A

Project Report

on

DATA CLASSIFICATION USING SCIP-MA ALGORITHM

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

North Maharashtra University, Jalgaon

Submitted by

Neha Pundlik Sonar
Ankit Kamlesh Dixit
Kajal Ekanath Shelke
Viren Anadshing Patil
Rahul Narayan Patil

Under the Guidance of

Dhanashree Tayade

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)

2016 - 2017

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the project entitled *Data Classification Using SCIP-MA Algorithm*, submitted by

Neha Pundlik Sonar
Ankit Kamlesh Dixit
Kajal Ekanath Shelke
Viren Anadshing Patil
Rahul Narayan Patil

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of North Maharashtra University, Jalgaon.

Date: May 9, 2017
Place: Jalgaon

Dhanashree Tayade
Guide

Prof. Dr. Girish K. Patnaik
Head

Prof. Dr. K. S. Wani
Principal

Acknowledgement

First of all we would like to extend our deep gratitude to almighty God, who has enlightened us with power of knowledge. We would like to express our sincere gratitude towards Prin cipal Prof. Dr. K. S. Wani for his encouragement during the work We wish to express our sincere and deep gratitude to Prof. Dr. Girish K. Patnaik (Head of Computer Department) for giving us such a great opportunity to develop this project. Inspiration and Guidance are invaluable in all aspects of life especially on the fields of gratitude and obligation and sympa thetic attitude which we received from our respected project guide, Mrs. Dhanashree Tayde whose guidance and encouragement contributed greatly to the completion of this project. We would like to thanks to all faculty members of Computer Engineering Department and all friends for their co-operation and supports in making this project successful. We would also like to thanks our parents for supporting us and helping us. We acknowledge our sincere gratitude to all who have directly or indirectly helped us in completing this project successfully.

Neha Pundlik Sonar
Ankit Kamlesh Dixit
Kajal Ekanath Shelke
Viren Anadshing Patil
Rahul Narayan Patil

Abbreviations

CBS Classify-By-Sequence

CSPs Classifiable Sequential Patterns

GSP Generalized Sequential Patterns

IP Interesting Patterns

SCIP Sequence Classification based on Interesting Patterns

SCIP-MA Sequence Classification based on Interesting Pattern Matching cohesion rule based classifier

Contents

List of Figures

List of Tables

Abstract

Data mining is the process of analyzing data from different perspective and summarizing it into useful information. Classification is one of the important task in data mining. Interestingness of pattern in a given class is measured by the two terms- support and cohesion. The set of classification rules are generated from these interesting pattern. In existing system these set of classification rules are used to classify the dataset. But the set of rules may contain independent rules or rules which are not correlate to each other. Such independent rules increases classification time. So, there is need to prune such independent rules, for which lift technique is used. The lift technique will reduce the amount of time require for classification of data.

Keywords: *Interesting Patterns, Classification Rules , LIFT Method, SCIP-MA Classification.*

Chapter 1

Introduction

Data mining is the process of identifying the valid information from huge dataset. There are various tasks in data mining such as classification, clustering, prediction, time series analysis, pattern mining, etc. Data mining is helpful in different domains such as market analysis, decision support, fraud detection, business management. Pattern Mining is a pop ular technique which consists of finding subsequences or item set appearing frequently in a set of sequence.

In Section 1.1 background is described. Motivation is described in Section 1.2. In Section 1.3 problem definition is described. Scope is described in Section 1.4. In Section 1.5 objective is described. Organization of report is described in Section 1.6. In the last section summary is presented.

1.1 Background

The classification task can be defined as assigning class labels to new sequences based on the knowledge gained in the training stage. There exist a number of studies integrating pattern mining techniques and classification, such as classification based on association rules (CBA), sequential pattern based sequence classifier , the Classify-By-Sequence(CBS) algorithm, and so on. These combined methods can produce good results as well as provide users with information useful for understanding the characteristics of the dataset. In practice, most datasets used in the sequence classification task can be divided into two main cases. In the first case, the class of a sequence is determined by certain items that co-occur within it, though not always in the same order. In this case, a classifier based on sequential patterns will not work well, as the correct rules will not be discovered, and, with a low enough threshold, the rules that are discovered will be far too specific. In the other case, the class of a sequence is determined by items that occur in the sequence almost always in exactly the same order. At first glance, a sequence based classifier should outperform an itemset based classifier in this situation. However, itemset based classifiers will do better when the pattern

sometimes occurs in an order different from the norm. This robustness means that itemset based classifiers can handle cases where small deviations in the subsequences that determine the class of the sequences occur. Moreover, due to a simpler candidate generation process, itemset based methods are much faster than those based on sequential patterns.

