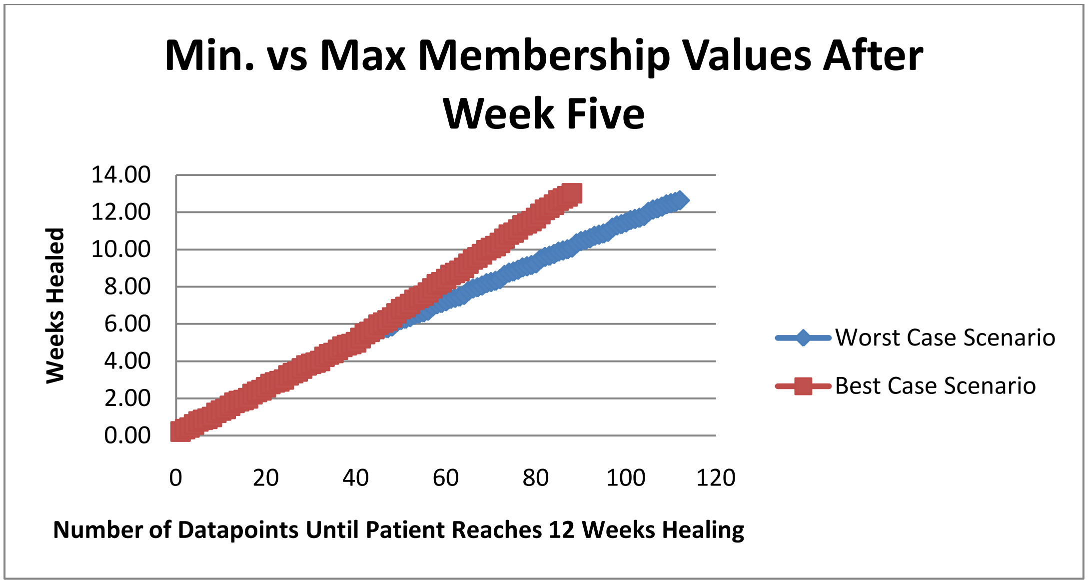


Figure 16: Minimum VS Maximum Healing Trails a Patient Can Follow Based on what the Patient has Done Through Week 5

5.3 Mathematical Model to Predict Results

To develop a clinical decision support system that can use machine learning to make decisions about partial data, it is important to quantify each value of partial data to result in a final decision or prediction representing each fuzzy rule.

Selecting a modeling technique to accurately describe the relationship between the fuzzy attributes, components, membership values and rules can vary depending on the settings involved in the physical system. For instance, the situation might be best suited for a numerical and piecewise summation methodology, or it might be best represented using a system of fuzzy differential equations.

A system that can use a numerical, piecewise summation methodology will assume that each parameter is independent of other factors. This means that if pain were to increase, the stress levels of a patient would remain unaffected. Furthermore, there must be a rate of healing that can be quantified from week to week to determine how far each patient has progressed. To measure this healing process, a generalized formula is shown below.

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

$$TotalHealing = \sum_{n=1} W_n Heal = W_1 Heal + W_2 Heal + \ldots + W_n Heal$$

Equation 2: Each Week's Healing Represented by the Summation of Each Attribute for Each Week

$$W_n Heal = W_n (age_n + workout_n + pain_n + stress_n + \#OfDiseases_n + \#OfMeds_n + DaysOnMeds_n + TotalNutrition_n)$$

This equation takes the summation of every week of healing until the patient has fully healed. Additionally, each week takes the various fuzzy attributes and sums the amount of healing each attribute contributes to the healing that week. Also, the fuzzy membership values are used to fluctuate the amount of healing done each week. For instance, if there is a positive correlation between healing and being high in a specific attribute, then more healing is received for that patient for that attribute in that week. An example is shown in the equation below:

Equation 3: Weighted Weekly Attribute Based on the Fuzzy Attribute and the Fuzzy Membership Value

$$W_{nweighted} age = W_n age * (fuzzymembe\ rshipvalue)$$

This equation adjusts the values for each attribute at each week and based on how the patient follows their recovery, that patient will heal more quickly or slowly based on how they do.

Since the original data for the patients had a spectrum of 9-16 weeks healing, the fuzzy membership categories for the summation function must equal 9 weeks in an optimum setting and 16 weeks for a poor situation. Therefore, in an ideal situation if all membership category values are set to $1.\overline{33}$, a patient would have an equivalent of 12 weeks healed in a 9 week period, whereas in a poor setting, if the membership values were set to 0.75, a patient would have an equivalent healing of 16 week. These values can be derived using the equations below:

Equation 4: Total Healing by a Patient Represented by Past Healing Plus New Healing

Healing = Old Healing + New Incremental Healing

Equation 5: New Healing Represented by the Week Number, Fuzzy Parameter, and the Fuzzy Membership Value Assigned

New Incremental Healing = (Week #)(Fuzzy Parameter)*(Fuzzy Membership)*

In equations 4 and 5, healing is calculated using a recursive formula and the incremental value changing the recursive iteration is based on the fuzzy values obtained in the learning phase. Refer to Appendix K2 for a detailed description of this output.

5.4 Fuzzy Predictor Implementation

In order for the system to act as a fuzzy predictor (and not just a fuzzy decision maker), mathematics are used to simulate the possibilities of outcomes based on the important fuzzy components identified by the user. Mathematical modeling can be used to predict when a patient is healed by modeling the initial starting point (the initial fuzzy attributes and fuzzy membership values) of each patient, the possible ending points (which represent all the fuzzy rules/decisions that the system could possibly make), and assuming a linear approximation between the two points. Refer to Figure 17 for a diagram of this idea.

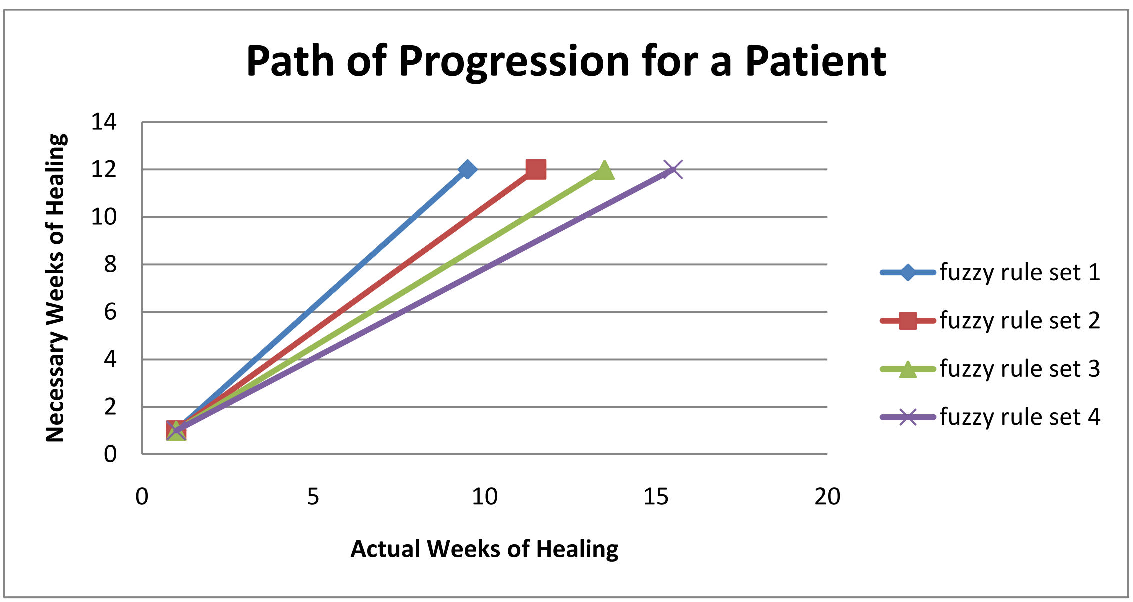


Figure 17: Progression Path for Patients Based on the Different Fuzzy Rules Assigned

As can be seen in Figure 17, the starting point occurs when a patient begins physical therapy. This point is held constant for this model, however this point can vary based on a patient's condition. The y-axis represents the total amount of recovery required for a patient to heal. In this case, it is assumed that there is a requirement of 12 weeks to be fully healed at an average rehabilitation rate. The x-axis represents the number of weeks the patient has been in physical therapy. In the case of fuzzy rule set 1, the patient was only in physical therapy for a 9-10 week duration and had the equivalent healing of an average patient at 12 weeks of healing.

This model uses the initial values established in the fuzzy components as boundaries of where a patient is starting, currently is at, or when a patient has completed physical therapy. Then, by converting this system back into a mathematical model, a linear approximation is made to estimate the healing path for a patient following a specific lifestyle and work-out routine.

5.5 Converting the Fuzzy System into a Mathematical Model

In order to make this fuzzy system a mathematical model that can translate back as possible fuzzy rule sets to make a predicted decision, the fuzzy components must be parameterized, simulated based on predicted patient habits from previous weeks, and then translated back to a specified fuzzy rule to allow the program to make a viable decision about whether the patient has healed. Thus, forecasting when a patient will be healed at a specific time in the future.

To parameterize the fuzzy model to allow a mathematical model to work, the fuzzy components must first become a series of parameters or bounds. The fuzzy attributes are now assumed as parameters that will consist of the same name as the fuzzy attributes previously listed. The parameters are similar to before with the fuzzy attributes and using all the same assumptions as before:

1. Age
2. Workout
3. Pain
4. Stress
5. Number of diseases
6. Number of medications
7. Days on medications
8. Total nutrition for that week

The actual number of weeks a patient spends in physical therapy is dependent on the understanding and assumption that a patient is required a total healing period of 12 weeks at the average patient recovery rate. Also, healing has been categorized as a quantity that a patient acquires each week for each attribute. Using a piecewise summation of these parameters for each week, a user can view the healing of a patient as a quantity of amount healed per week per attribute. Refer to the example in Table 22.

