MILKMINER: A DAIRY FARM ANALYSIS AND LEARNING SYSTEM

AYŞENUR GENÇ

IŞIK UNIVERSITY
2014

MILKMINER: A DAIRY FARM ANALYSIS AND LEARNING SYSTEM

AYŞENUR GENÇ
B.S., Computer Engineering, Gebze Institute of Technology, 2010

Submitted to the Graduate School of Arts and Sciences
in partial fulfillment of the requirements for the degree of
Master of Science
in
Information Technologies

IŞIK UNIVERSITY
2014

IŞIK UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

MILKMINER: A DAIRY FARM ANALYSIS AND LEARNING SYSTEM

AYŞENUR GENÇ

APPROVED BY:

Assist. Prof. Dr. Gülay Ünel ______________________
(Thesis Supervisor)

Assist. Prof. Dr. Cüneyt Sevgi ______________________

Assist. Prof. Dr. Ali İnan ______________________

APPROVAL DATE:

MILKMINER: A DAIRY FARM ANALYSIS AND LEARNING SYSTEM

Abstract

Agriculture and animal breeding industry is getting its share from the rapid advances in technology which enables the world wide use of automation systems. The use of automatic milking systems in dairy farms for milking has also increased.

The goal of this thesis is to contribute to the development of performance enhancing processes by analyzing the data collected in dairy farms for discovering new rules and relationships using data mining. The outputs of this thesis will be used by Triodor Company as an analysis and learning system.

Currently, the data collected in various dairy farms all around the world that use automated systems is stored for use in local or global databases as a part of the automation projects. Since these automation technologies are still in the development phase, research and development on detailed study, analysis and relationship recognition among data is in early stages and limited.

Currently, the central database populated by the software technologies developed by Triodor contains data about each individual farm such as key performance measurements collected daily from more than 30 countries and 4000 dairy farms for various types of users. This large scaled dataset is not used for any operation or analysis other than simple query answering. If this large database and external data sources (such as weather, vegetation) are analyzed for the detection of potential hidden relationships among data then it will be possible to realize improvements in these farms regarding various types of criteria such as performance, sustainability, and product quality.

In this thesis, an analysis and learning system that works on the data collected from dairy farms is developed. In the implementation of this system, the dairy farm database design is analyzed in terms of the content to be used in analysis purposes, and then the database is analyzed using data mining methods. As a result, a system with quantitative analysis techniques via data mining methods is generated.

Keywords: Dairy Farm Automation, Automatic Milking, Data Mining

MILKMINER: SÜT ÇİFTLİĞİ ANALİZ VE ÖĞRENME SİSTEMİ

Özet

Gittikçe hızlanan teknolojik ilerlemelere bağlı olarak tarım ve hayvancılık alanları da bu gelişmelerden nasibini almakta ve dünya genelinde ilgili birçok konuda otomasyon sistemleri kullanılmaktadır. Mandıra çiftliklerinde süt üretimi konusunda da son yıllarda otomatik süt üretim sistemlerinin (automatic milking systems) kullanımı hızla artmaktadır.

Bu tezin amacı mandıra çiftliklerinde toplanan verilerin veri madenciliği teknikleri kullanılarak işlenmesi, analiz edilmesi ve yeni kurallar ile ilişkilerin araştırılıp tespit edilerek çiftliklerin verimini artırıcı yeni süreçler geliştirilmesine katkı sağlamasıdır. Tezden elde edilecek olan çıktılar, Triodor firması tarafından analiz ve öğrenme sistemi olarak kullanılacaktır.

Hal-i hazırda dünya üzerinde otomasyon sürecine girmiş olan çeşitli mandıra çiftliklerinde ortaya çıkan veriler toplanarak, hem mandırada mevcut yerel veritabanlarında hem de otomasyon projeleri kapsamında merkezi veritabanlarında saklanmakta ve kullanılmaktadır. Bu tür teknolojiler halen gelişim sürecinde olduğu için veriler üzerinde detaylı inceleme, analiz ve veriler arası ilişki tespiti gibi konularda yapılan çalışmalar başlangıç sürecinde olup oldukça sınırlıdır.

Triodor firmasının geliştirdiği yazılım teknolojileri ile dünya üzerinde 30’dan fazla ülkede 4000’den fazla mandıra çiftliğinden, çiftliklerin her birinin çeşitli verileri ve anahtar performans ölçümleri bir sunucu üzerindeki merkezi veritabanında günlük bazda toplanıp kullanıcılara sunulmaktadır. Çok geniş ölçekteki bu veriler halen basit sorgulara cevap verme dışında başka herhangi bir işleme ve analize tabi tutulamamaktadır. Eğer bu geniş veritabanı ve çeşitli dış kaynaklı veri ve parametreler (hava durumu, bitki örtüsü vb.) analiz edilerek bunlar arasında saklı bulunan muhtemel ilişkiler tespit edilebilirse bu çiftliklerin süt verimi, üretim konusundaki sürdürülebilirliği, ürün kalitesi gibi birçok konuda önemli gelişmeler sağlanabilecektir.

Bu tez kapsamında mandıra çiftliklerinde toplanan veriler üzerinde çalışan bir analiz ve öğrenme sistemi oluşturulmuştur. Bu sistemin oluşumunda öncelikle merkezi mandıra çiftliği veritabanı tasarımı analizde kullanılacak içerik bakımından incelenmiş, daha sonra veritabanı üzerinde veri madenciliği yöntemleri ile analizler yapılmıştır. Sonuç olarak niceliksel analiz tekniklerini sunan bir sistem ortaya çıkmıştır.

Anahtar Kelimeler: Çiftlik Otomasyonu, Otomatik Süt Üretim Sistemleri, Veri Madenciliği

Acknowledgements

There are many people who helped to make my years at the graduate school most valuable. First, I thank Gülay Ünel, my major professor and dissertation supervisor. Having the opportunity to work with her over the years was intellectually rewarding and fulfilling.

Many thanks to Department of Information Technologies academic staff, who patiently answered my questions. I would also like to thank to my graduate student colleagues who helped me all through the years full of class work and exams.

The last words of thanks go to my family. I thank my parents Ayşe Genç, Nurettin Genç for their patience and encouragement. Lastly I thank my friend Serhat Durmaz, for his endless support through this long journey.

to my parents

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last quarter of the twentieth century, increasing labor costs in dairy farm products has led to the idea of automatic milking [1].

Decreasing milk prices and increasing input costs forced farmers to search methods for increasing production per man-hour. Thus, milk production automation has been used by commercial farms in the Netherlands in 1992 for the first time [2]. Advances in computer and control technology in the last twenty years provided new methods for automatic / semi automatic milking, feeding, and cleaning operations. The use of such high standards for milk producers does not only mean labor flexibility, but also more advanced social life. In addition to this, automation system devices and sensors provide important indicators such as ambient conditions, system performance, forage quality, animal health, and milk production. Manufacturers try to improve dairy farm conditions by examining these indicators which have a crucial role in order to increase quality, thus production amount is also increased. On the other hand, nowadays, there is a rapid decision-making necessity under the influence of global competition; increased size and complexity of system parameters make decision-making processes of the business difficult. Therefore, the importance of accurate and meaningful information is increasing under the influence of growing data stack.

The aim of the MILKMINER project is to develop an analysis and learning system on the data collected from dairy farms using data mining techniques in order to contribute to the productivity improvement milking processes and improvements on farm conditions. The data collected from farms are stored in a central database and used for querying purposes. Farm datasets are analyzed with the methods developed in the project which provide predictable or unpredictable patterns, relations, and rules.

Improvement of the methods and environmental conditions in milk and milk products sector may be possible by using the outputs of the project. Besides expected contributions to the industry, contributing to the academic knowledge is also among the objectives of the project.

Chapter 2

Related Work

In this section, main thesis related research areas are presented which are automatic milking and data mining.

2.1 Automatic Milking

Automatic milking [2, 3] focuses on milk production from dairy animals without human labour.

A considerable part of milk production is carried out automatically to decrease labour force with the advances in automatic milking. Correspondingly, a large number of farmers use the semiautomatic or automatic cow traffic control, and milk production machines and systems.

Since 1970, a considerable part of the studies and research about automatic milk production were on increasing production rate, voluntary milk production and decreasing work process at traditional dairy farms. Voluntary milk production determines time and period of milking according to the willingness of the animal. Automatic milk production systems require full automation in order to provide milking any time and any period within a day.

Since the beginning of 1990s, automatic milk production units are commercialized and there is a relative achievement in terms of voluntary milk production methods. A considerable part of the research and development in this area [4, 5, 6, 7, 8, 9, 10, 11, 12], is centered in Holland, and a large number of farms which have applied these methods, are in Denmark and Holland.

2.2 Data Mining

The size of accessible data has been increasing both over Web and on personal and institutional grounds as a result of technological developments in recent years. Devices such as cell phones, electronic household utensils and sensors are added to traditional data gatherers such as computers, therefore not only the size of data but also open data producing rate is increasing progressively. As a result, importance of information extraction from datasets and subsequently data mining, are increasing. There is a large number of data mining applications such as Web mining, finance, marketing, security, genetics in addition to agriculture and livestock breeding areas.

Detecting patterns from datasets is a research problem attacked by scientists throughout centuries. For instance, Bayes Theorem (1700s) and Regression Analysis (1800s) are traditional methods for obtaining patterns from datasets. Automatic data processing and analysis methods replaced traditional data analysis methodologies as a result of the innovations and developments in computer science. These innovations are developed by means of techniques on neural networks, clustering, genetic algorithms (1950s), decision trees (1960s), and vector support machines (1990s) in computer science. Data Mining makes use of techniques such as these for detecting hidden patterns from data. In other words, data mining is a process of searching and obtaining correlations, descriptions and predictions on large datasets [13].

There is a series of studies for developing standards for data mining. As an illustration, European Cross Industry Standard Process for Data Mining (CRISP-DM 1.0) and 2004 Java Data Mining standard (JDM 1.0) can be mentioned. Besides these studies, open source and free software systems are developed such as R language, Weka, KNIME, RapidMiner, jHepWork. All of these systems can use models implemented with PMML (Predictive Model Markup Language) designed for standard demonstration of data mining models among different applications [14].

Data mining methodologies are developed and used for application of agriculture and livestock breeding internationally [15, 16, 17, 18, 19]. A method developed on the use of data mining in milk production and clinical mastitis exploration [15] is one of the works that is most related to our project. This study has a limited scope compared to our work because the data used belongs to 1109 cows from 9 dairy farms and data is analyzed for only clinical mastitis purpose. Another related study is on the usage of data mining software on health problems of animals [18]. Purpose of this study is examining existing data mining software instead of developing a new system. In addition to these studies, there are also applications of data mining methodologies on agricultural areas [16, 17, 19].

Chapter 3

Database Analysis and Improvements

MILKMINER database is populated by the software technologies developed by Triodor, and stores data collected from more than 30 countries and 4000 dairy farms for various types of users all around the world that use automated systems.