The above observations motivate the proposed research, Sequence Classification based on Interesting Patterns (SCIP). First of all, present algorithms to mine both types of interesting patterns itemsets and subsequences. As a second step, convert the discovered patterns into classification rules, and propose two methods to build classifiers to determine the class to which a new instance belongs. In the first method, select the rules to be applied based on their confidence, while the second uses a novel approach by taking into account how cohesive the occurrence of the pattern that defines the rule is in the new instance. Finally, step away from pattern based classification and evaluate the quality of our pattern miner by using our patterns as features in a variety of feature based classifiers.

When looking for interesting patterns in sequences, a pattern is typically evaluated based on how often patterns occurs (support). However, the proximity of the items that make up the pattern to each other (cohesion) is important, too . If two classes are determined by exactly the same items, traditional pattern based classifiers may struggle. For example, if class A is determined by the occurrence of h with a shortest interval of 2 and class B by the occurrence of h with a shortest interval of 5, pattern h will not be enough to be able to tell the difference, and this will be solved by considering the cohesion information. Therefore, use both cohesion and support to define interesting patterns in a sequence dataset. Finally, utilise these interesting patterns to build classifiers.

Basic Concepts

In data mining, pattern mining is one of the most important aspects.Although,the patterns mined contain redundant items or subsequences.To eliminate the redundant part various techniques such as frequent pattern mining is used.The drawback of frequent pattern mining is that it introduces monotonicity in the patterns.Thus,lay emphasis on the use of interesting pattern mining.Interesing pattern mining mines the interesting patterns of data pertaining to a particular application.Interesting pattern mining makes use of the Interestingness measure to rate the pattern as to how interesting it is.For finding the interesting measure,the methods support and cohesion are used.Support measures in how many sequences the pattern appears, while cohesion measures how close the items making up the pattern are to each other on average, using the lengths of the shortest intervals containing the pattern in

different sequences.Support and Cohesion are elaborated further. Support:-Support is an indication of how frequently the item-set appears in the database.

1. Let X be an item-set, X $\rightarrow$ Y an association rule and T a set of transactions of a given database.

2. The support value of X with respect to T is defined as the proportion of transactions in the database which contains the item-set X.

Cohesion refers to the degree to which the elements of a item set belong together. Cohesion measures how close the items making up the pattern are to each other on average, using the lengths of the shortest intervals containing the pattern in different sequences.

Interestingness can be measured using the aforementioned methods,support and cohesion. The product of Support and Cohesion will provide us with Interestingness measure over a class of sequential patterns.The items that make up the dataset are converted into Rules after measuring the interestingness. These rules further help in classifying data.The problem is that there are redundant rules,which occupy a lot of space and take up CPU time for processing.To avoid such a scenario,pruning of rules is to be done.This pruning of rules is done is using "Lift".

The lift of a rule is defined as, the ratio of the observed support to that expected if X and Y were independent. If some rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.If the lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.

1.2 Motivation

The pattern mining makes use of interesting patterns. There are numerous ways to find out interesting patterns,but the use of support along with cohesion provides good results with respect to the time factor. The rules are drawn based on these interesting patterns. Confidence is used to prune rules. The problem with Confidence is that it does not account for the dependency of antecedent and consequent of a rule rather,it accounts for the truth value of a rule. So, a rule pertaining to a particular application might be true for another it might not be. In such cases,Lift comes into play. Lift prunes only those rules that are

independent of each,thus those rules that are not true for a particular instance but can be true for another are not pruned which is the case when confidence is used.