Table 22: Fuzzy Attribute Weight Towards Total Healing

Fuzzy Attribute	Fuzzy Parameter	12 Week Program Weight
Age	a	2
Workout	w	2
Pain	p	1
Stress	s	1.5
# of Diseases	d	2
Number of Medications	m	1
Days on Medications	z	1
Total Nutrition	n	1.5
	Base Number of Weeks	12

After the fuzzy parameters have been assigned a program weight, the fuzzy categories were converted to a numerical constant (see Table 23). These values were assigned in an optimal solution (all parameter categories are 1.33) would consist of healing in 9 weeks, and the worst solution (all parameter categories are 0.75) would consist of healing in 16 weeks.

Table 23: Fuzzy Category Values Based on a Positive Correlation to Healing Quickly

Low is a positive correlation to healing quickly	
LOW	1.33
MID	1
HIGH	0.75

High is a positive correlation to healing quickly	
HIGH	1.33
MID	1
LOW	0.75

Additionally, each parameter is assigned a value as dynamic or static. The parameter category will change on a week-by-week basis or remain constant and a relative weight relating to that week's total amount healed. (See the column weight in Table 24 and Figure 18.) Each parameter is also matched with the positive correlation category membership value based on a more efficient healing contribution.

Table 24: Parameter Identification Summary

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%	

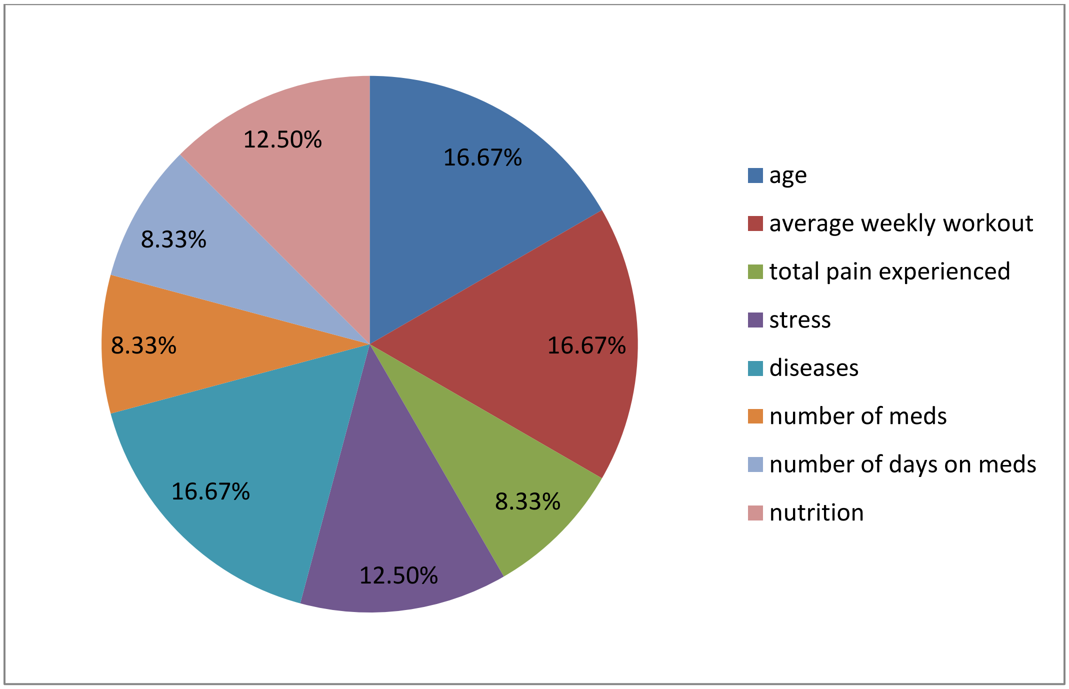


Figure 18: Weight of Each Attribute and how it Contributes to the Overall Healing Process

Additionally, each parameter is assigned a value as dynamic or static (the parameter category will change on a week by week basis or remain constant) and a relative weight relating to that week's total amount healed (the column weight in the table below). Each parameter is also matched with the positive correlation category membership value based on what would assist in a more efficient healing contribution.

First, each parameter has a 12 week program weight that must be divided by 12 to make this an average weekly value that a patient can progress through each week.

Second, for each week, each parameter must be multiplied to the fuzzy category constant based on if the patient was categorized as high, mid or low for that week (as deemed by the expert). These values can be seen in the table below. Some important things to note is that the healing progress will take the sum of all preceding healing before it, thus showing a week by week healing log of the patient based on the fuzzy rules previously established and the fuzzy concepts used in the decision making tool. Also, it can be seen that the healing progress is 1.02 at the end of the first week, meaning this patient is 0.02 weeks ahead of schedule.

Table 25: Week One Example of Healing for a Patient

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

This week-by-week calculation will continue until the patient reaches a healing progress of 12 weeks or higher.

This design allows a user to input fuzzy category constants into the system and it will generate an output that can assess how the patient is progressing. Additionally, the user can determine at a specific rate, how long it will take for the patient to heal, or what a patient might have to do in order to change or keep up their current progress. An example of this is in the figure below. In this case, the patient is healing at a random rate. However, if the patient were to maximize his possible time, the patient could heal in 11 weeks.

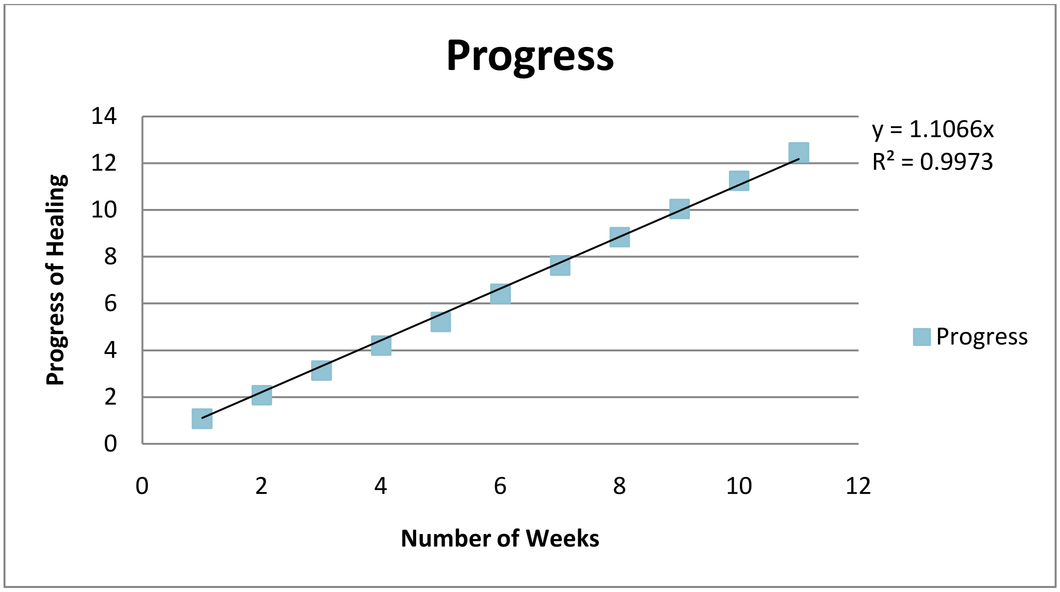


Figure 19: Healing Progress of a Patient

In Figure 19, the slope of the best fit linear curve was a slope value greater than one, which means this patient has healed quicker than the normal patient. In a normal setting, the patient would have a slope value of one. If the patient healed slower than the normal person, the slope value would be less than one.

Types of applications using WebFDM is divided into groups: CDM (Complete Data Maturity) and PDM (Partial Data Maturity).

CDM Applications: In order to use the learning engine, the mining data should have the same maturity of the experiments participated in the learning phase. This means to evaluate a patient's performance, a patient's recovery period should fall in the range of recovery period of those patients that were involved in the initial phase and the learning phase. Another example, in case of CDM for FAOES applications, one year of sales data is needed to evaluate sales performance since the data used to train the engines were all full year sales from different employees. This is no prediction since we only provide fuzzy decisions.

PDM (Partial Data Maturity) Applications: For FAOES applications, partial yearly sales data for a given employee(s) is available. Our fuzzy engine based on the learned component can predict how these employees could finish. So fuzzy predictors can provide several key strategies and later on can revisit the modify prediction if the actual data is different from planned data.

Immediate Data Maturity (IDM): Immediate and past data, if available, will be used to provide fuzzy decisions. In case of FAOES of CRM, the customer has been called several times with some waiting time and a source of inquiry to be our fault then specify decision reached using the learned engine.

Chapter 6

Conclusion

6.1 Concluding Remarks

In this research, we were able to create a database called KRT that held information regarding different patient's recovery from a total knee replacement. We were successful in using the WebFDM methodology and implementation using an active database to incorporate fuzzy logic analysis. The newly extended fuzzy database (FKRT) was tuned to reach its optimal stage and then was used for data mining. Since WebFDM only supported full data set to train the fuzzy database in the learning stage, the concept of fuzzy database predictor was hypothesized. We have demonstrated through extensive examples and mathematical modeling a methodology that uses the learned fuzzy database (FKRT) to predict the new patient recovery based on a partial data set, meaning the ability to predict the patient recovery when the patient just started the recovery or in the early stage of recovery.

6.2 Future Research

WebFDM should be expanded to support a fuzzy predictor based on a partial data set. Also, WebFDM should support additional limitations that were identified in the background review:

- Improve the input process to the system. Currently, the software assumes that users know PL/SQL and SQL languages to insert the data in a table of the Oracle database. The feature to import data from an Excel file to the database will help users to manage data in Excel files and add them to the database easily without knowing a database language.
- Expand the software to reduce the extreme redundancy of identifying all possible rules that may exist in the system by implementing the Combs method.
- WebFDM supports data mining based on Complete Dataset Maturity (CDM) but not on Partial Dataset Maturity (PDM).
- WebFDM treats all fuzzy attributes with the same set of fuzzy categories, however, it would be useful to have different fuzzy category for each different fuzzy attribute.