In our project, MILKMINER database is used for analysis purposes in order to detect potential hidden relationships among data. This database is analyzed to make potential improvements in farms possible regarding various types of criteria such as:

- performance
- sustainability
- product quality

The tables and relationships among the data in MILKMINER database are given in Figure 3.1 and Figure 3.2. Database is created using MS SQL Server 2008 R2.

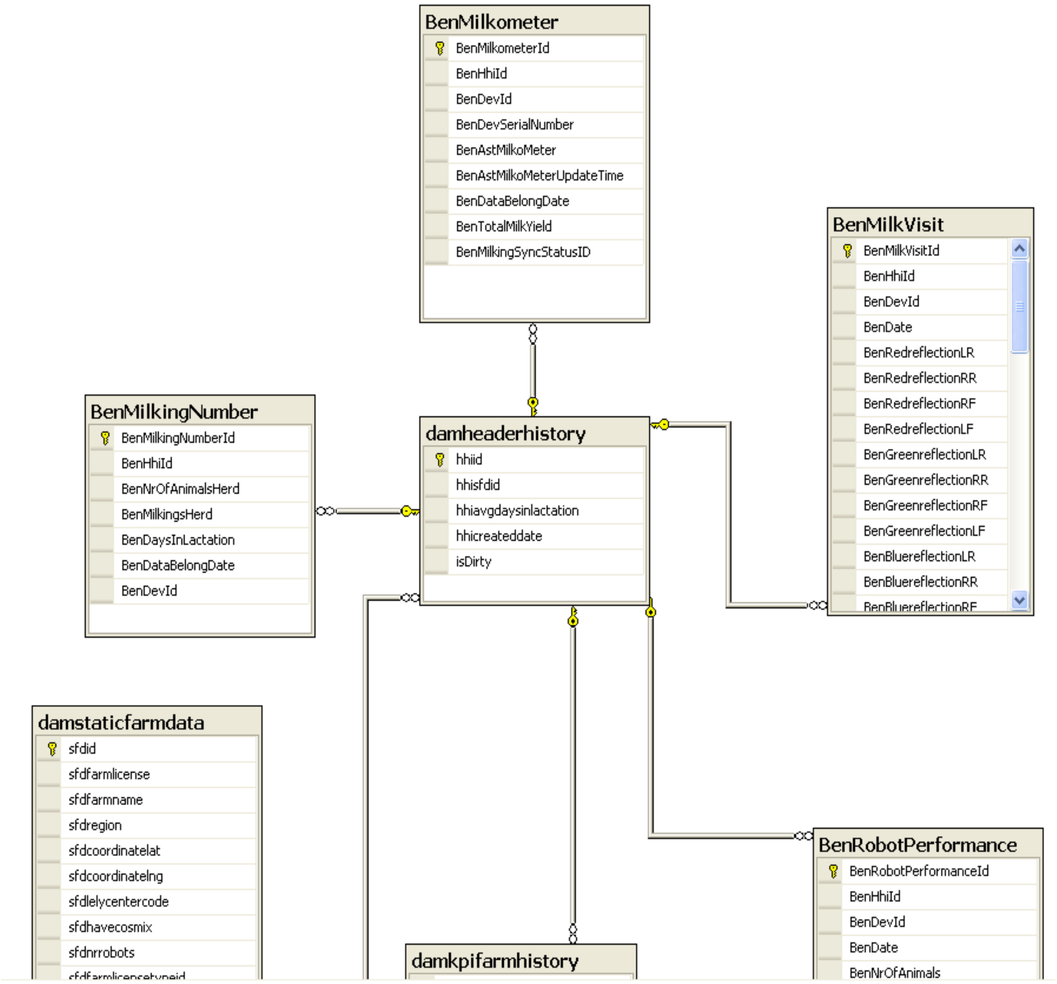


Figure 3.1 MILKMINER Database Diagram Part 1

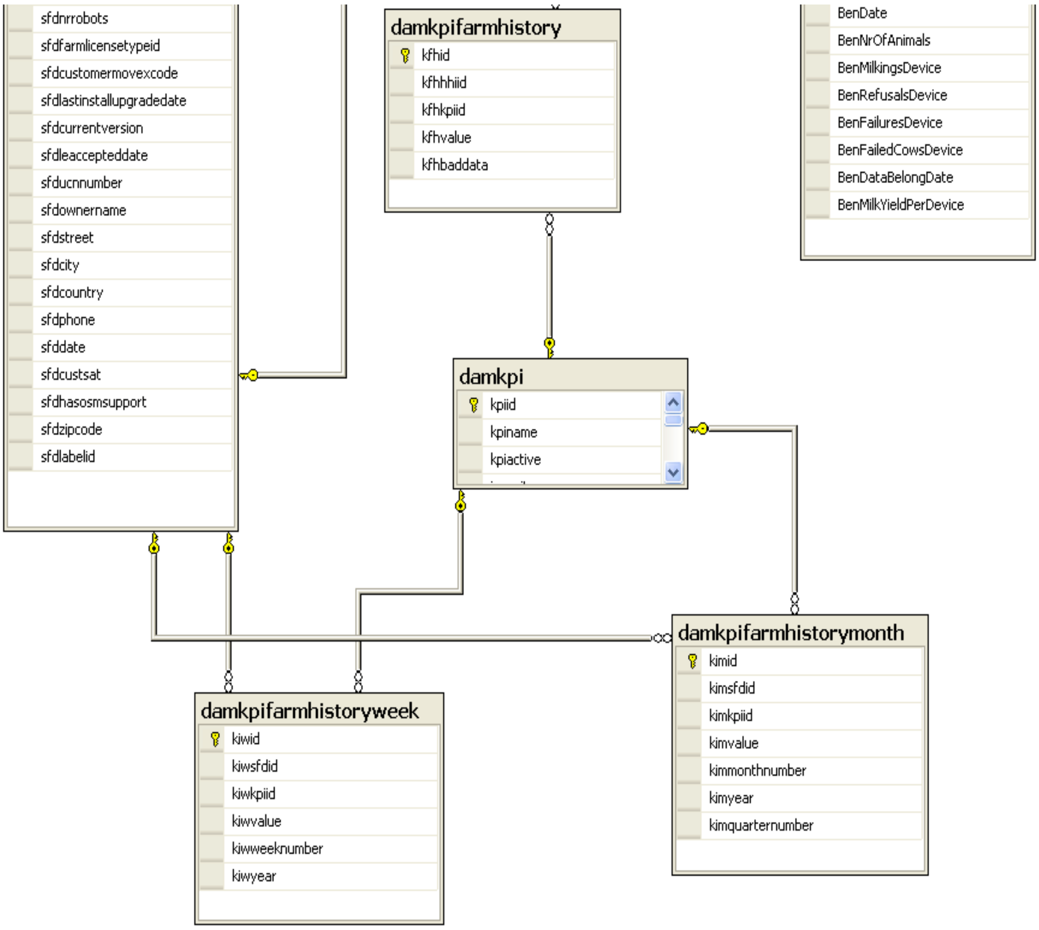


Figure 3.2 MILKMINER Database Diagram Part 2

Important details about the tables for our study are provided below.

BenMilkometer

The number of robot based milking and the total amount of milk are stored in this table on a daily basis. Descriptions of the attributes which are important for our analysis purposes are provided below.

BenDataBelongDate: stores the date of the data collected.
BenAstMilkoMeter: stores total milking number of a robot until the data collection date.

BenTotalMilkYield: stores total amount of milk a robot milked until the data collection date.

BenMilkingNumber

The table stores dairy farm based milking data. Descriptions of the attributes which are important for our analysis purposes are provided below.

BenDataBelongDate: stores the date of the data collected.
BenNrOfAnimalsHerd: stores the number of individual cows milked on the data collection date.
BenMilkingsHerd: stores the number of milking in the farm on the data collection date.

Damheaderhistory:

The table stores the date and validity of the data on a daily basis. Descriptions of the attributes which are important for our analysis purposes are provided below.

Hhiavgdaysinlactation: stores the average of milk lactation period of animals in the dairy farm.
isDirty: stores the validity of KPIs (Key Performance Indicator).

BenMilkVisit:

The table stores robot based daily milking process information. The columns ending with LR, RR, RF and LF store content based information about the milk collected by the sensors of the robot.

BenDate: stores the date of the data collected.

Damstaticfarmdata:

The table stores basic data about the dairy farm. Descriptions of some of the attributes which are important for our analysis purposes are provided below.

Sfdregion: stores the country/area in which the dairy farm is located.
Sfdnrrobots: stores the number of milking robots in the dairy farm.

BenRobotPerformance:

The table stores information about the daily processes of the robots. Descriptions of some of the attributes which are important for our analysis purposes are provided below.

BenDataBelongDate: stores the date of the data collected.
BenNrOfAnimals: stores the number of individual cow milked by the robot on the data collection date.
BenMilkingsDevice: stores number of milking processes that are not refused by the robot on to the data collection date.
BenRefusalsDevice: stores number of milking processes that are refused by the robot on to the data collection date.
BenFailuresDevice: stores the number of failed milking processes by the robot on to the data collection date.
BenMilkYieldPerDevice: stores total amount of milk collected by the robot on the data collection date.

Damkpifarmhistory:

The table stores dairy farm based daily values of KPIs.

Damkpifarmhistoryweek:

The table stores weekly average values of KPIs.

Damkpifarmhistorymonth:

The table stores monthly average values of KPIs.

Damkpi:

The table stores the list of KPIs.

The tables that store data about KPIs which measure performance are critical tables for analysis purposes.

In the analysis process, instead of analyzing superficial information, analyzing detailed information can lead to more specific results. For instance, instead of climate information, storing vegetation data is more beneficial in terms of specifying principal parameters.

Data which that can be added to the database for potentially extracting more information are given below.

- The geographical location and climate of the area where the farm is located.
- Farm conditions (e.g. parameters that determine the cleanliness of the farm).
- Population details.

Under these main topics, the following parameters may be included:

- Weather conditions (temperature, humidity),
- Distance from the sea, proximity to the sea
- Altitude information,
- Vegetation,
- Barn calcification period,
- Cow disease rate,
- Cow in the open air and indoor air rates,
- Whether music is used in the farms or not,
- Type of the bait,
- Ratio of male cows to female cows.

The additional data that can be collected will be revisited as a future work of our study.

Chapter 4

System Requirements

System requirements in the scope of the project MILKMINER include following software and their required hardware features.

- **Sql Server 2008 R2**
- **Weka**
- **Eclipse**

4.1 Descriptions

Short descriptions for each requirement are given below:

Sql Server 2008 R2

This database management system is used to manage Benchmark datasets on which Weka program works.

Weka

WEKA (Waikato Environment for Knowledge Analysis) is a widely used worldwide open source machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka is a free software available under the GNU General Public License.

The following functionalities are in the scope of WEKA:

- Data Preprocessing
- Data Classification
- Data Clustering
- Data Association Mining

Appropriate algorithms are selected and applied to datasets obtained from the Benchmark database and selected algorithms are implemented as modules to our analysis system.