1.3 Problem Definition

Classification system classifies dataset into various classes. User inputs dataset which is to be classified. The classification involves the measure of interesting pattern in a class of sequence. This involves use of techniques such as support and cohesion. The interesting patterns are then used to build a sequence classifier. The pattern can be both, itemset and subsequences. The support measures in how many sequence the pattern appears. Cohesion measures how close the items are to each on an average. The classifier is built using the classification rules. There may be additional rules which occupy a lot of space. These rules also take time for processing. There is a need to prune such rules. This pruning is done by using a technique known as Lift. Lift is the ratio of observed support to the expected, if x and y were independent. A rule that has independent events are pointless to process. Such rules are eliminated using Lift. The pruned subset classifies the interesting patterns of the dataset into a sequential pattern.

1.4 Scope

Classification uses many techniques such as frequent pattern mining is used. The drawback of frequent pattern mining is that it introduces monotonicity in the patterns. Thus, lay emphasis on the use of interesting pattern mining. There are redundant rules that are drawn for the sole purpose of classification of data. But these redundant rules do not help in any way to classify data. Hence, processing such rules takes up time and space. To prune such rules, Lift is used. Later on, a rule based classifier is built which classifies the data.

1.5 Objective

Project objectives are the specific objectives for which the project works to achieve them within a stipulated time. They should directly address the problem mentioned in the Problem Definition.

1. To calculate support,cohesion methods value.
2. To find Interesting pattern.
3. To prune the redundant classification rules by Lift method.
4. Classification of given data set.

1.6 Organization of Report

This section provides information about the organisation of the report distributed in different chapters, along with the sections discussed in each chapter.

Chapter 1 is introduction. It discusses about the background information, problem definition, scope and objectives of the project.

Chapter 2 is the system analysis. It discusses about the literature survey of code clone detection, proposed system, different types of feasibility study, risk analysis, scheduling and effort allocation.

Chapter 3 is system requirement specification. It discusses about the different types of requirements, such as functional, non-functional, hardware, software, performance and design requirements.

Chapter 4 is system design. It focuses on the architecture of system, data flow and the UML diagrams.

Chapter 5 is implementation. It illustrates the details of implementation, environmental details and flow of development of the system.

Chapter 6 is system testing. It discusses different levels of testing, presents test cases and their results.

Chapter 7 is result and analysis. It discusses about the result and analysis of result.

Chapter 8 is conclusion and future scope. It discusses about the conclusion and future scope of project.

1.7 Summary

In this chapter, introduction of system is described. In the next chapter, system analysis is presented.

Chapter 2

System Analysis

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. System analysis chapter will shows overall system analysis of the concept, description of the system, meaning of the system. System analysis is the study of sets of interacting entities, including computer systems analysis. The development of a computer based information system includes a systems analysis phase which produces or enhances the data model which itself is to creating or enhancing a database. There are a number of different approaches to system analysis. When a computer based information system is developed, systems analysis would constitute the development of a feasibility study, involving determining whether a project is economically, socially, technologically and organizationally feasible. The main purpose behind the development of proposed system is to overcome the drawbacks of existing system.

In the Section 2.1 literature survey is described. Existing techniques of classification is described in Section 2.2. In Section 2.3 feasibility study of proposed system is described. Risk analysis is described in Section 2.4. In Section 2.5 project scheduling is described. Effort allocation table is described in Section 2.6. In the last section summary is presented.

2.1 Literature Survey

Since the concept of sequential pattern mining was first described by Agrawal and Srikant [1], other sequential pattern mining methods have been developed, such as Generalized Sequential Patterns (GSP) [7], SPADE [10], PrefixSpan [4], and SPAM [3]. PrefixSpan typically shows better performance than GSP and SPADE, but, when dealing with dense databases, the performance of PrefixSpan may be worse than that of SPADE. A number of sequence based classifiers have been based on these methods.

Lesh et al. in [5] has combined sequential pattern mining and a traditional Naive Bayes classification method to classify sequences. They introduced the Feature Mine algorithm

which leveraged existing sequence mining techniques to efficiently select features from a sequence dataset. The experimental results showed that BayesFM (combination of Naive Bayes and FeatureMine) is better than Naive Bayes only. Although a pruning method is used in their algorithm, there was still a large number of sequential patterns used as classification features. As a result, the algorithm could not effectively select discriminative features from a large feature space.