Bibliography

[Andriacchi, 1997] Andriacchi, Thomas P. "Gait biomechanics and the evolution of total joint replacement". 1997. Elsevier Science B.V. www.sciencedirect.com

[Azarbod, 2005] Cyrus Azarbod, “Fuzzy Active Database Research”, Original Tutorial Paper, Minnesota State University, Mankato, December 2005.

[Azarbod, 2006] Cyrus Azarbod, Hamed Sallam, Jafar Ali “An Automated Fuzzy Active Database for Employee Performance Evaluation Using Oracle”, 4th ACS/IEEE International
Conference on Computer System Applications, Dubai/Sharjah, UAE, March 8-11, 2006.

[Azarbod, 2007] Cyrus Azarbod, Cindy Thompson “A Fuzzy Active Relational Database for Employee Performance Evaluation ”, SEDE - 2007 International Conference on Software Engineering and Data Engineering, Las Vegas, Nevada, July 9-11, 2007.

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

[Bankar, 2010] Bankar, Anagha: Web Based Fuzzy Data Mining and Visualization. Thesis. MNSU – Mankato. 2010

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

[Giori, 2001] Giori, Nicholas J. "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, 2009] Haynes, Richard A. "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.

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

[Howell, 2009] Stephen M. Howell, MD; 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.

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

[Jain, 2010] A.K. JAIN, M.N. MURTY, AND P.J. FLYNN “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.

[Kaehler, 2010] Steven D. Kaehler, “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

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

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

[Mommersteeg, 1995] Mommersteeg, T.J.A. "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.

[NIH, 2004] "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

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

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

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

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

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

[Stein, 2009] Stein, Robyn; Hool, Caitlin. "Duke Study Finds Total Knee Replacements Increase Mobility and Motor Skills in Older Patients". 25 Jun 2009. The Institute for Health Technology Studies.
http://www.inhealth.org/wtn/Page.asp?PageID=WTN000099.

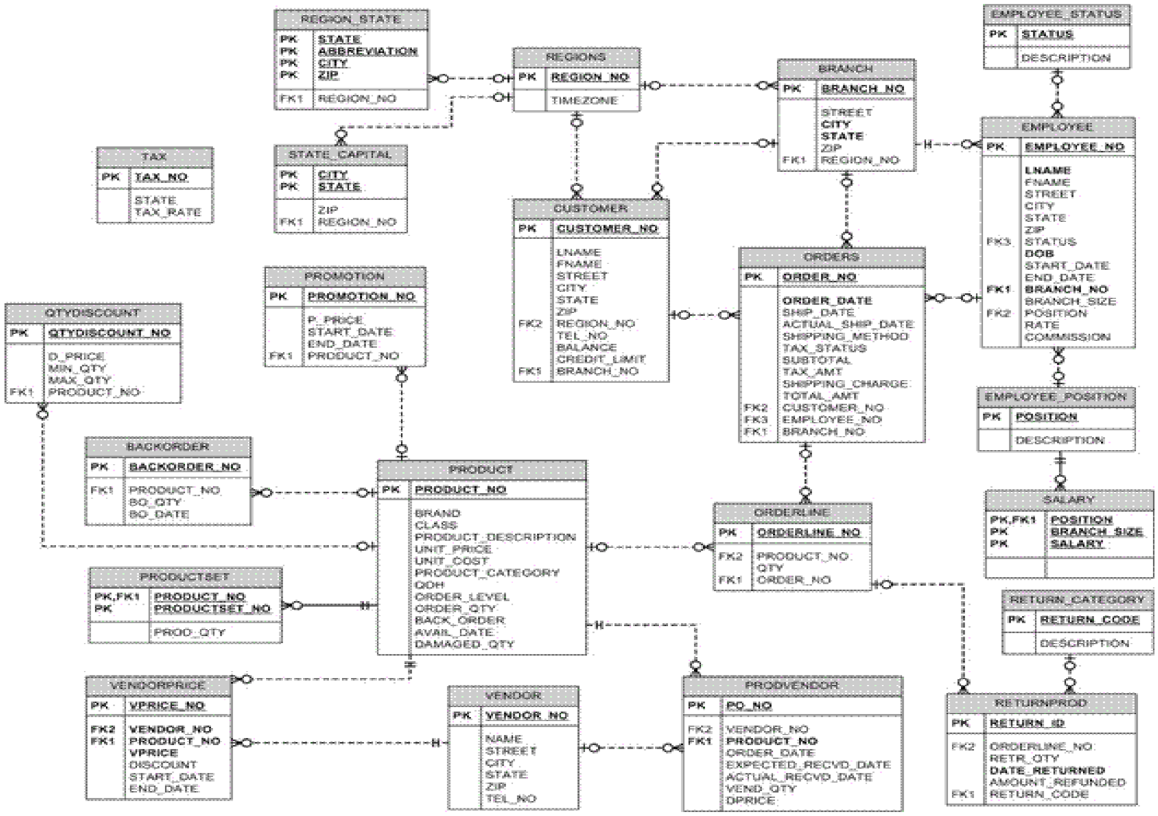
[Varadarajan, 2009] Varadarajan, Kartik M. "Can in vitro systems capture the characteristic difference between the flexion-extension kinematics of the healthy and TKA knee?". 17 June 2009. www.elsevier.com/locate/medengphy.

[Weber, 2009] Weber, Scott. Clinical Decision Support Systems and How Critical Care Clinicians Use Them. Journal of Healthcare Information Management. Vol. 21, No 2. http://www.himss.org/content/files/jhim/21_2/09_focus_clinical.pdf

[Wilson, 1996] Wilson, D.R. "A three-dimensional geometric model of the knee for the study of joint forces in gait". Accepted on 10 Jan. 1996. Elsevier Science B.V. Gait and Posture 5 (1997) 108-115. www.sciencedirect.com.

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

Appendix A1: OES Data Model

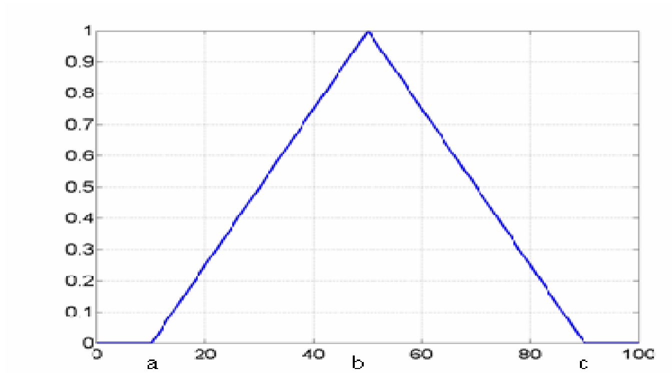


Appendix B2: FAOES Triangle and Trapezoid Function

Triangle Function

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

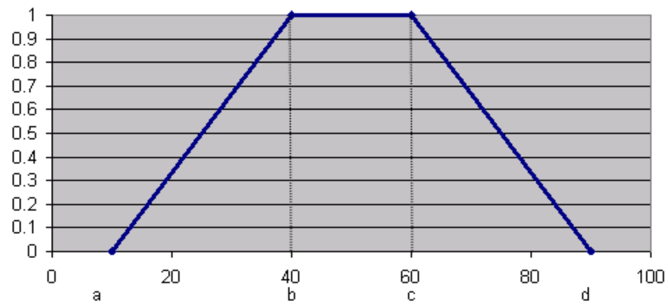
Attribute Name	Fuzzy Categories	Fuzzy Range	Membership Value (a,b,c)
Number Orders	Poor	[0,9]	(0,0,9)
	Below Average	[8,20]	(8,15,20)
	Average	[18,30]	(18,25,30)
	Above Average	[28,39]	(28,35,40)
	Excellent	[39,200]	(39,45,1000000)
Number Products	Poor	[0,40]	(0,0,40)
	Below Average	[38,60]	(38,50,60)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

Trapezoid Function

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 Trapezoid)

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 B3: 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

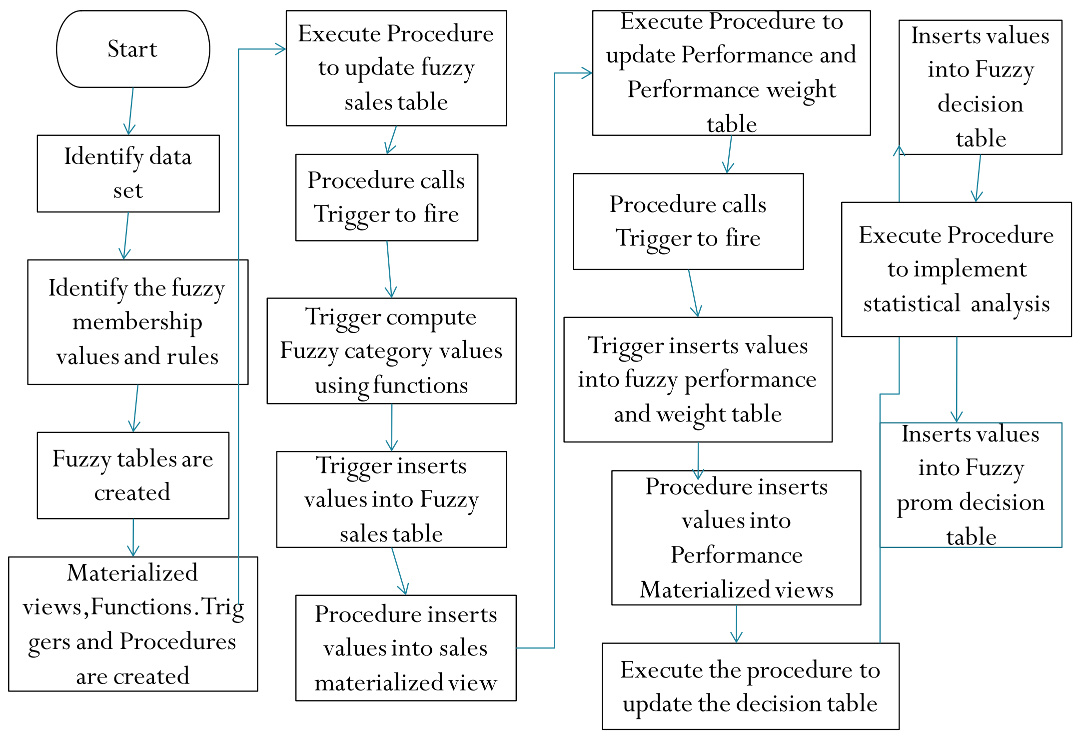
Appendix B1: ARDIF-FAOES query comparing both their results