Eclipse

"In computer programming, Eclipse is an Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins" [20, 21].

4.2 Hardware Requirements

Server hardware requirements are as given in Table 4.1.

Table 4.1 – Hardware requirements for server

Component	Requirement
Hard Disk	Disk space requirements will vary with the SQL Server 2008 R2 components you install.
Driver	A CD or DVD drive, as appropriate, is required for installation from disc.
Display	SQL Server 2008 R2 graphical tools require Super VGA or higher resolution: at least 800x600 pixel resolution.
Other Devices	Pointing device: A Microsoft mouse or compatible pointing device is required.

Weka

Weka does not have strict hardware requirements; it can run on both 32 bit and 64 bit operating systems.

Eclipse

Eclipse hardware requirements are as follows:

- MACOS Leopard - 64 bit
- Windows XP ,Vista, Windows 7 - 32/64 bit
- Oracle Enterprise Linux 5 - 32/64 bit
- RedHat Linux 5 - 32/64 bit
- Recommend 2GB memory for IDE and 2GB if running server locally

4.3 Software Requirements

Sql Server 2008 r2

Server Software requirements are as provided in Table 4.2.

Table 4.2 – Software requirements for server [22]

Component	Requirement
Framework	SQL Server Setup installs the following software components required by the product: • .NET Framework 3.5 SP1 • SQL Server Native Client SQL Server Setup support files

Software	SQL Server Setup requires Microsoft Windows Installer 4.5 or a later version After installing required components, SQL Server Setup will verify that the computer where SQL Server 2008 R2 will be installed also meets all the other requirements for a successful installation.
Network software	Network software requirements for the 64-bit versions of SQL Server 2008 R2 are the same as the requirements for the 32-bit versions. Supported operating systems have built-in network software. Stand-alone named and default instances support the following network protocols: • Shared memory • Named Pipes • TCP/IP • VIA **Note** Shared memory and VIA are not supported on failover clusters. **Note** The VIA protocol is deprecated. This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.
Virtualization	SQL Server 2008 R2 is supported in virtual machine environments running on the Hyper-V role in Windows Server 2008 SP2 Standard, Enterprise and Datacenter editions. The virtual machine must run an operating system supported for the specific SQL Server 2008 R2 edition listed later in this topic. In addition to resources required by the parent partition, each virtual machine (child partition) must be provided with sufficient processor resources, memory, and disk resources for its SQL Server 2008 R2 instance.

	Within the Hyper-V role on Windows Server 2008 SP2, a maximum of four virtual processors can be allocated to virtual machines running Windows Server 2008 SP2 32-bit or 64-bit editions. A maximum of 2 virtual processors can be allocated to virtual computers that are running Windows Server 2003 32-bit editions. For virtual computers that host other operating systems, a maximum of one virtual processor can be allocated to virtual computers.
Internet Software	Microsoft Internet Explorer 6 SP1 or a later version is required for all installations of SQL Server 2008 R2. Internet Explorer 6 SP1 or a later version is required for Microsoft Management Console (MMC), SQL Server Management Studio, Business Intelligence Development Studio, the Report Designer component of Reporting Services, and HTML Help.

Weka

Weka software requirements are given below.

- 1.4, 1.5, 1.6 or 1.7 Java versions are necessary to run 3.4 Weka version.
- 1.6 and 1.7 Java version is necessary to run 3.5 Weka version (for the subversion 3.5.0 and 3.52, 1.4 Java version can also be used).
- 1.5, 1.6 or 1.7 Java version is necessary to run 3.6 Weka version.
- 1.7 Java version is necessary to run 3.7 Weka version (for the subversion 3.7.0, 1.5 Java version can also be used).

Java 5.0 and later in combination with Linux/Gnome has problems with the default Look'n'Feel. There is a workaround for this problem which was introduced with version 3.4.5/3.5.0. From Weka 3.6.5/3.7.4 Mac OS X users will need Java for Mac OS X 10.6 Update 3 (java 1.6.0_22) or later [23].

Eclipse

Eclipse runs on any computer that has a functional Windows 7, Vista, XP Pro, or 2008 Server operating system with all Microsoft Service Packs and updates installed.

4.4 Data Mining Module System Requirements

System requirements includes the following features:

- Eclipse Java EE IDE.
- JBoss Maven Integration 1.0
- JavaServer Faces 2.0
- JavaScript 1.0
- Java 1.6
- RichFaces
- Weka library
- Tomcat v7.0 Server

Short descriptions for these requirements are given below:

4.4.1 Eclipse Java EE IDE

The Eclipse IDE for Java EE Developers contains everything you need to build Java and Java Enterprise Edition (Java EE) applications. The Eclipse IDE for Java EE Developers provides superior Java editing with incremental compilation, Java Enterprise Edition support, a graphical HTML/JSP/JSF editor, database management tools, and support for most popular application servers [20].

4.4.2 JBoss Maven Integration

“Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information." [27].

4.4.3 JavaServer Faces

JavaServer Faces (JSF) is a development framework in order to facilitate Java-based Web application development based on the structure of Model-View-Controller (MVC).

4.4.4 JavaScript

JavaScript is a client-side interpreted scripting language.

4.4.5 Java

Java is an open source, object oriented, platform independent, highly efficient, multi task, high level, interpreted language developed by Sun Microsystems engineer James Gosling.

4.4.6 RichFaces

RichFaces is an open source library unified with ajax4jsf which is a project for adding Ajax capabilities to the JavaServer Faces Web application framework [28, 29].

4.4.7 Weka Library

This is a library with a large number of implemented functions for mining datasets.

4.4.8 Tomcat v7.0 Server

Tomcat is a Servlet container and a Java Server Page application program.

Chapter 5

Data Mining Techniques Used in MILKMINER

Data mining is a process converting huge data sets to qualified information using statistical analysis techniques and artificial intelligence algorithms, in addition to other methods developed in the area. In order to apply data mining techniques, open source and commercial applications are available.

In MILKMINER project, the algorithms implemented in Weka are used as data mining techniques. Weka has filters for data cleaning, and conversion in addition to data classification, clustering, association mining functionalities.

Data mining techniques used in the context of this application program are explained below:

5.1 Data Cleaning: Data Cleaning is an important and time consuming step in data mining processes. Before importing data to WEKA, empty or unqualified data are extracted from a dataset using data cleaning techniques, thus increasing the accuracy of the results can be possible.

5.2 Data Conversion: After the data cleaning step, data conversion step begins. The filters used in data conversion process in the scope of MILKMINER project, are listed below:

WEKA.filters.Filter
WEKA.filters.supervised.attribute.Discretize

As stated in [23], this is “an instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes. Discretization is by simple binning. Skips the class attribute if set.”

WEKA.filters.Filter
WEKA.filters.unsupervised.attribute.Remove

As stated in [23], this is “an filter that removes a range of attributes from the dataset. Will re-order the remaining attributes if invert matching sense is turned on and the attribute column indices are not specified in ascending order.”

WEKA.filters.Filter
WEKA.filters.SimpleFilter
WEKA.filters.SimpleBatchFilter
WEKA.filters.unsupervised.attribute.InterquartileRange

As stated in [23], this is “a filter for detecting outliers and extreme values based on interquartile ranges. The filter skips the class attribute.”

5.3 Classification – clustering – association mining : After data cleaning and conversion, purposive classification, clustering, and association mining algorithms can be used.

In MILKMINER project we especially focus on association minig and classification algorithms.

5.3.1 Classification

Classification technique is a supervised learning method. Number of classes is known. Each sample in the dataset is represented by the values of a set of attributes where one of these attributes is used as a class label. A model is constructed using a training set, and tested using a test set for determining the accuracy of the model.

The classification method from WEKA used in the scope of this project is given below:

java.lang.Object
WEKA.classifiers.AbstractClassifier
WEKA.classifiers.trees.m5.M5Base
WEKA.classifiers.trees.M5P

“M5Base implements base routines for generating M5 Model trees and rules. M5Base is just the base class used by both M5P (tree learner) and M5Rules.” [24]

5.3.2 Clustering

Clustering is an unsupervised learning method. Clustering is the process of distributing data into different groups according to a predefined similarity/distance measure.

In scope of this Project, the clustering method used from WEKA is given below:

WEKA.clusterers.AbstractClusterer
WEKA.clusterers.RandomizableClusterer
WEKA.clusterers.SimpleKMeans

“This method clusters data using the k-means algorithm. It can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean.” [25]

5.3.3 Association Mining

Association mining is a method for discovering interesting relations and patterns between attributes/variables in large datasets.

In the scope of this project, the association mining method used from WEKA is provided below:

WEKA.associations.AbstractAssociator
WEKA.associations.Apriori

“This is a class from WEKA implementing an Apriori-type algorithm. The method iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence.” [26]

The algorithm has an option to mine class association rules. The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

Chapter 6

Development of Data Mining Module

Data mining module is designed and implemented in the scope of the MILKMINER project.

6.1 Purpose

In this study, general purpose of the data mining module is providing a plain and simple user interface for the selected algorithms in WEKA. Data mining module is a dynamic Web project providing effective usage by determining functionalities needed and implementing these functionalities. The algorithm selection, file selection and the results are available in the interface. Operation of the system is demonstrated in the diagram below:

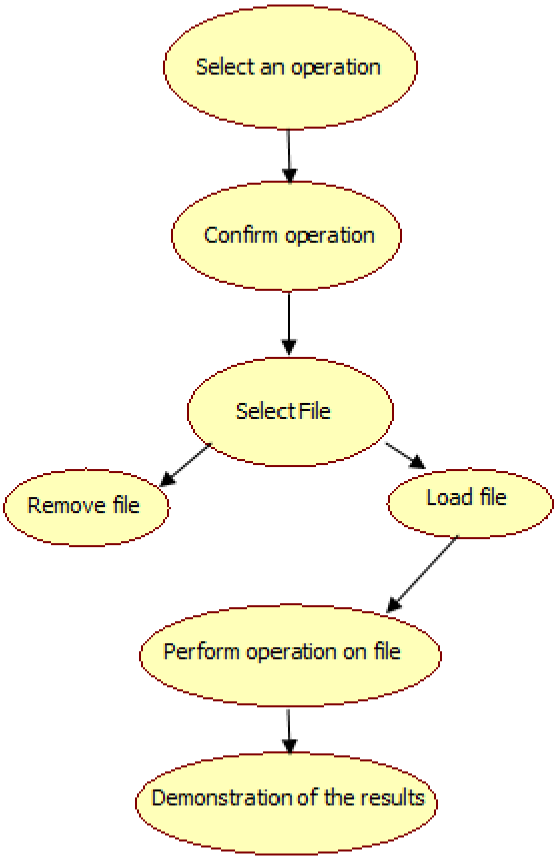


Figure 6.1 System Operation Flow

6.2 System Properties

There are 7 user actions in this module as listed below:

- Select the algorithm to be applied.
- Click "confirm the algorithm" button
- Select the data in a file with specified format on which the will be applied
- Load the file
- Remove the file selected
- Remove the file loaded
- Click "apply the algorithm" button

After the basic steps, results of the algorithm over the selected dataset are given to the user in the result area of the form. The details of the user actions and responses are provided below.