Tseng and Lee in [9] proposed the Classify-By-Sequence(CBS) algorithm for classifying large sequence datasets. The main methodology of the CBS method is mining Classifiable Sequential Patterns (CSPs) from the sequences and then assigning a score to the new data object for each class by using a scoring function. They proposed a number of alternative scoring functions and tested their performance. The results showed that the length of a CSP is the best attribute for classification scoring.

Exarchos et al. in [8] has proposed a two-stage methodology for sequence classification based on sequential pattern mining and optimisation. In the first stage, sequential pattern mining is used, and a sequence classification model is built based on the extracted sequential patterns. Then, weights are applied to both sequential patterns and classes. In the second stage, the weights are tuned with an optimisation technique to achieve optimal classification accuracy. However, the optimisation is very time consuming, and the accuracy of the algorithm is similar to that of Feature Mine.

Cheng Zhou et al. in [2] has proposed in 2016 approach mine the frequent and confident(or discriminative) patterns for building a classifier. the cohesion of a pattern to identify interesting patterns and confident method use for classification rules.Using confident method accuracy of result is increase. But method take a more time for classification and also consume more storage space,hence in proposed system use Lift method instead confident method for better performance.

2.2 Proposed System

The proposed system classifies data using the interesting patterns. Interesting pattern is find by using two methods support and cohesion.

The patterns considered could be both itemsets or subsequences (sequential patterns), which is why provide formal definitions applying to both pattern types below. The support count of a pattern is defined as the number of different sequences in which the pattern occurs; regardless of how many times the pattern occurs in any single sequence. In other

words, when looking for the support count of a pattern alone, user can stop looking at a sequence as soon as user have encountered the first occurrence of the pattern. To determine the interestingness of a pattern, however, it is not enough to know how many times the items making up the pattern occur. also like to know how close they appear to each other. To do this, define interesting patterns in terms of both support and cohesion. Our goal is to first mine interesting patterns in each class of sequences, and then use them to apply a sequence classifier, i.e., a function from the set of sequences S to the set of class labels L.

Classification rules are drawn from interesting patterns. By using Confidence method, find a rule to be true for a particular instances. But the same rule wont be true for another instances. Hence, 'Lift method' is used in proposed system. 'Lift' prunes only those rules in which the antecedent and consequent are independent of each other. The rules which possess dependency are stored in the databases. These rules can be used for classification depending on different instances.

2.3 Feasibility Study

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. The entire feasibility of the project is comprehended by economical feasibility, operational feasibility and technical feasibility.

Feasibility has applied to code clone detection using hybrid approach pertains to the following areas:

1. Economical Feasibility
2. Operational Feasibility
3. Technical Feasibility

2.3.1 Economic Feasibility

Economical feasibility refers to whether the project can be developed at an affordable price. Talking from any organization's point of view, the economical feasibility refers that the organization should be able to finance the project. Moreover, the returns from the project has also to be considered.

To decide whether a project is economically feasible, various factors are considered as:

1. Cost benefit analysis

2. Maintenance costs

The proposed system is computer based and it does not require any additional hardware components, hence there is no cost of hardware involved. Considering, the software part required for the project, the project is to be developed in python and python is open source therefore there is no need to required cost for software. It requires average computing capabilities, which are very basic requirements hence it doesn't include additional economic overheads, which renders the system economically feasible.

2.3.2 Operational Feasibility

Operational feasibility of the project describes the ease with which even the naive user can operate the developed system. The developed system should be as easy and user friendly to operate and should be self-comprehensive.To determine the operational feasibility of the system, the awareness level of the users should take into consideration. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system.Also the system is very friendly to use.

The Proposed system provide classification using minimum time. Performance of the proposed system is enhanced by lift method. To determine the operational feasibility of the system, the awareness level of the users should take into consideration. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly to use.

2.3.3 Technical Feasibility

Technical feasibility of a project, is performing a check whether the development of project is possible with the available technological resources. The technical feasibility is a very important aspect to be considered before the official commencement of the project by the organization. The technical feasibility is checked by pondering over the functional requirements of the user.