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
1054	1054	Martin	TRI	GIVE RAISE AND GIFT
1060	1060	Bixler	TRI	GIVE 2% RAISE
1061	1061	Harris	TRI	GIVE RAISE AND GIFT
1062	1062	Blazek-White	TRI	GIVE WARNING
1065	1065	Parker	TRI	GIVE 15% RAISE
1066	1066	Bond	TRI	GIVE RAISE AND GIFT
1067	1067	Adams	TRI	GIVE 2% RAISE

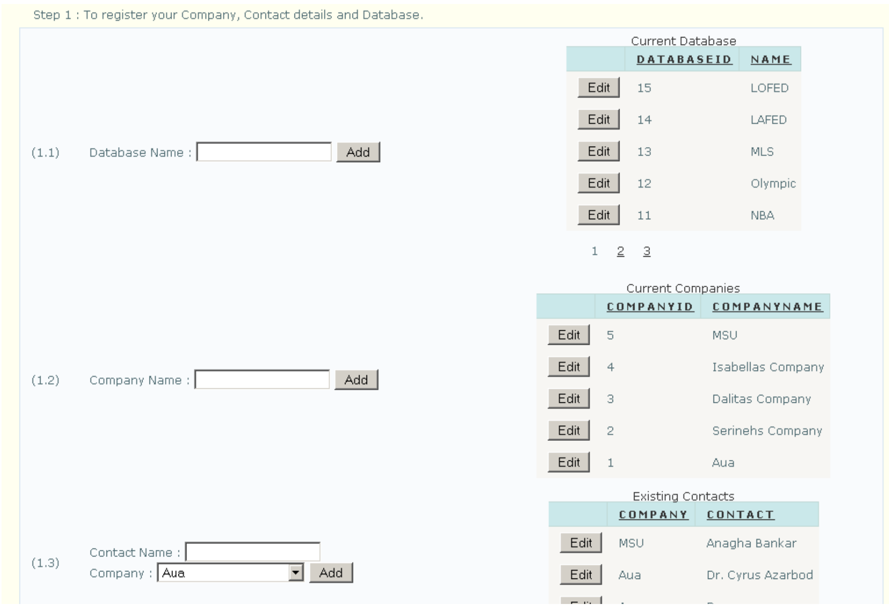
1069	1069	Alan	TRI	GIVE RAISE AND GIFT
1070	1070	Nabb	TRI	GIVE RAISE AND GIFT
1071	1071	Claggett	TRI	GIVE WARNING
1076	1076	Bullit	TRI	GIVE WARNING
1077	1077	McMillan	TRI	GIVE WARNING
1078	1078	Wright	TRI	no decision
1080	1080	Stone	TRI	FIRE EMPLOYEE
1084	1084	Johnston	TRI	GIVE 10% RAISE
1086	1086	Holman	TRI	GIVE 15% RAISE
1088	1088	Hanzdo	TRI	GIVE WARNING

Appendix B2: FAOES detailed flowchart to implement FAOES

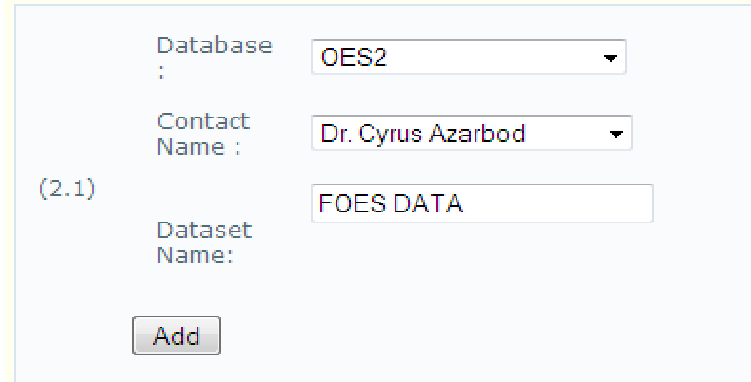
Simple Flowchart for Implementing Methodology



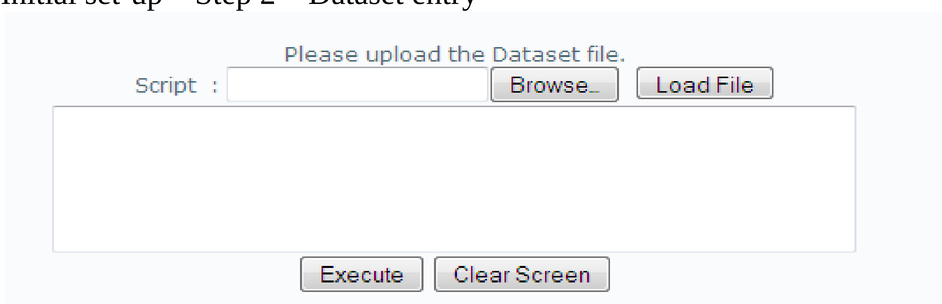
Appendix C1: Initial set-up – Step1 Add Contact name



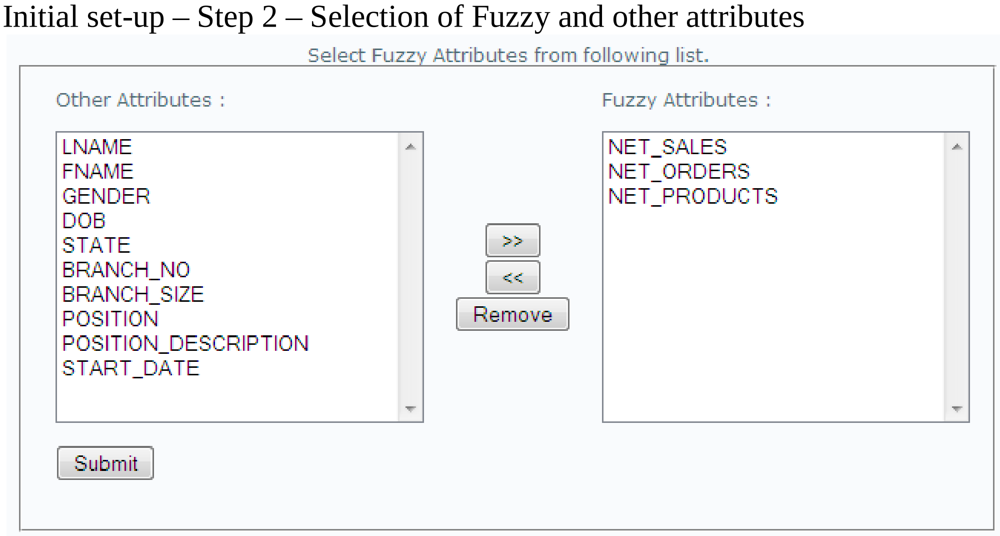
Appendix C2: Initial set-up – Step 2 Add Dataset Name



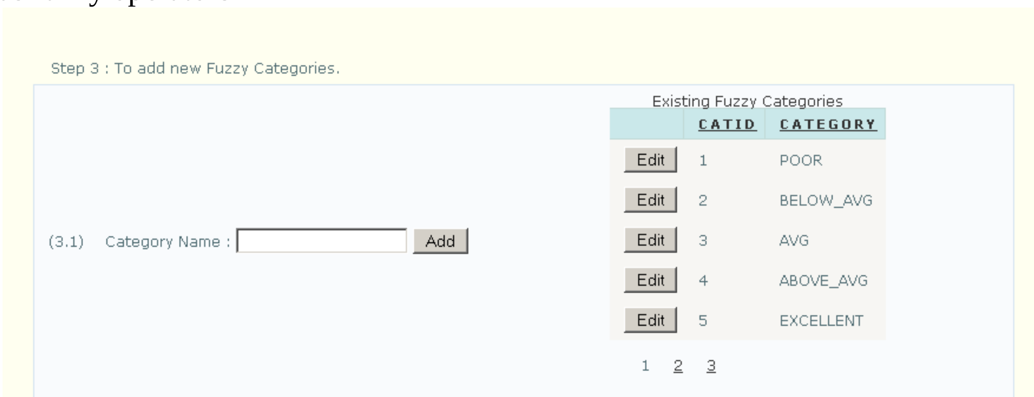
Initial set-up – Step 2 – Dataset entry



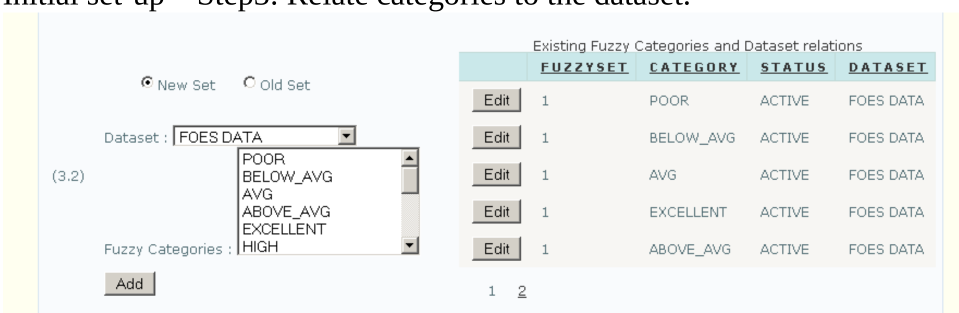
Initial set-up – Step 2 – Selection of Fuzzy and other attributes