Use Case diagram of the system is demonstrated by the following figure.

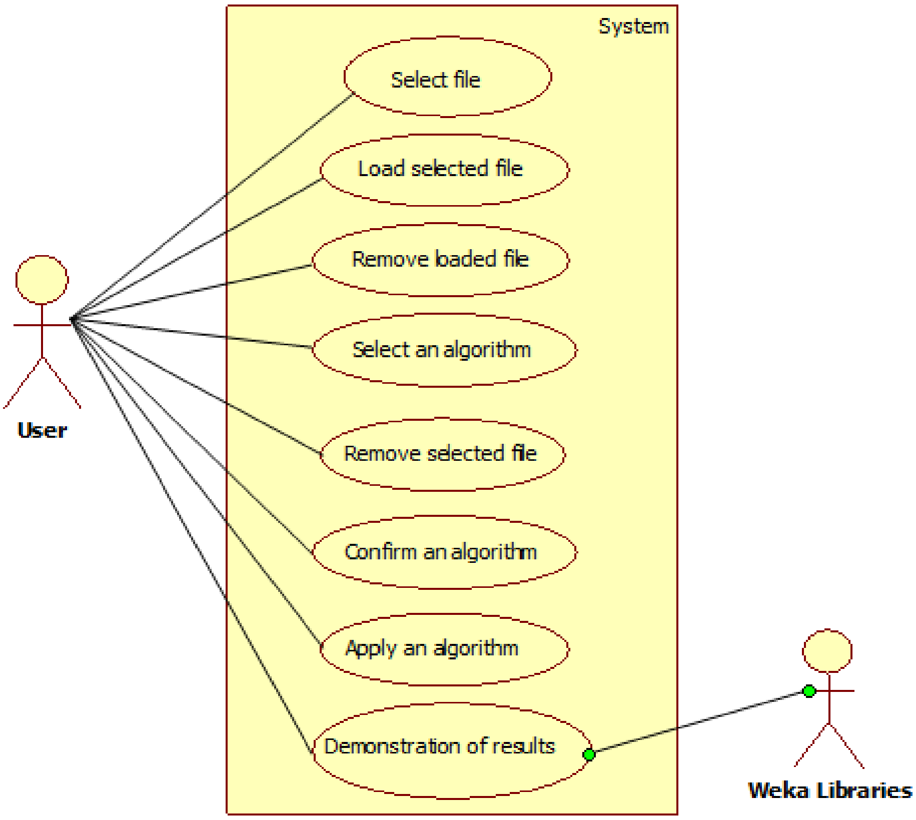


Figure 6.2 Use Case Diagram

System has 4 classes/interfaces. In the following table, class names and descriptions are given:

Table 6.1 – Class names and descriptions

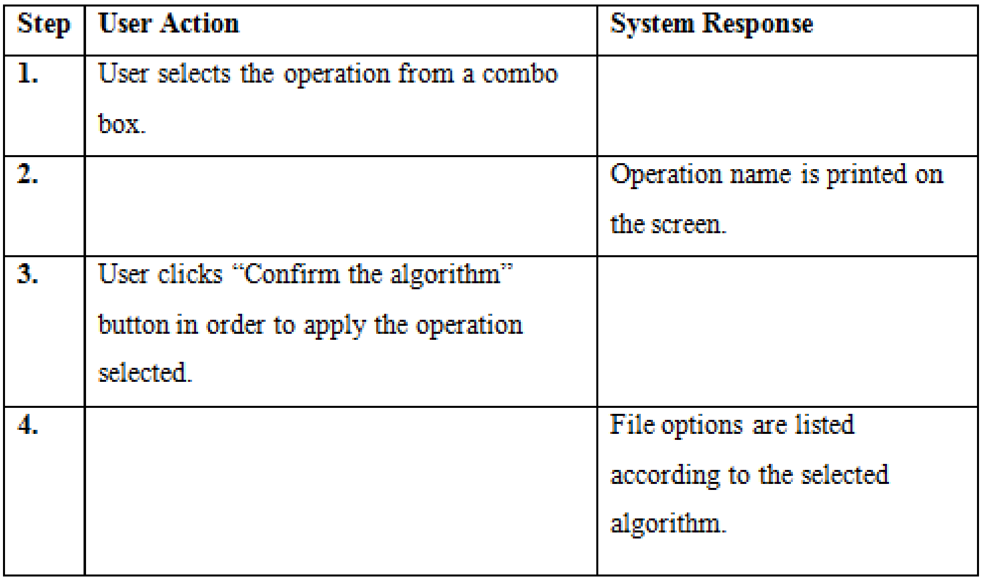
Class/Interface	**Description**
User Interface	Provides the interface of the system to the user.
ApprioriService	Provides the service class which contains the Apriori algorithm process.
M5pPredictorService	Provides service class which contains the m5p prediction algorithm process.
FarmBean	Contains functions as a backend service of the user interface.

6.3 User Actions and System Responses

In this section possible user actions and system responses are presented.

- **Select and confirm an operation to be applied**: In this action, user selects the operation and confirms it, system presents file options according to selected algorithm.

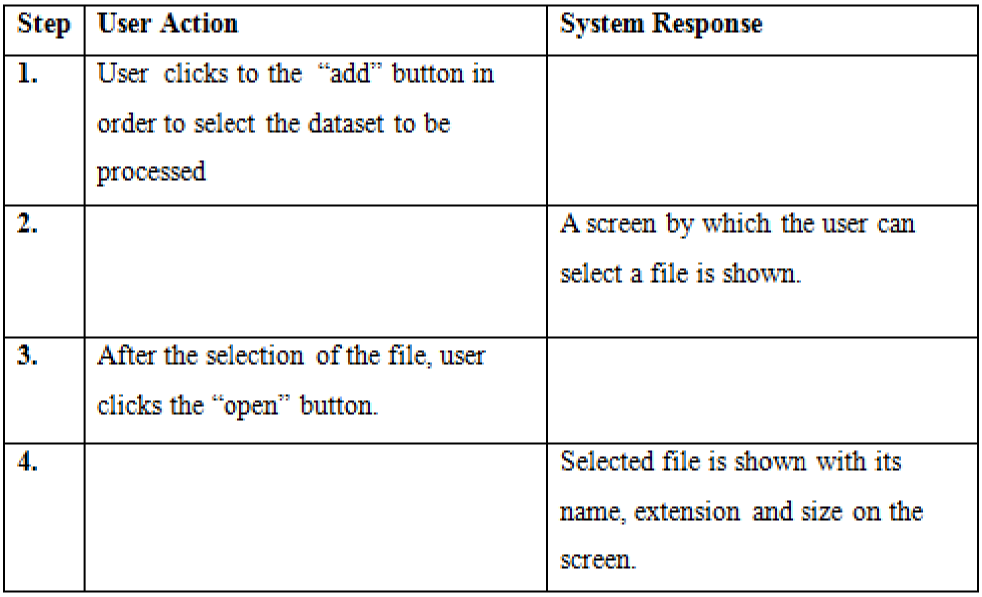
Table 6.2 – Select and confirm an operation



Step	User Action	System Response
1.	User selects the operation from a combo box.	
2.		Operation name is printed on the screen.
3.	User clicks "Confirm the algorithm" button in order to apply the operation selected.	
4.		File options are listed according to the selected algorithm.

- **Select file including data set:** In this action, user specifies the dataset to be analyzed.

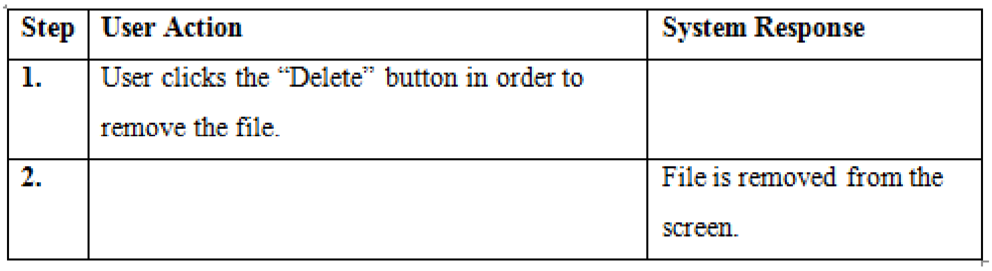
Table 6.3 – Select file including data set



Step	User Action	System Response
1.	User clicks to the "add" button in order to select the dataset to be processed	
2.		A screen by which the user can select a file is shown.
3.	After the selection of the file, user clicks the "open" button.	
4.		Selected file is shown with its name, extension and size on the screen.

- **Remove the file including data set action:** In this action, user deletes the selected file.

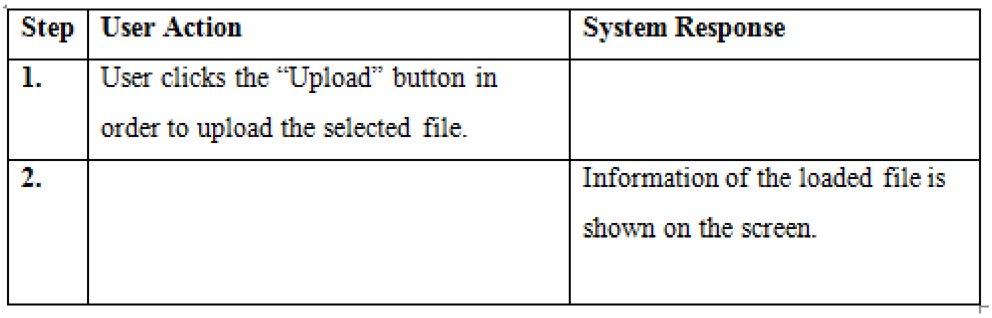
Table 6.4 – Remove the file including data set



Step	User Action	System Response
1.	User clicks the "Delete" button in order to remove the file.	
2.		File is removed from the screen.

- **Load the selected file to the system action:** In this action, user uploads the file; subsequently sees the information of the loaded file.

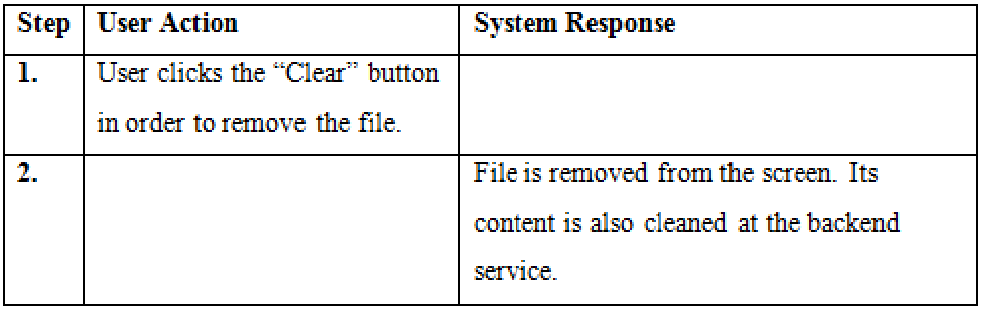
Table 6.5 -- Load the selected file



Step	User Action	System Response
1.	User clicks the "Upload" button in order to upload the selected file.	
2.		Information of the loaded file is shown on the screen.