To determine whether the proposed system is technically feasible, the technical issues involved behind the system should taken into consideration. Proposed system uses python technology. Python is an open source technology, it is available for free of cost and conveniently. As far as platform for the project is concerned, it is decided to perform the project on the window OS. Therefore, the project has to be done on any Windows OS and also on Ubuntu . Thus, it becomes quite sure the project is technically feasible.

2.4 Risk Analysis

Project risk analysis is the identification and quantification of the likelihood and impact of events that may damage the project. Risk analysis is an opportunity to help solve problems and to enhance communications within the project for a more effective team effort. Risk analysis has applied to code clone detection using hybrid approach pertains to the following areas:

1. Technical Risks
2. Business Risks
3. Project Risks

2.4.1 Technical Risks

Technical risk is the risk that some feature of the correct system can not be implemented due to a technical reason. Technical risk is exposure to loss arising from activities such as design and engineering, manufacturing, technological processes and test procedures.

To determine whether the proposed system has technical risk or not, the technical issues involved behind the system should taken into consideration. proposed system uses python technology. As python is an open source technology, it dose not have any technical risk. Thus, no technical risk associated with project.

2.4.2 Business Risks

The term business risk refers to the possibility of inadequate profit or even loss due to uncertainties e.g., changes in tastes, preferences of consumers, strikes, increased competition, change in government policy, obsolesce etc. Business risks implies uncertainty in profits or danger of loss and the events that could pose a risk due to some unforeseen events in future, which causes business to fail.

There are numerous techniques proposed to classification in software system. The proposed system use a lift method for identify classification rules which is able to required less time for classification. Therefore there is no business risk associated with project.

2.4.3 Project Risks

Project risk is defined as an uncertain event or condition that, if it occurs, has a positive or negative effect on a projects objectives. Threaten the project plan, that is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project

risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirement problems and their impact on a software project.

As per as the project is concerned, the task in project is completed under estimated schedule and project is completed before deadline. Therefore there is no project risk associated with project.

2.5 Project Scheduling

This section specifies the project scheduling of the project.Software project scheduling is an activity that distributes estimated effort across the planed project duration by allocating the effort to specific software engineering task. A project schedule is a listing of project's milestones, activities,and deliverables, usually with the intended start and finish dates. Generally, the schedule of the project is developed using the well known gantt chart. gantt chart is a very popular, convenient tool which helps in developing the schedule not only for the projects but also for many other purposes in which scheduling is required.

For developing gantt chart, the activities of the project, the commencement date and the duration for each activity, are given as input to the popular applications for developing gantt chart and in return these applications provide the gantt chart.

Data Classification Using SCIP_MA Algorithm.planner - Planner

WBS	Name	Start	Finish	Work	Duration	Slack	Cost	Assigned	% Complete
1	Project Docu	Aug 9	Aug 17	7d	7d	157d	0		0
2	**Requriment**	**Aug 9**	**Aug 31**	**21d**	**17d**	**147d**	**0**		5
2.1	Rereferenc	Aug 9	Aug 18	8d	8d	156d	0		0
2.2	Drawback I	Aug 15	Aug 23	7d	7d	153d	0		0
2.3	Algorithm	Aug 24	Aug 31	6d	6d	147d	0		0
3	Planning	Sep 1	Sep 9	7d	7d	140d	0		0
4	**Design**	**Sep 12**	**Oct 7**	**20d**	**20d**	**120d**	**0**		35
4.1	UML	Sep 12	Sep 23	10d	10d	130d	0		0
4.2	DFD	Sep 20	Sep 26	5d	5d	129d	0		0
4.3	System Arc	Oct 3	Oct 7	5d	5d	120d	0		0
5	**Implementa**	**Oct 10**	**Jan 12**	**76d**	**69d**	**51d**	**0**		50
5.1	Module1	Oct 10	Nov 18	30d	30d	90d	0		0
5.2	Module2	Nov 15	Dec 19	25d	25d	69d	0		0
5.3	Module3	Dec 15	Jan 12	21d	21d	51d	0		0
6	Testing	Jan 31	Mar 3	24d	24d	15d	0		0
7	**Documenta**	**Feb 15**	**Mar 24**	**40d**	**28d**		**0**		85
7.1	Report	Feb 15	Mar 21	25d	25d	3d	0		0
7.2	Paper	Mar 6	Mar 24	15d	15d		0		100