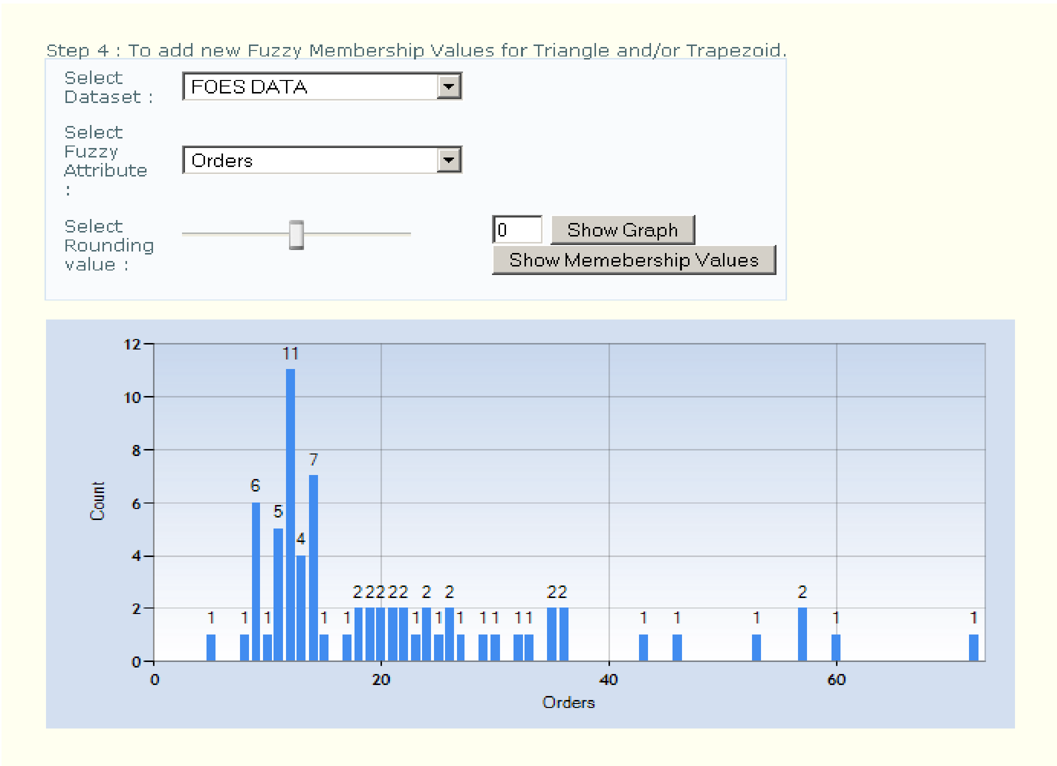
Appendix C3: Initial set-up – Step 3 Add Categories: Fuzzy categories are also called as fuzzy operators



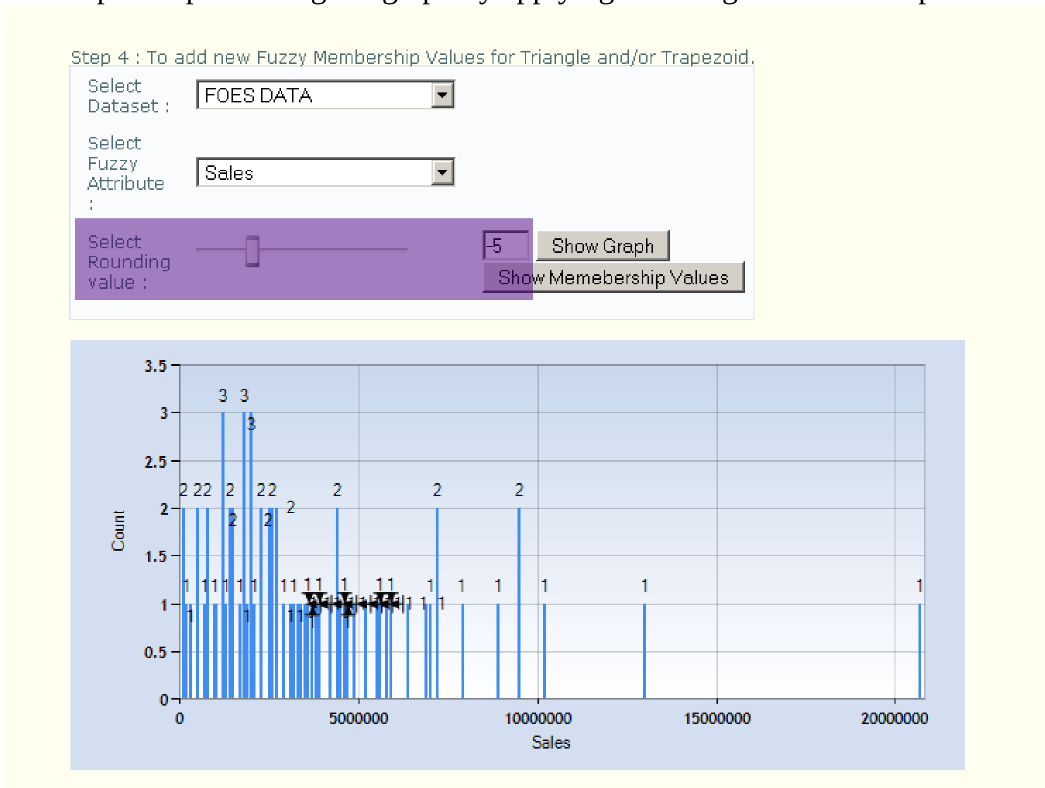
Initial set-up – Step3: Relate categories to the dataset.



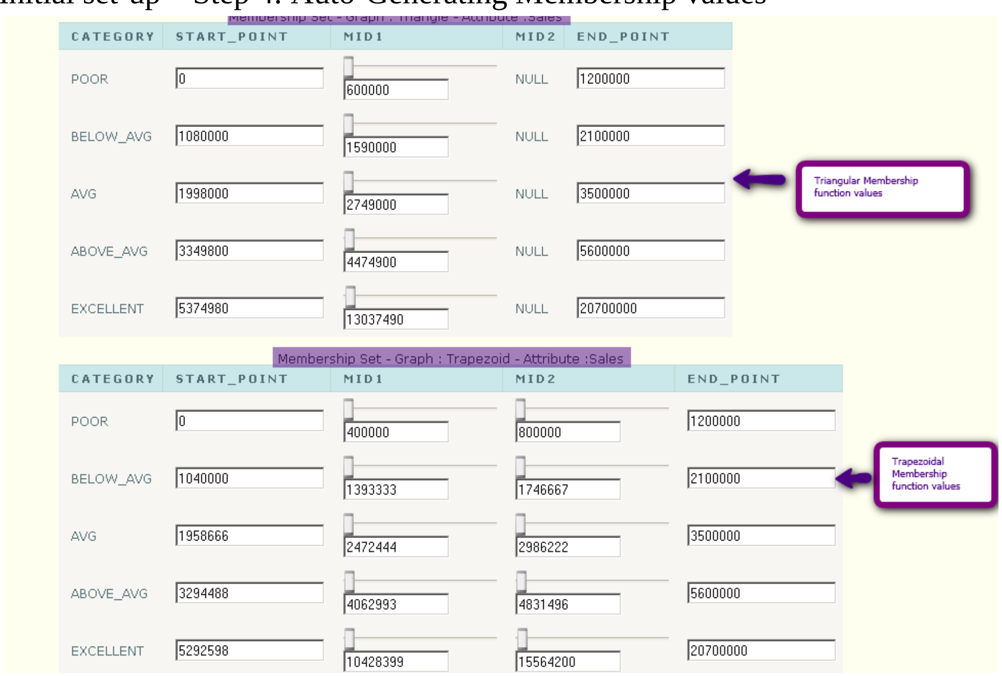
Appendix C4: Initial set-up – Step 4 Selection of Dataset, Fuzzy Attribute and Rounding value



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



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

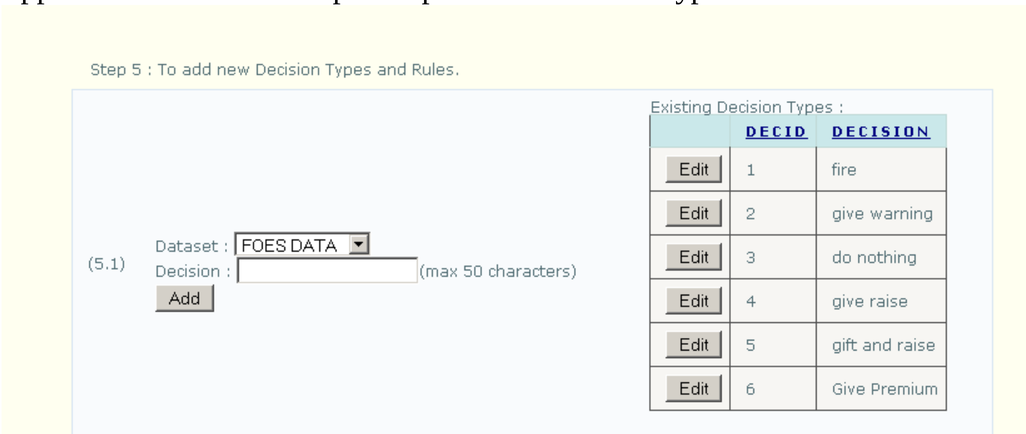


CATEGORY	START_POINT	MID1	MID2	END_POINT
POOR	0	600000	NULL	1200000
BELOW_AVG	1080000	1590000	NULL	2100000
AVG	1998000	2749000	NULL	3500000
ABOVE_AVG	3349800	4474900	NULL	5600000
EXCELLENT	5374980	13037490	NULL	20700000