- **Remove the loaded file action:** In this action, user removes the loaded file.

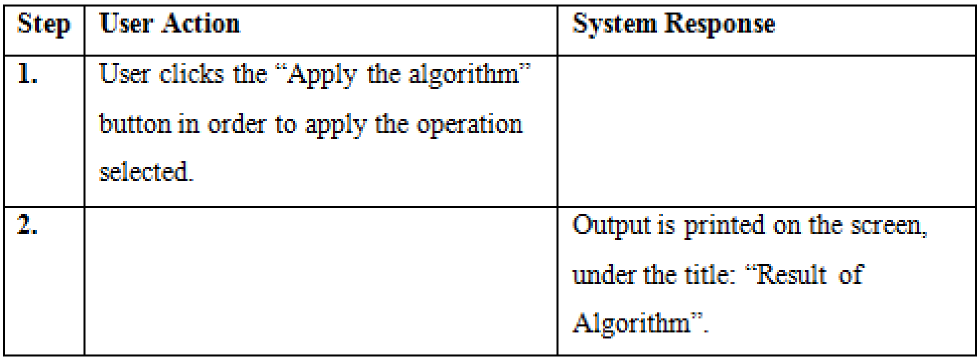
Table 6.6 -- Remove the loaded file



Step	User Action	System Response
1.	User clicks the "Clear" button in order to remove the file.	
2.		File is removed from the screen. Its content is also cleaned at the backend service.

- **Apply an operation, show result action:** In this action, user applies selected the methodology on the loaded file, subsequently sees the results on screen.

Table 6.7 -- Apply an operation



Step	User Action	System Response
1.	User clicks the “Apply the algorithm” button in order to apply the operation selected.	
2.		Output is printed on the screen, under the title: “Result of Algorithm”.

6.4 UML Class Diagram

UML Class Diagram of the system is demonstrated by the following figure.

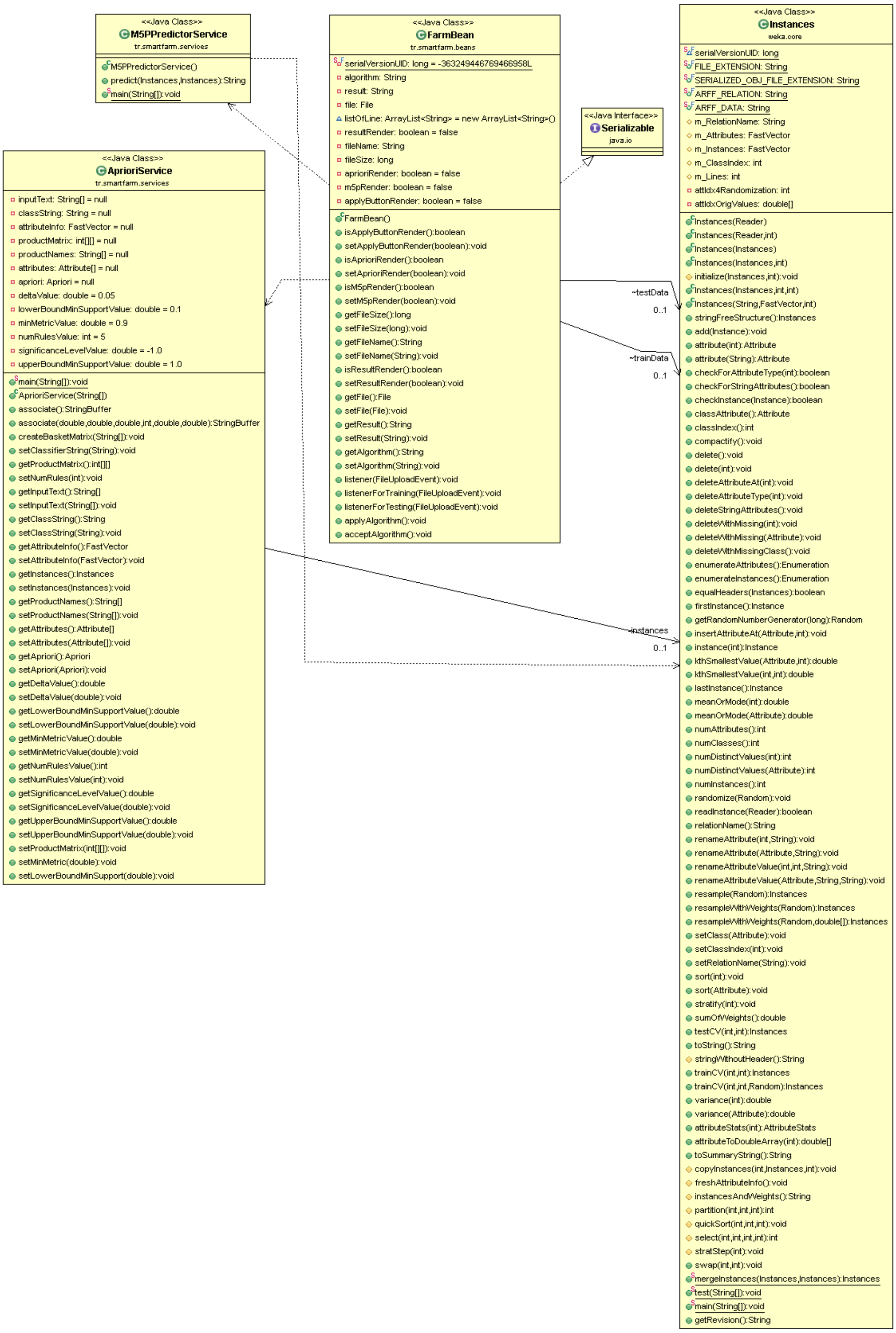

<<Java Class>>
M5PPredictorService
tr.smartfarm.services
M5PPredictorService()
predict(Instances,Instances):String
main(String[]):void
<<Java Class>>
FarmBean
tr.smartfarm.beans
serialVersionUID: long = -363249446769466958L
algorithm: String
result: String
file: File
listOfLine: ArrayList<String> = new ArrayList<String>()
resultRender: boolean = false
fileName: String
fileSize: long
aprioriRender: boolean = false
m5pRender: boolean = false
applyButtonRender: boolean = false
FarmBean()
isApplyButtonRender():boolean
setApplyButtonRender(boolean):void
isAprioriRender():boolean
setAprioriRender(boolean):void
isM5pRender():boolean
setM5pRender(boolean):void
getFileSize():long
setFileSize(long):void
getFileName():String
setFileName(String):void
isResultRender():boolean
setResultRender(boolean):void
getFile():File
setFile(File):void
getResult():String
setResult(String):void
getAlgorithm():String
setAlgorithm(String):void
listener(FileUploadEvent):void
listenerForTraining(FileUploadEvent):void
listenerForTesting(FileUploadEvent):void
applyAlgorithm():void
acceptAlgorithm():void
<<Java Interface>>
Serializable
java.io
<<Java Class>>
AprioriService
tr.smartfarm.services
inputText: String[] = null
classString: String = null
attributeInfo: FastVector = null
productMatrix: int[][] = null
productNames: String[] = null
attributes: Attribute[] = null
apriori: Apriori = null
deltaValue: double = 0.05
lowerBoundMinSupportValue: double = 0.1
minMetricValue: double = 0.9
numRulesValue: int = 5
significanceLevelValue: double = -1.0
upperBoundMinSupportValue: double = 1.0
main(String[]):void
AprioriService(String[])
associate():StringBuffer
associate(double,double,double,int,double,double):StringBuffer
createBasketMatrix(String[]):void
setClassifierString(String):void
getProductMatrix():int[][]
setNumRules(int):void
getInputText():String[]
setInputText(String[]):void
getClassString():String
setClassString(String):void
getAttributeInfo():FastVector
setAttributeInfo(FastVector):void
getInstances():Instances
setInstances(Instances):void
getProductNames():String[]
setProductNames(String[]):void
getAttributes():Attribute[]
setAttributes(Attribute[]):void
getApriori():Apriori
setApriori(Apriori):void
getDeltaValue():double
setDeltaValue(double):void
getLowerBoundMinSupportValue():double
setLowerBoundMinSupportValue(double):void
getMinMetricValue():double
setMinMetricValue(double):void
getNumRulesValue():int
setNumRulesValue(int):void
getSignificanceLevelValue():double
setSignificanceLevelValue(double):void
getUpperBoundMinSupportValue():double
setUpperBoundMinSupportValue(double):void
setProductMatrix(int[][]):void
setMinMetric(double):void
setLowerBoundMinSupport(double):void
~testData
0..1
~trainData
0..1
-instances
0..1
<<Java Class>>
Instances
weka.core
serialVersionUID: long
FILE_EXTENSION: String
SERIALIZED_OBJ_FILE_EXTENSION: String
ARFF_RELATION: String
ARFF_DATA: String
m_RelationName: String
m_Attributes: FastVector
m_Instances: FastVector
m_ClassIndex: int
m_Lines: int
attIdx4Randomization: int
attIdxOrigValues: double[]
Instances(Reader)
Instances(Reader,int)
Instances(Instances)
Instances(Instances,int)
initialize(Instances,int):void
Instances(Instances,int,int)
Instances(String,FastVector,int)
stringFreeStructure():Instances
add(Instance):void
attribute(int):Attribute
attribute(String):Attribute
checkForAttributeType(int):boolean
checkForStringAttributes():boolean
checkInstance(Instance):boolean
classAttribute():Attribute
classIndex():int
compactify():void
delete():void
delete(int):void
deleteAttributeAt(int):void
deleteAttributeType(int):void
deleteStringAttributes():void
deleteWithMissing(int):void
deleteWithMissing(Attribute):void
deleteWithMissingClass():void
enumerateAttributes():Enumeration
enumerateInstances():Enumeration
equalHeaders(Instances):boolean
firstInstance():Instance
getRandomNumberGenerator(long):Random
insertAttributeAt(Attribute,int):void
instance(int):Instance
kthSmallestValue(Attribute,int):double
kthSmallestValue(int,int):double
lastInstance():Instance
meanOrMode(int):double
meanOrMode(Attribute):double
numAttributes():int
numClasses():int
numDistinctValues(int):int
numDistinctValues(Attribute):int
numInstances():int
randomize(Random):void
readInstance(Reader):boolean
relationName():String
renameAttribute(int,String):void
renameAttribute(Attribute,String):void
renameAttributeValue(int,int,String):void
renameAttributeValue(Attribute,String,String):void
resample(Random):Instances
resampleWithWeights(Random):Instances
resampleWithWeights(Random,double[]):Instances
setClass(Attribute):void
setClassIndex(int):void
setRelationName(String):void
sort(int):void
sort(Attribute):void
stratify(int):void
sumOfWeights():double
testCV(int,int):Instances
toString():String
stringWithoutHeader():String
trainCV(int,int):Instances
trainCV(int,int,Random):Instances
variance(int):double
variance(Attribute):double
attributeStats(int):AttributeStats
attributeToDoubleArray(int):double[]
toSummaryString():String
copyInstances(int,Instances,int):void
freshAttributeInfo():void
instancesAndWeights():String
partition(int,int,int):int
quickSort(int,int,int):void
select(int,int,int,int):int
stratStep(int):void
swap(int,int):void
mergeInstances(Instances,Instances):Instances
test(String[]):void
main(String[]):void
getRevision():String

Figure 6.3 UML Class Diagram

6.5 System Architecture

MILKMINER application runs in a Java servlet container and contains following parts:

- Model including application specific functionality and data.
- Controller performing user actions.
- View presenting user interface.
- Xml file containing application configuration resource.