Figure 2.1: Project Scheduling

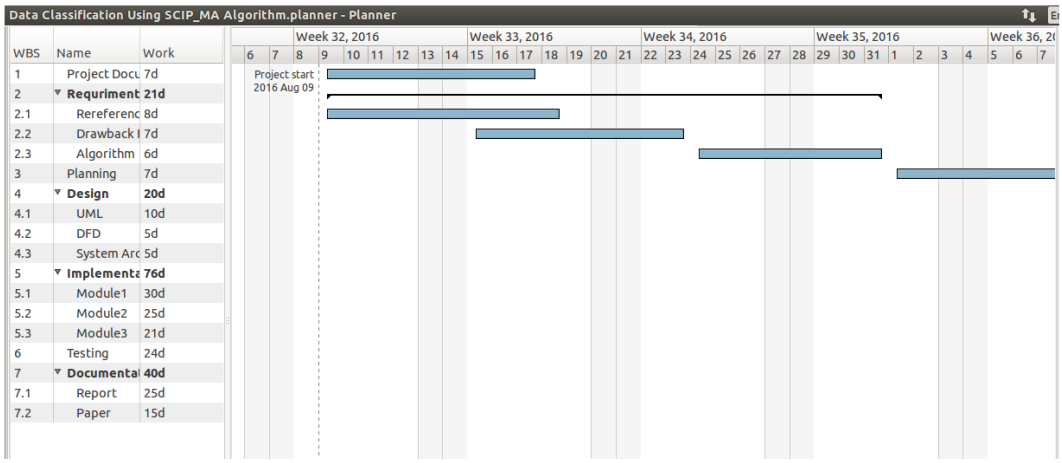

Figure 2.2: Gantt Chart

Figure 2.2 illustrates the gantt chart for the project. The gantt chart provides the comprehensive understanding of the carried out activities with time required for each activity. Also the chart shows the total time given to project as well as the sequence of activities in which the project work is carried out.

2.6 Effort Allocation

A recommended distribution of effort across the definition and development phases is often referred to as the 40-20-40 rule. Forty percent of all effort is allocated to front-end analysis and design. A similar percentage is applied to back-end testing. It can be correctly inferred that coding (20% of effort) is de-emphasized. The characteristics of each project must dictate the distribution of effort. Work expended on project planning rarely accounts for more than 23 percent of effort, unless the plan commits an organization to large expenditures with high risk. Requirements analysis can comprise of 10-25% of project effort. Effort expended on analysis or prototyping increases in direct proportion with project size and complexity. A range of 20 to 25 percent of effort is normally applied to software design. Because of the effort applied to software design, code should follow with relatively little difficulty. A range of 15- 20 percent of overall effort can be achieved. Testing and subsequent debugging can account for 30-40% of software development effort.

Table 2.1: Effort Allocation

	Neha Sonar	Ankit Dixit	Kajal Shelke	Viren Patil	Rahul Patil
Project Planning	20%	20%	20%	20%	20%
Requirement Gathering	20%	25%	20%	20%	15%
Design	25%	20%	20%	15%	20%
Coding	25%	35%	15%	15%	10%
Testing	20%	20%	20%	20%	20%
Documentation	30%	15%	15%	20%	20%

Table 2.1 illustrates the allocation of efforts by each group member for various phases of project such as planning, analysis and design

2.7 Summary

In this chapter, system analysis is described. In the next chapter, system requirement is presented.

Chapter 3

System Requirement Specification

Software Requirements Specifications is the official statement of what is required of the system developers. It includes both user requirements and a detailed specification of the system requirements. Requirement analysis is done in order to understand the problem the software system is to solve.