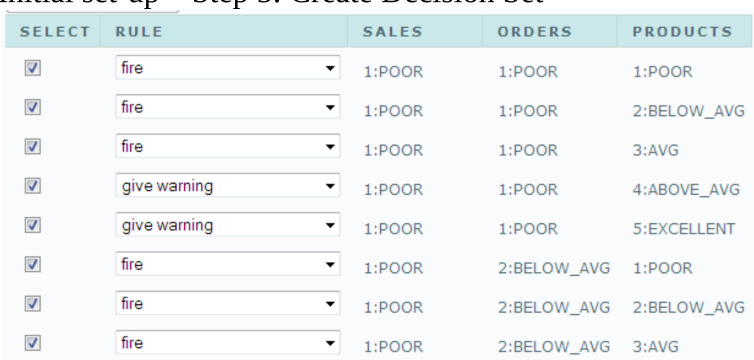
Membership Set - Graph : Trapezoid - Attribute :Sales

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

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

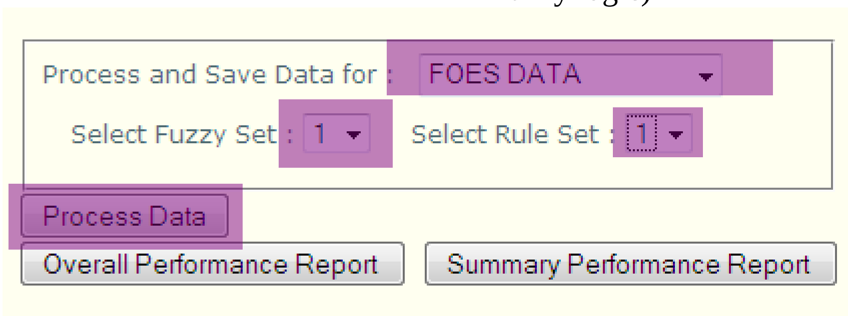


Initial set-up – Step 5: Create Decision Set

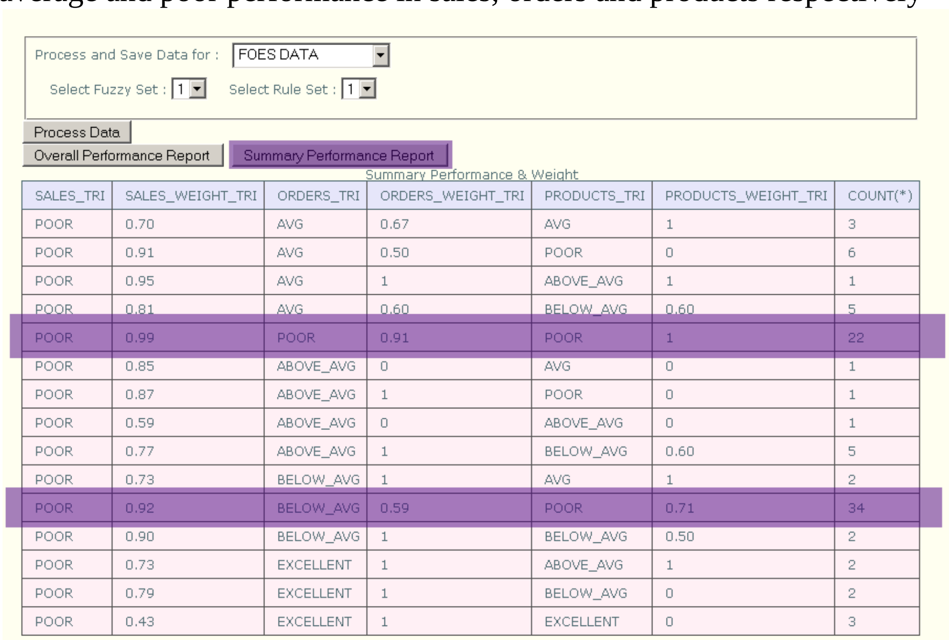


SELECT	RULE	SALES	ORDERS	PRODUCTS
☑	fire	1:POOR	1:POOR	1:POOR
☑	fire	1:POOR	1:POOR	2:BELOW_AVG
☑	fire	1:POOR	1:POOR	3:AVG
☑	give warning	1:POOR	1:POOR	4:ABOVE_AVG
☑	give warning	1:POOR	1:POOR	5:EXCELLENT
☑	fire	1:POOR	2:BELOW_AVG	1:POOR
☑	fire	1:POOR	2:BELOW_AVG	2:BELOW_AVG
☑	fire	1:POOR	2:BELOW_AVG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

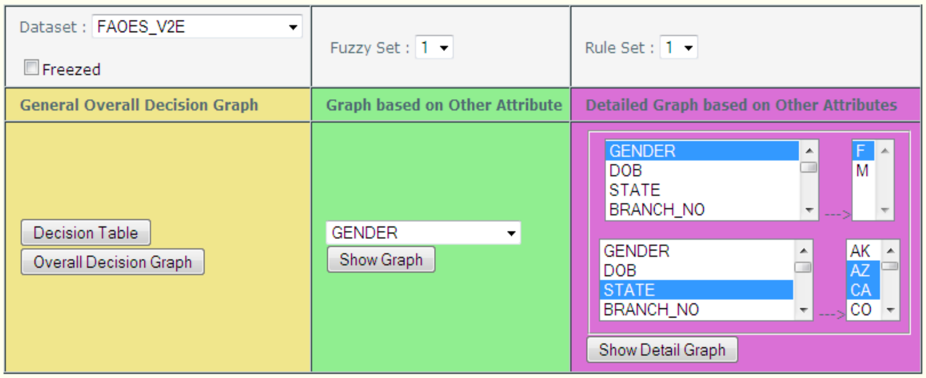
Initial set-up – Step7: Overall Performance report

Overall Performance Report | Summary Performance Report

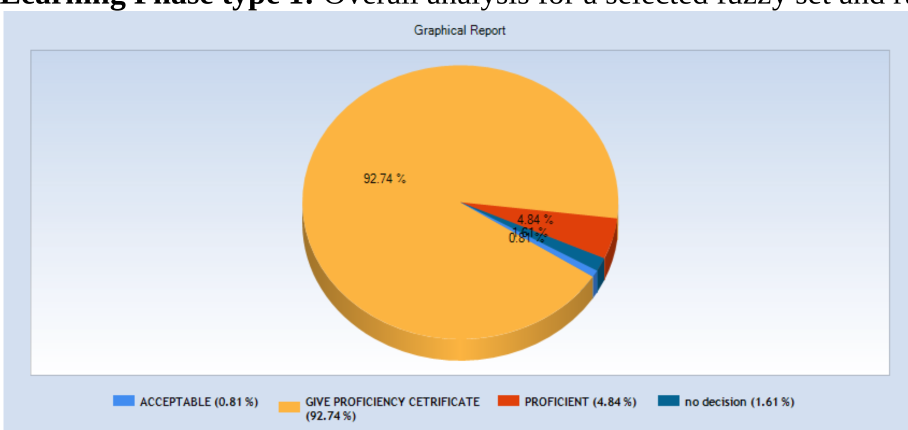
Overall Performance & Weight

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

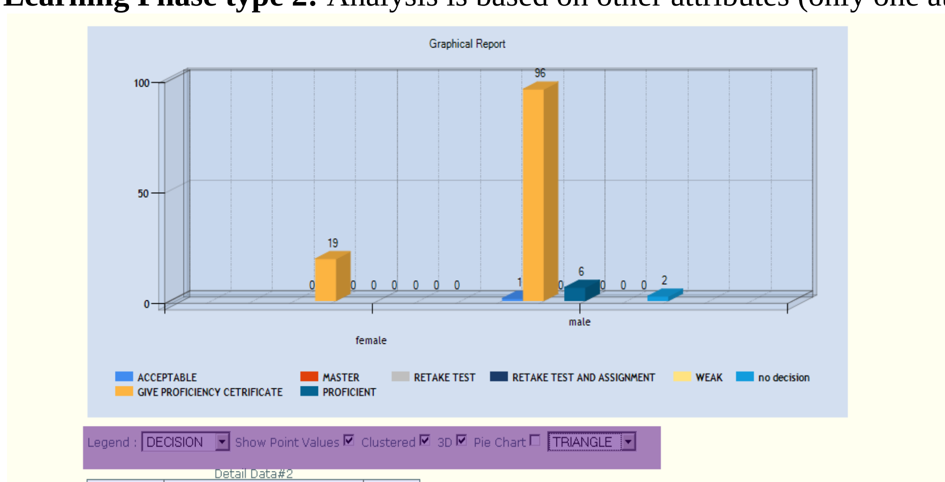
Appendix C7: Learning Phase: Analysis toolbox **(**the customized analysis toolbox helps to identify the optimal results (mark the freeze box to be used in mining)



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



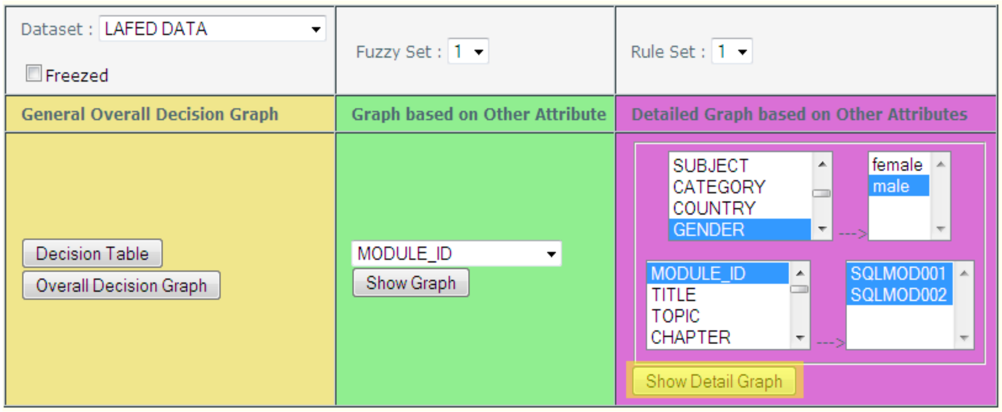
Learning Phase type 2: Analysis is based on other attributes (only one attribute)