MILKMINER application uses JSF library and Tags which contain event handlers and validators and rendering UI components.

File is a necessary input to analyze dataset. In order to apply the algorithms, extension of the file to be loaded should be “csv” which means that the file should contain the attribute names and values should be in comma separated or space separated format.

System architecture is shown in Figure 6.4

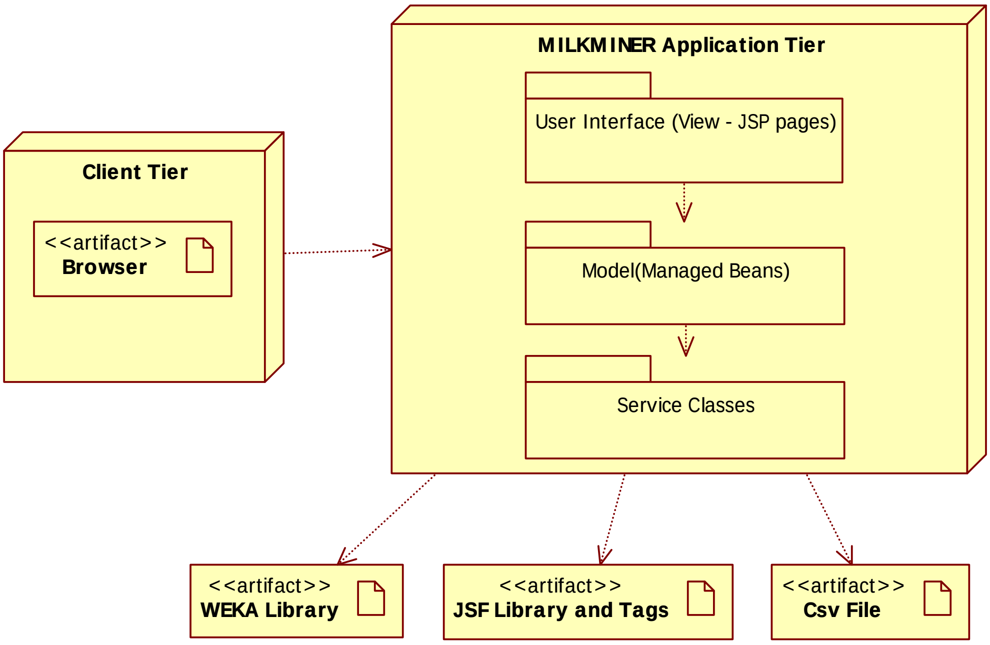


Figure 6.4 System Architecture

.

6.6 Algorithms Used in Application

Two Algorithms are used in MILKMINER application. One of them is Apriori which is the most popular association mining algorithm having an implementation in Weka. The other algorithm is M5p which is a classification algorithm for making prediction on numeric attributes. In this section, these two algorithms are explained in detail.

6.6.1 Apriori

Apriori is a classic algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis [30].

There are advantages of this algorithm such as ability of processing large itemsets and ease of implementation. Besides these advantages, there are also disadvantages such as requiring a large amount of memory when the data size is large, and inefficiency.

6.6.2 M5p

This algorithm implements base routines for generating M5 Model trees and rules [24].

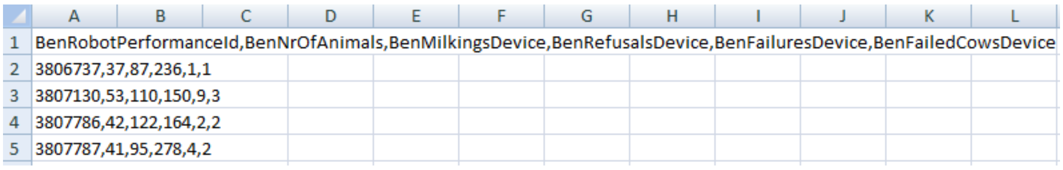
M5 is a method developed by Quinlan for inducing trees of linear regression models (model trees). “M5 model trees family is an accurate data-driven modelling approach leading to transparent models that can be easily understood by the decision makers.” [31].

M5P model tree is a decision tree with linear regression functions at the leaves.

This model can be used to predict a numeric class attribute. M5P tree algorithm can also deal with categorical and continuous variables and variables with missing values [32].

6.7 Application Inputs, Outputs and User Interface

There are two algorithms used in the data mining module. Application requires a file in CSV format as an input in order to execute Apriori algorithm and requires two files in CSV format in order to execute M5p where one of them is a training set, and the other one is a test set. An example input file is provided in Figure 6.5

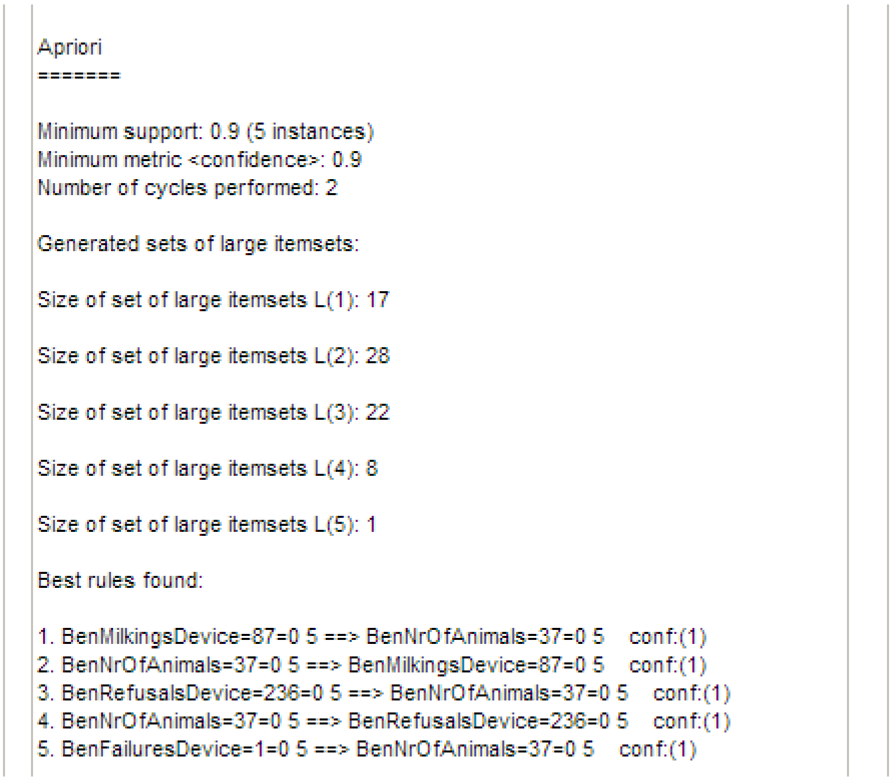


	A
1	BenRobotPerformanceId,BenNrOfAnimals,BenMilkingsDevice,BenRefusalsDevice,BenFailuresDevice,BenFailedCowsDevice
2	3806737,37,87,236,1,1
3	3807130,53,110,150,9,3
4	3807786,42,122,164,2,2
5	3807787,41,95,278,4,2

Figure 6.5 An Example Input File

Most of the database management systems have capabilities for exporting data in CSV format.

Apriori algorithm tries to find the best rules possible according to the given parameters. After the algorithm is run on an example input file, we get results in the format shown in Figure 6.6.

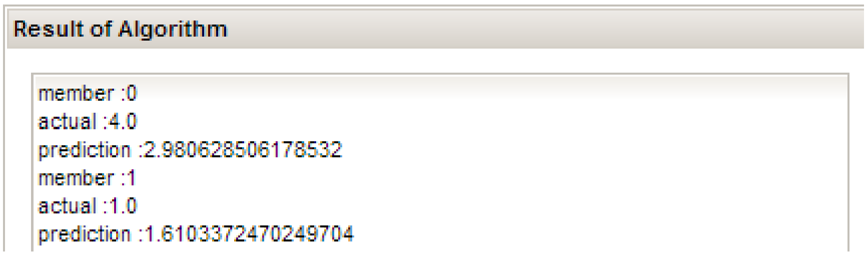


```
Apriori
=======

Minimum support: 0.9 (5 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 2

Generated sets of large itemsets:

Size of set of large itemsets L(1): 17

Size of set of large itemsets L(2): 28

Size of set of large itemsets L(3): 22

Size of set of large itemsets L(4): 8

Size of set of large itemsets L(5): 1

Best rules found:

1. BenMilkingsDevice=87=0 5 ==> BenNrOfAnimals=37=0 5    conf:(1)
2. BenNrOfAnimals=37=0 5 ==> BenMilkingsDevice=87=0 5    conf:(1)
3. BenRefusalsDevice=236=0 5 ==> BenNrOfAnimals=37=0 5    conf:(1)
4. BenNrOfAnimals=37=0 5 ==> BenRefusalsDevice=236=0 5    conf:(1)
5. BenFailuresDevice=1=0 5 ==> BenNrOfAnimals=37=0 5    conf:(1)
```

Figure 6.6 Results of the Apriori Algorithm on an Example Input File

An association rule in the output is an implication of the form A n ==> B m where A and B are conditions on the attributes of the dataset, n is the number of tuples A holds, m is the number of tuples B holds given A. The confidence value of a rule given as conf(x) in the output is the conditional probability P(B|A) which is m/n.

As an output of M5P algorithm, values of the last column attribute in the test set are predicted according to a model which is built using the data in the training set. After the algorithm is run on an example file, we get the following results:



```
member :0
actual :4.0
prediction :2.980628506178532
member :1
actual :1.0
prediction :1.6103372470249704
```

Figure 6.7 Results of the M5P Algorithm on an Example Input File

In the result of the M5P algorithm, “member” specifies the index of the predicted field. In the second row, "actual" gives the actual value of the specified field. Actual value does not necessarily have to exist in the test set. Finally "prediction" provides the predicted value of the specified field.