In Section 3.1, Hardware requirements of the system are described. Software Requirements are described in Section 3.2. In Section 3.3, Functional Requirements of system are described. Non functional Requirements are described in Section 3.4. In Section 3.5 Performance Requirements are described. Finally summary is presented in last section.

3.1 Hardware Requirements

Hardware requirements give the physical component required for the proposed system. The hardware requirement includes a system with following configurations:

1. Processor : Pentium IV or above
2. Display Type : VGA and higher.
3. RAM : 256MB
4. Storage Memory : 1GB

3.2 Software Requirements

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional description, a representation of system behavior, an indication of performance requirements and design

constraints, appropriate validation criteria, and other information pertinent to requirements. The various software requirements of the system are summarized here:

1. Operating system : Windows 7/8, Ubuntu.
2. System Type : 64-bit/32-bit operating system.
3. Front end : Python
4. Python Version : 3.5.2

3.3 Functional Requirement

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. The functional requirements of the proposed system are:

1. The system should be able to Provide classification of given dataset.
2. The system should be able to Find interesting pattern.
3. The system should be able to identify classification rules.
4. The system should be able to Prune the unrelated classification rules.

3.4 Non Functional Requirement

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non functional requirements is detailed in the system architecture. The non functional requirements are:

1. The system should be able to take a input dataset.
2. The user should be able to abort the operation in the middle processing.
3. Warnings and error messages should be provided to user throughout the system.
4. Fast response time.

5. Easy enhancement.

6. Execution qualities, such as usability.

7. User interface - The system is designed in such a way that instructions are given clearly to navigate through the System.

3.5 Performance Requirements

The performance requirements of the proposed system are:

1. The system should take the minimum possible time for the operation to perform.

2. The system should be robust.

3.6 Summary

In this chapter, system requirement is described. In the next chapter, system design is presented.

Chapter 4

System Design

System Design chapter provides graphical structure of the project by using various UML diagrams. System design provides the understanding and procedural details necessary for implementing the system recommended in the system study. Design is a meaningful engineering representation of something that is to be built. It can be traced to a customers requirements and at the same time assessed for quality against a set of predefined criteria for good design. In the software engineering context, design focuses on four major areas of concern are data, architecture, interfaces and components [6].

In Section 4.1 architecture of proposed system is described. Data flow diagrams are described in Section 4.2. In Section 4.3 all UML diagrams such as use case diagram, sequence diagram, class diagram component diagram, state chart diagram, activity diagram, deployment diagram of the project are described. Finally summary is presented in last section.

4.1 System Architecture

The fig.4.1 shows the working of Data Classification System in systematic manner. System Architecture contains four main procedures such as Find Interesting Patterns, Find classification Rules, Prune additional classification rules, Apply Classifier for data classification. As shown in fig.4.1 support and cohesion methods are use for interesting patterns and then IPs are used for identify classification rules. After getting all classification rules lift method is used for pruning unrelated classification rules. In last dataset classification is done by using Sequence Classification based on Interesting Pattern Matching cohesion rule based (SCIP-MA) classifier.

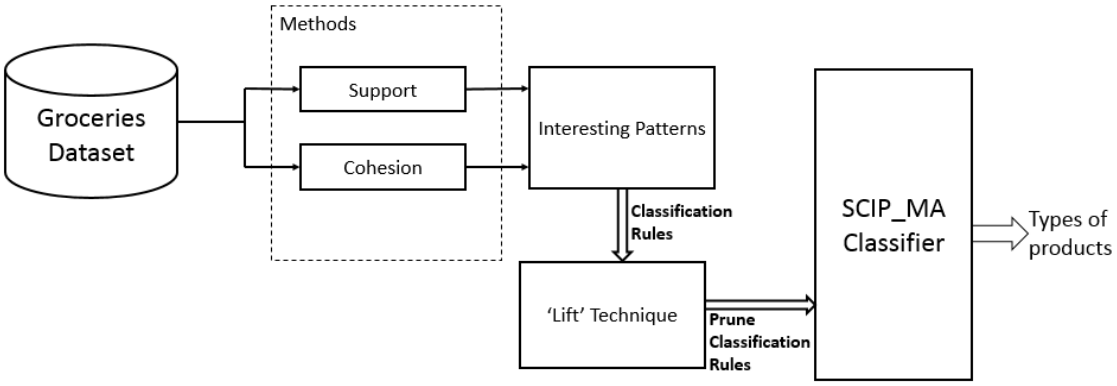

Figure 4.1: System Architecture

4.2 Data Flow Diagram

As information moves through software, it is modified by a series of transformations.Data Flow Diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble chart. The data flow diagram may be used to represent a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling [6].