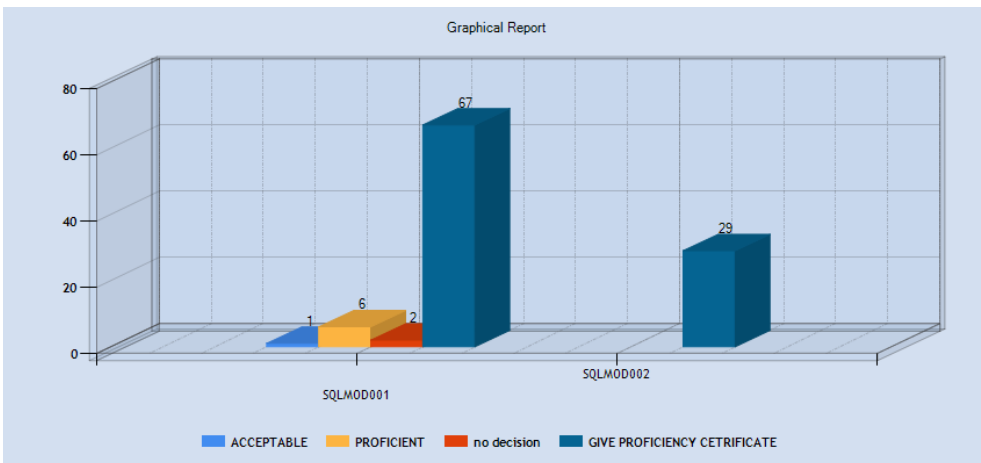


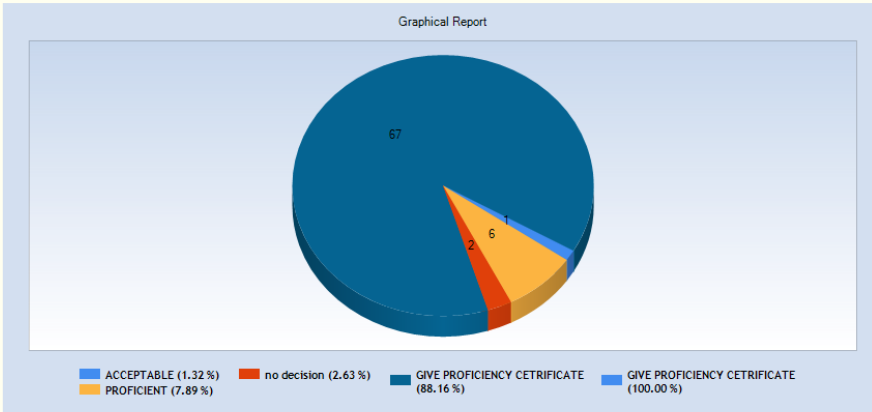
Detail Data#2

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) );
```

```
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,
```

```
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
1384	23	385	NO_MED	Low	-10	-7	-4	1	TRP
1386	23	385	NO_MED	Mid	-2	-1	0	2	TRP
1388	23	385	NO_MED	High	1	5	9	14	TRP
1390	23	387	NUTRIT	Low	7990	9495	-999	11000	TRI
1392	23	387	NUTRIT	Mid	9947	11473	-999	13000	TRI
1394	23	387	NUTRIT	High	11932	14971	-999	18010	TRI
1396	23	387	NUTRIT	Low	7990	8993	9996	11000	TRP
1398	23	387	NUTRIT	Mid	10298	11198	12098	13000	TRP
1400	23	387	NUTRIT	High	12369	14249	16129	18010	TRP
1402	23	382	PAIN	Low	690	745	-999	800	TRI
1404	23	382	PAIN	Mid	762	831	-999	900	TRI
1406	23	382	PAIN	High	852	981	-999	1110	TRI

1408	23	382	PAIN	Low	690	726	762	800	TRP
1410	23	382	PAIN	Mid	774	816	858	900	TRP
1412	23	382	PAIN	High	871	950	1029	1110	TRP
1414	23	383	STRESS	Low	-8	-2	-999	4	TRI
1416	23	383	STRESS	Mid	0	3	-999	6	TRI
1418	23	383	STRESS	High	4	11	-999	18	TRI
1420	23	383	STRESS	Low	-8	-4	0	4	TRP
1422	23	383	STRESS	Mid	2	3	4	6	TRP
1424	23	383	STRESS	High	5	9	13	18	TRP
1426	23	381	WRKOUT	Low	2	8	-999	14	TRI
1428	23	381	WRKOUT	Mid	10	13	-999	17	TRI
1430	23	381	WRKOUT	High	15	21	-999	28	TRI
1432	23	381	WRKOUT	Low	2	6	10	14	TRP
1434	23	381	WRKOUT	Mid	12	13	14	17	TRP
1436	23	381	WRKOUT	High	15	19	23	28	TRP
1542	23	387	NUTRIT	Low	7000	9000	-999	10000	TRI
1544	23	387	NUTRIT	Mid	9000	11430	-999	14000	TRI
1546	23	387	NUTRIT	High	13000	14774	-999	20000	TRI
1548	23	387	NUTRIT	Low	7000	8000	9000	10000	TRP
1550	23	387	NUTRIT	Mid	9000	10000	13000	14000	TRP
1552	23	387	NUTRIT	High	13000	14000	19000	20000	TRP
1554	23	382	PAIN	Low	600	750	-999	900	TRI
1556	23	382	PAIN	Mid	800	900	-999	1000	TRI
1558	23	382	PAIN	High	950	1050	-999	1200	TRI
1560	23	382	PAIN	Low	600	700	800	900	TRP
1562	23	382	PAIN	Mid	800	900	950	1000	TRP
1564	23	382	PAIN	High	950	1000	1100	1200	TRP
1482	23	386	DAYS_MD	High	10000	15000	-999	25000	TRI
1484	23	386	DAYS_MD	Low	-10	-10	3000	4000	TRP
1486	23	386	DAYS_MD	Mid	3000	4000	10000	12000	TRP
1488	23	386	DAYS_MD	High	10000	12000	25000	25000	TRP
1478	23	386	DAYS_MD	Low	-10	3000	-999	4000	TRI
1480	23	386	DAYS_MD	Mid	3000	5000	-999	12000	TRI
1500	23	384	DISEAS	Low	-10	-4	-999	2	TRI
1502	23	384	DISEAS	Mid	1	2	-999	3	TRI
1504	23	384	DISEAS	High	2	8	-999	16	TRI
1506	23	384	DISEAS	Low	-10	-7	-4	2	TRP

1508	23	384	DISEAS	Mid	1	1	2	3	TRP
1510	23	384	DISEAS	High	2	6	11	16	TRP
1512	23	385	NO_MED	Low	-10	-4	-999	2	TRI
1514	23	385	NO_MED	Mid	1	3	-999	4	TRI
1516	23	385	NO_MED	High	3	7	-999	14	TRI
1518	23	385	NO_MED	Low	-10	-7	-4	2	TRP
1520	23	385	NO_MED	Mid	1	2	3	4	TRP
1522	23	385	NO_MED	High	3	5	9	14	TRP
1442	23	380	AGE	Low	0	39	-999	50	TRI
1444	23	380	AGE	Mid	45	55	-999	70	TRI
1446	23	380	AGE	High	65	72	-999	100	TRI
1448	23	380	AGE	Low	0	34	43	50	TRP
1450	23	380	AGE	Mid	45	53	59	70	TRP
1452	23	380	AGE	High	65	69	76	100	TRP
1566	23	383	STRESS	Low	0	2	-999	4	TRI
1568	23	383	STRESS	Mid	3	5	-999	7	TRI
1570	23	383	STRESS	High	6	8	-999	10	TRI
1572	23	383	STRESS	Low	0	1	3	4	TRP
1574	23	383	STRESS	Mid	3	4	6	7	TRP
1576	23	383	STRESS	High	6	7	9	10	TRP
1578	23	381	WRKOUT	Low	0	5	-999	10	TRI
1580	23	381	WRKOUT	Mid	7	12	-999	17	TRI
1582	23	381	WRKOUT	High	14	18	-999	21	TRI
1584	23	381	WRKOUT	Low	0	1	7	10	TRP
1586	23	381	WRKOUT	Mid	7	10	14	17	TRP
1588	23	381	WRKOUT	High	14	17	20	21	TRP