The user interface of the MILKMINER data mining module is shown in Appendix A. Note that, selected values for the algorithm specific parameters are embedded into the module for providing a plain and simple user interface. As a future work, an administrator module will be developed for setting and updating these parameters.

Chapter 7

Conclusions

In this thesis, data mining techniques are applied for analyzing MILKMINER project datasets.

The system that is the product of this project have a potential in terms of contributing to the development of data analysis techniques besides providing improvement of production, stability and health care in dairy farms.

Basic steps of this dairy farm data analysis and learning system project are:

- Selection of the data mining techniques which are appropriate for anayzing the datasets.
- Designing and developing a user interface for the data mining system that is appropriate for the users of MILKMINER.
- Designing and developing the techniques for the evaluation of the results of the analysis, incorporation of the results to the system and improving the results using learning techniques.

The design and implementation of the first two steps of the project is covered in this thesis and the last step is planned as a future work. In addition, a user interface for setting and updating algorithm specific parameters is under construction as another future work. Software product in the scope of this study is developed as a basic analysis tool which uses data mining techniques that can be continuously improved.

One of the important aspects of this study is the economical benefits it provides. Commercialization of the project outputs and their use in the dairy farms can increase the efficiency of milk production in the farms and in the livestock sector by providing "meaningful information".

In conclusion, the project results have a potential of contributing to the efficiency in milk production processes.

References

[1] W. Rossing, P. H. Hogewerf, “State of the Art of Automatic Milking Systems”, *Computers and Electronics in Agriculture*, Volume 17, Issue 1, 1-17, 1997.

[2] G. H. Klungel, B. A. Slaghuis, H. Hogeveen, "The effect of the introduction of automatic milking systems on milk quality", *J. Dairy* S*ci*., Volume 83, 1998-2003, 2000.

[3] O. Lind, A. H. Ipema, C. J. A. M. De Koning, T. T. Mottram, H. J. Herrmann, 2000, “Automatic milking”, *Bulletin of the IDF 348/2000*, 3-14, 2006.

[4] H. Schön, R. Artmann, H. Worstorff, “The Automation as a Key Issue in Future Oriented Dairy Farming”, *Proceedings of the International Symposium on Prospects for Automatic Milking*, EAAP publication 65, 7-22, 1992.

[5] C. J. A. M. De Koning, “Automatic Milking - Common Practice on Dairy Farms”, *Proceedings of the first North American conference on precision dairy management*, Toronto, Ontario, 52-67, 2010.

[6] C. J. A. M. De Koning, Y. V. De Vorst, A. Meijering, “Automatic Milking Experience and Development in Europe”, *Proceedings of the first North American Conference on Robotic Milking*, Toronto, Canada, 1-11, 2002.

[7] C. J. A. M. De Koning, J. Rodenburg, “Automatic Milking: State of the Art in Europe and North America”, *Automatic milking*, Wageningen Academic Publishers, Wageningen, the Netherlands, 27-40, 2004.

[8] W. Rossing, P. H. Hogewerf, “State of the Art of Automatic Milking Systems”, *Computers and Electronics in Agriculture*, Volume 17, Issue 1, 1-17, 1997.

[9] H. Hogeveen, W. Ouweltjes, “Sensors and Management Support in High-Technology Milking”, *Journal of Animal Science 2003*, Volume 81, 1-10, 2003.

[10] E. Mathijs, “Socio-economic Aspects of Automatic Milking”, *Proceedings of the international symposium Automatic Milking*, Wageningen Academic Publishers, Wageningen, The Netherlands, 46-55, 2004.

[11] W. Ouweltjes, C. J. A. M. De Koning, “Demands and Opportunities for Operational Management Support”, *Proceedings of the International Symposium Automatic Milking*, Wageningen Academic Publishers, Wageningen, The Netherlands, 433-442, 2004.

[12] R. Biji, S. R. Kooistra, H. Hogeveen, “The Profitability of Automatic Milking on Dutch Dairy Farms”, *Journal of Dairy Science,* Volume 90, 239-248, 2007.

[13] U**.** Fayyad, G. P. Shapiro, P. Smyth, "From Data Mining to Knowledge Discovery in Databases", *AI Magazine*, Volume 17, 37-54, 1996.

[14] M. Tsiknakis, “State-of-the-art report on standards”,
http://p-medicine.eu/, 93-94, 2011.

[15] C. Kamphuis, H. Mollenhorst, J. A. P. Heesterbeek, H. Hogeveen, “Data Mining to Detect Clinical Mastitis with Automatic Milking”, *Proceedings*

of the 5th IDF Mastitis Conference: Mastitis Research into Practice, Christchurch, New Zealand, 2010.

[16] S. J. Cunningham, G. Holmes, “Developing innovative applications in agriculture using data mining”, *Proceedings of the Southeast Asia Regional Computer Confederation Conference*, 1999.

[17] A. Mucherino, P. Papajorgji, P. M. Pardalos, “A Survey of Data Mining Techniques Applied to Agriculture”, *Operational Research*, Volume 9, Issue 2, 121-140, 2009.

[18] V. E. Bridges, “Utilization of Data Mining Software for the Identification of Emerging Animal Health Issues: Benefits and Limitations”, *Proceedings of the 9th International Symposium on Veterinary Epidemiology and Economics*, 2000.

[19] P. Revathi, M. Hemalatha, “Efficient Classification Mining Approach for Agriculture”, *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, Volume 1, Issue 2, 2011.

[20] Eclipse, Eclipse Site http://www.eclipse.org/downloads/moreinfo/jee.php

[21] Eclipse-Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Eclipse_(software)

[22] Hardware and Software Requirements for Installing SQL Server 2008 R2, Microsoft Developer Network, http://msdn.microsoft.com/en-us/library/ms143506(v=sql.105).aspx

[23] Weka Filters, Weka document archive, http://weka.sourceforge.net/doc.stable/weka/filters/Filter.html.

[24] Weka Classifiers, Weka document archive, http://weka.sourceforge.net/doc.dev/weka/classifiers/Classifier.html

[25] Weka Clusterers, Weka document archive, http://weka.sourceforge.net/doc.dev/weka/clusterers/Clusterer.html

[26] Weka Associations, Weka document archive, http://weka.sourceforge.net/doc.dev/weka/associations/package-summary.html

[27] Maven, Apache Maven Project Online, http://maven.apache.org/

[28] Rich faces-Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/RichFaces

[29] J. J. Garrett, "Ajax: A New Approach to Web Applications", *Adaptive Path*, 2005.

[30] L. I. U. Xing-tao, S. H. I. Bing, X. I. E. Ying-wen, "An improved Apriori algorithm for mining association rules", *Journal of Shandong University (Natural Science)*, 2008.

[31] D. P. Solomatine, M. B. L. A. Siek, “Flexible and Optimal M5 Model TRees with applications to flow predictions”, *Proceedings of the Sixth International Conference on Hydroinformatics*, World Scientific, 1-7, 2004.

[32] C. Zhan, A. Gan, M. Hadi, “Prediction of Lane Clearance Time of Freeway Incidents Using the M5P Tree Algorithm”, *Intelligent Transportation Systems, IEEE Transactions on,* Lehman Center for Transp. Res., Florida Int. Univ., Miami, FL, USA, Volume 12, Issue 4, 2011.

Appendix A Data Mining Module User Interface

The following screenshots provide examples of data mining module usage steps.

For Apriori Algorithm:

User selects the Apriori algorithm to be executed from combo box and clicks "Accept algorithm" button in order to apply the operation as shown in Figure A.1

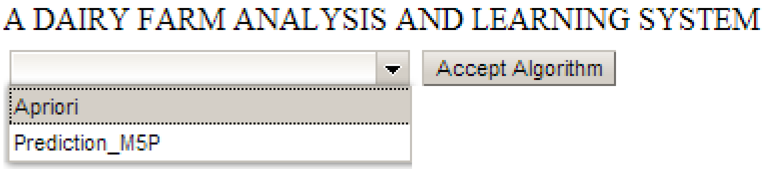


Figure A.1 Selecting Apriori Algorithm

User clicks to the "Add" button in order to select the dataset to be processed as shown in Figure A.2

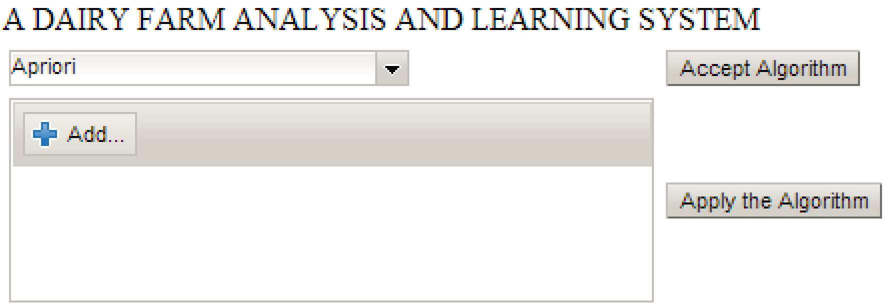
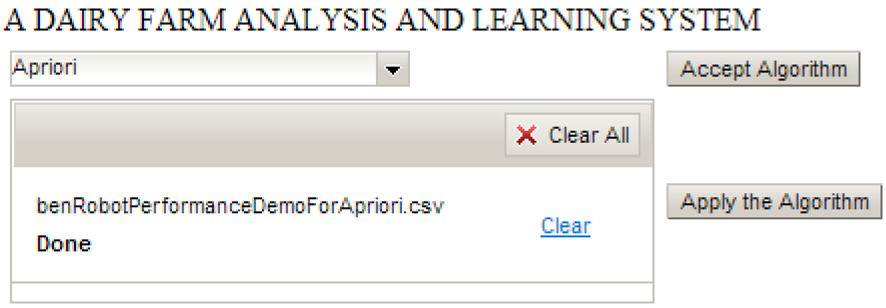


Figure A.2 Adding file for Apriori Algorithm

User clicks the "Upload" button in order to upload the selected file, after uploading file, information is demonstrated as shown in Figure A.3



A DAIRY FARM ANALYSIS AND LEARNING SYSTEM

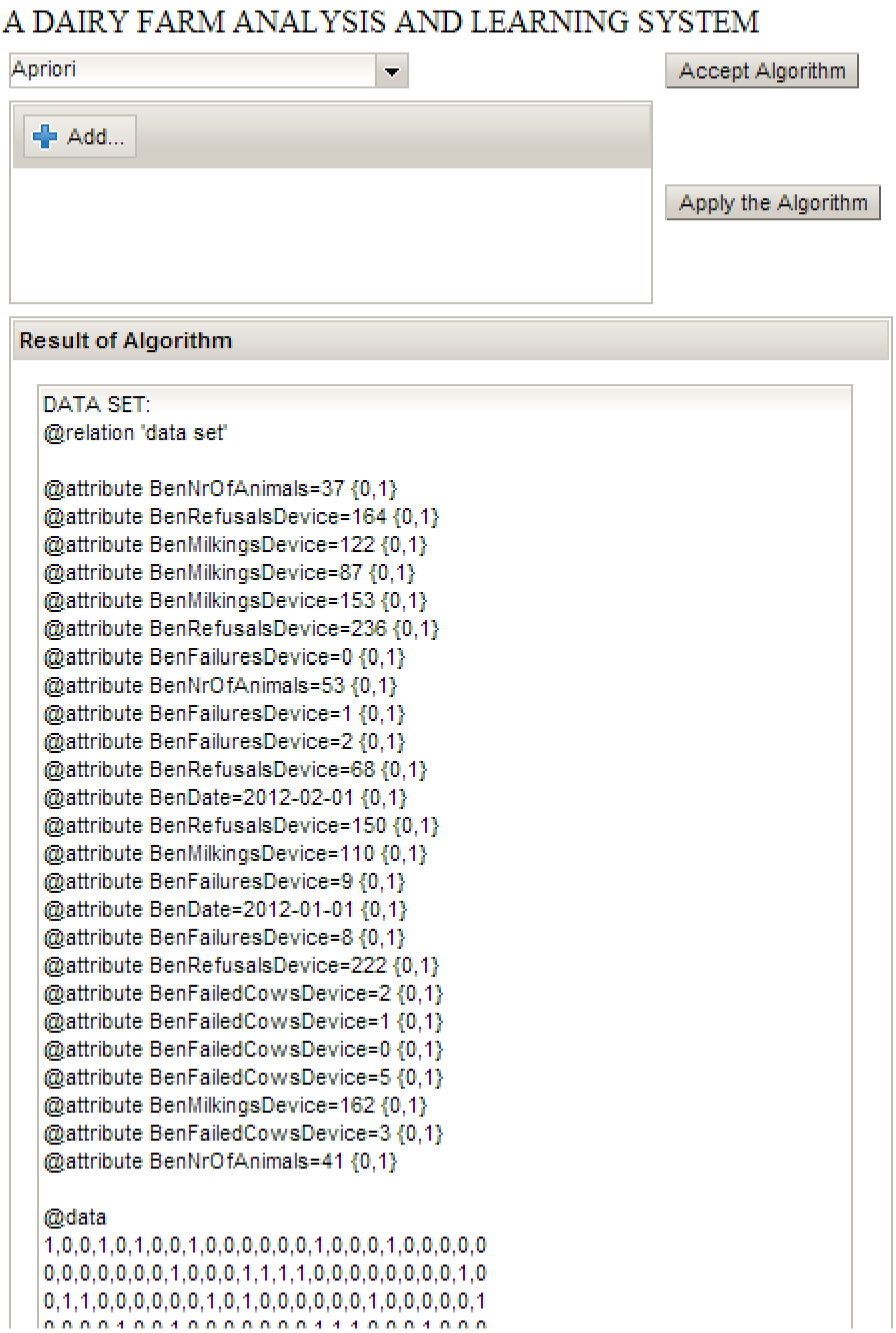
Apriori

benRobotPerformanceDemoForApriori.csv

Done

Figure A.3 After Uploading the File for Apriori Algorithm

User clicks the "Apply the Algorithm" button in order to apply the operation selected and output is printed on the screen as shown in Figure A.4



```
DATA SET:
@relation 'data set'

@attribute BenNrOfAnimals=37 {0,1}
@attribute BenRefusalsDevice=164 {0,1}
@attribute BenMilkingsDevice=122 {0,1}
@attribute BenMilkingsDevice=87 {0,1}
@attribute BenMilkingsDevice=153 {0,1}
@attribute BenRefusalsDevice=236 {0,1}
@attribute BenFailuresDevice=0 {0,1}
@attribute BenNrOfAnimals=53 {0,1}
@attribute BenFailuresDevice=1 {0,1}
@attribute BenFailuresDevice=2 {0,1}
@attribute BenRefusalsDevice=68 {0,1}
@attribute BenDate=2012-02-01 {0,1}
@attribute BenRefusalsDevice=150 {0,1}
@attribute BenMilkingsDevice=110 {0,1}
@attribute BenFailuresDevice=9 {0,1}
@attribute BenDate=2012-01-01 {0,1}
@attribute BenFailuresDevice=8 {0,1}
@attribute BenRefusalsDevice=222 {0,1}
@attribute BenFailedCowsDevice=2 {0,1}
@attribute BenFailedCowsDevice=1 {0,1}
@attribute BenFailedCowsDevice=0 {0,1}
@attribute BenFailedCowsDevice=5 {0,1}
@attribute BenMilkingsDevice=162 {0,1}
@attribute BenFailedCowsDevice=3 {0,1}
@attribute BenNrOfAnimals=41 {0,1}

@data
1,0,0,1,0,1,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0
0,0,0,0,0,0,0,1,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,0
0,1,1,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1
0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,1,1,1,0,0,0,1,0,0,0
```

Figure A.4 An Example Output of the Apriori Algorithm

For M5p Algorithm:

User selects the M5p algorithm to be executed from combo box and clicks "Accept algorithm" button in order to apply the operation as shown in Figure A.5

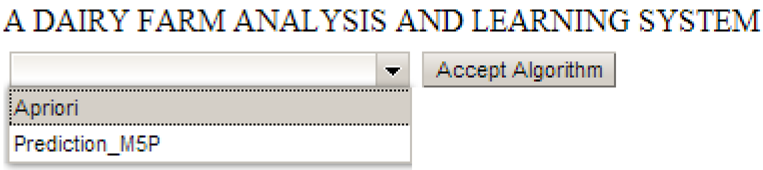


Figure A.5 Selecting M5P Algorithm

User clicks to the "Add" button firstly to select file including training set and secondly test set as shown in Figure A.6

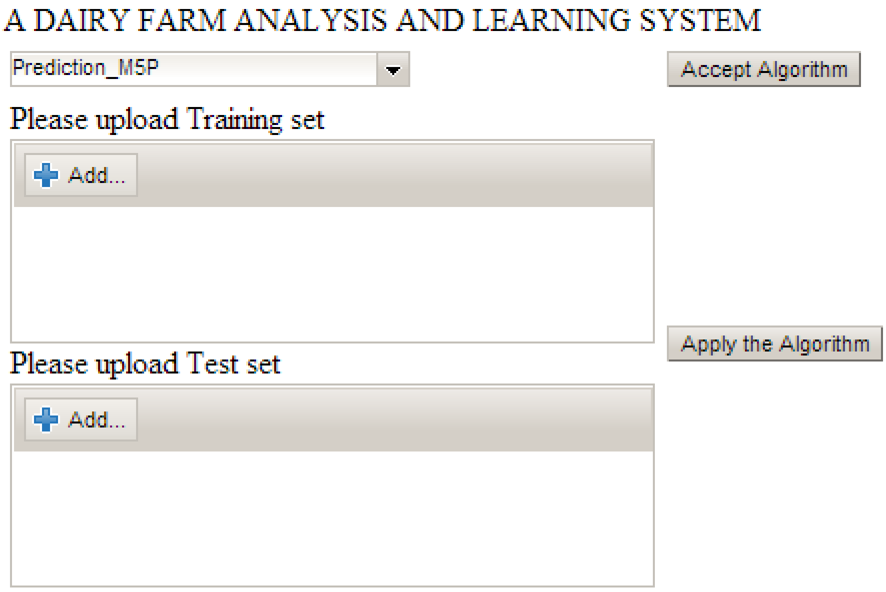
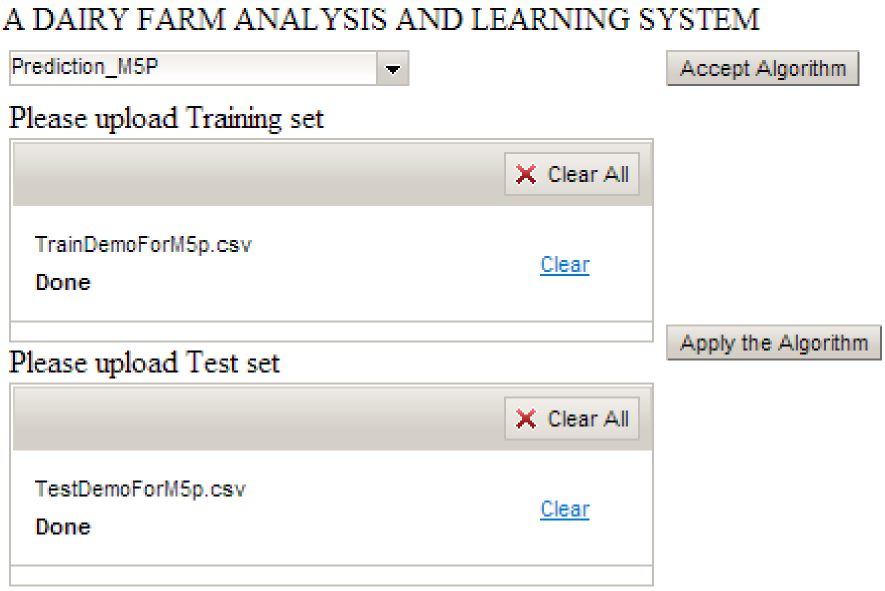


Figure A.6 Adding file for the M5P Algorithm

User clicks the "Upload" button for both training and test sets in order to upload the selected files, after uploading files, information is demonstrated as shown in Figure A.7



A DAIRY FARM ANALYSIS AND LEARNING SYSTEM

Please upload Training set

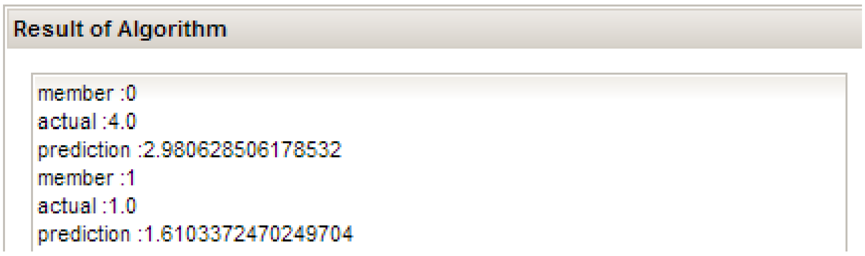
TrainDemoForM5p.csv
Done

Please upload Test set

TestDemoForM5p.csv
Done

Figure A.7 After Uploading files for M5P Algorithm

User clicks the "Apply the Algorithm" button in order to apply the operation selected and output is printed on the screen as shown in Figure A.8



Result of Algorithm

```
member :0
actual :4.0
prediction :2.980628506178532
member :1
actual :1.0
prediction :1.6103372470249704
```

Figure A.8 An Example Output of the M5P Algorithm

Curriculum Vitae

Ayşenur Genç was born on 23 May 1987, in İstanbul. She received his BS degree in Computer Engineering in 2010 from Gebze Institute of Technology. She worked as a Software Development Specialist at a private company from 2010 to 2012. She also has received BS degree in Philosophy. Her research interests include data mining and software development. Since 2012 she has been a Senior Software Developer at a private company.