Appendix H – KRT Partial Fuzzy Rule Set

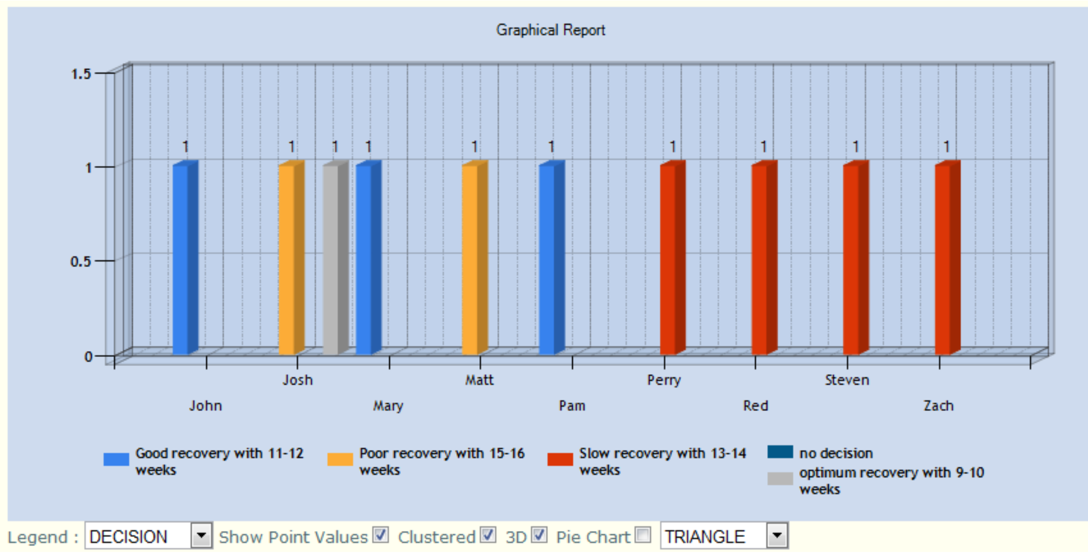
ID	AGE_TRI	WRKOUT_TRI	PAIN_TRI	STRESS_TRI	DISEAS_TRI	NO_MED_TRI	DAYS_MD_TRI	NUTRIT_TRI	AGE_TRP	WRKOUT_TRP	PAIN_TRP	STRESS_TRP	DISEAS_TRP	NO_MED_TRP	DAYS_MD_TRP	NUTRIT_TRP
1	Mid	High	Mid	Mid	Mid	Mid	Low	Mid	Mid	High	Low	Mid	Mid	Mid	Low	Mid
2	High	High	Mid	High	Mid	Mid	Low	Low	High	High	Mid	High	Mid	Mid	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

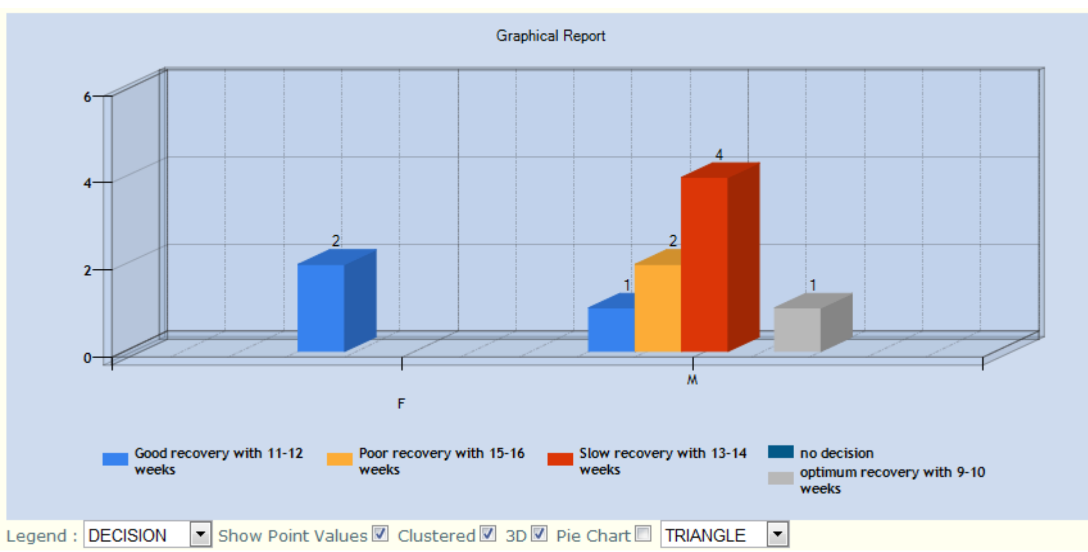
ID	FUZZYCATSETID	GRAPH	DEC_TRI	DEC_TRP
1	1	TRI-TRP	Poor recovery with 15-16 weeks	Poor recovery with 15-16 weeks
2	1	TRI-TRP	no decision	no decision
3	1	TRI-TRP	Poor recovery with 15-16 weeks	no decision
4	1	TRI-TRP	no decision	no decision
5	1	TRI-TRP	Good recovery with 11-12 weeks	Good recovery with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

Appendix J - Data Mining Examples

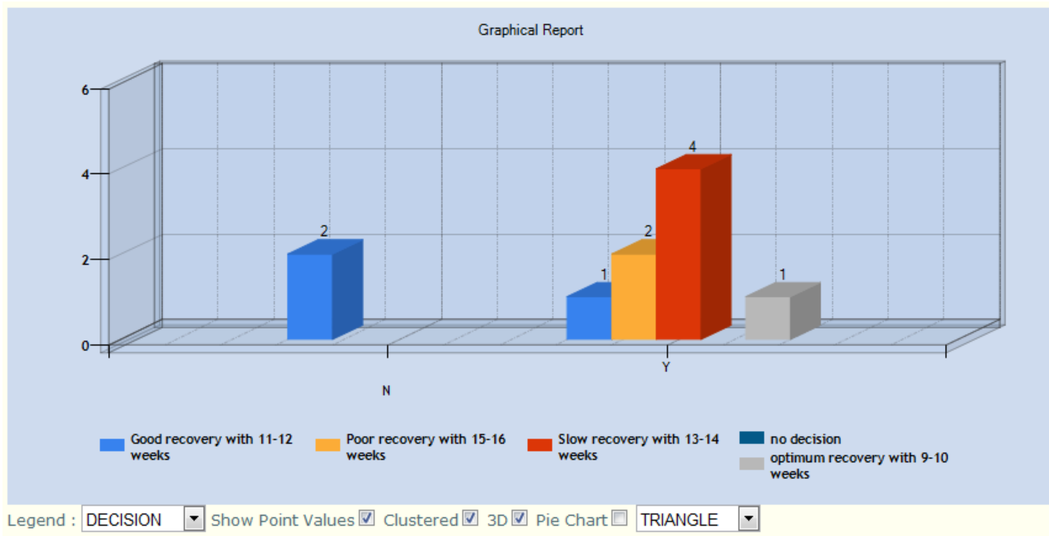
Decision about individual patients through visualization



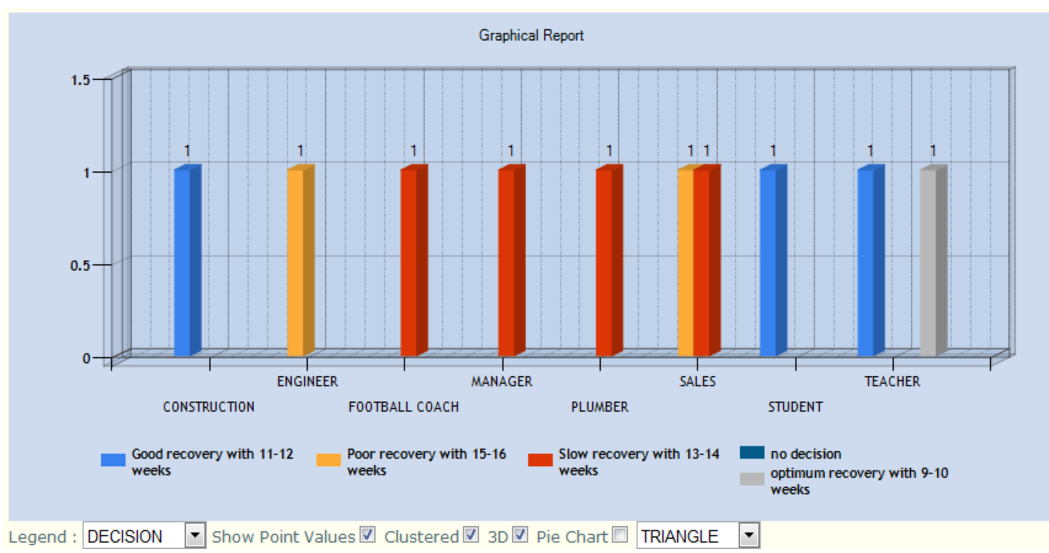
Overall decision based on gender



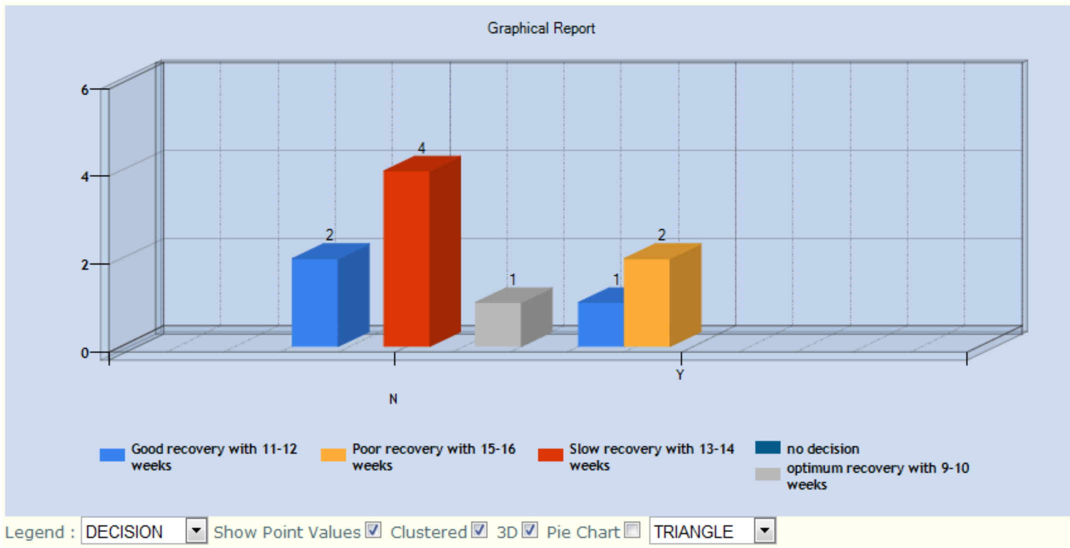
Decision based on patients with insurance



Overall decision based on each patient's occupancy



Overall decision based on each patient's smoking habits



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