The data flow diagram serves two purposes:

- To provide an indication of how data are transform as the moves through the system.
- To depict the function that transforms the data flow.

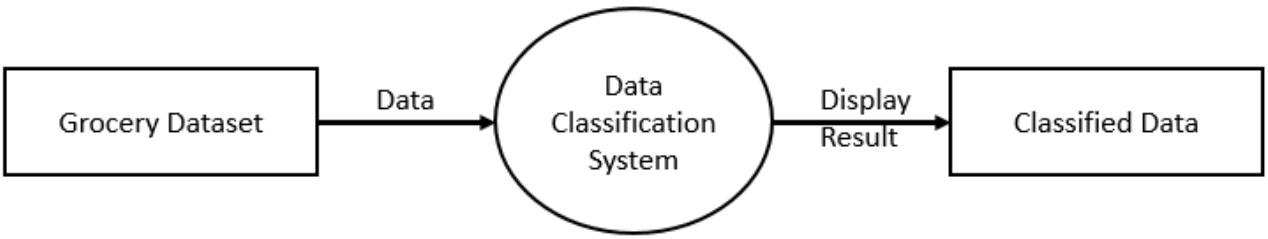

Figure 4.2: Level 0 DFD

A level 0 DFD, also called a fundamental system model or a context model. The DFD level 0 shows the abstract of the whole system.

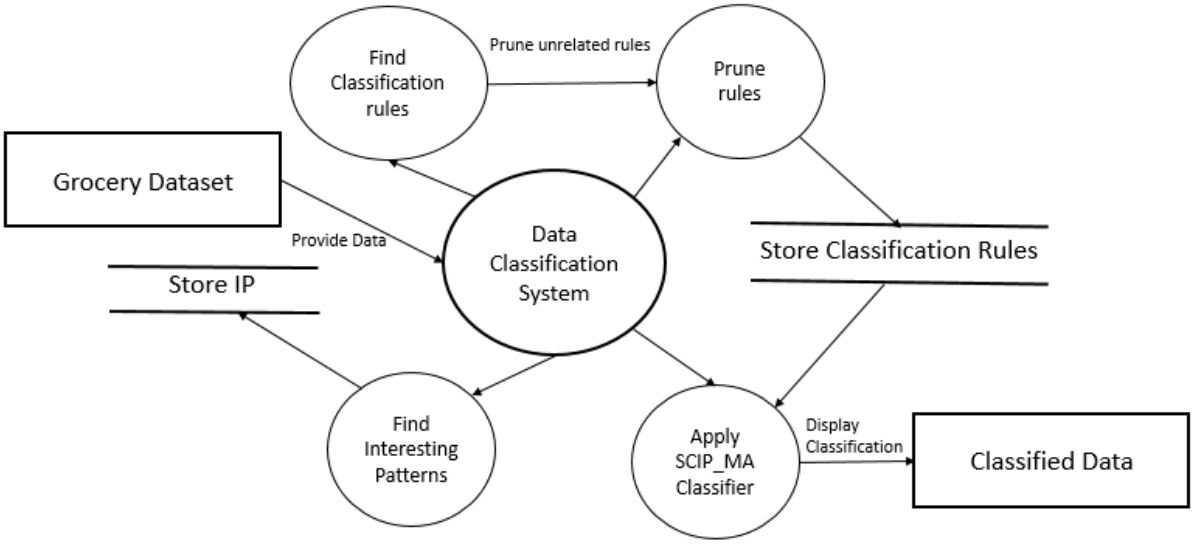

Figure 4.3: Level 1 DFD

The Figure shows the level 1 DFD. The DFD level 1 shows some internal structure of the system also identifies data stores that are used by the major processes.

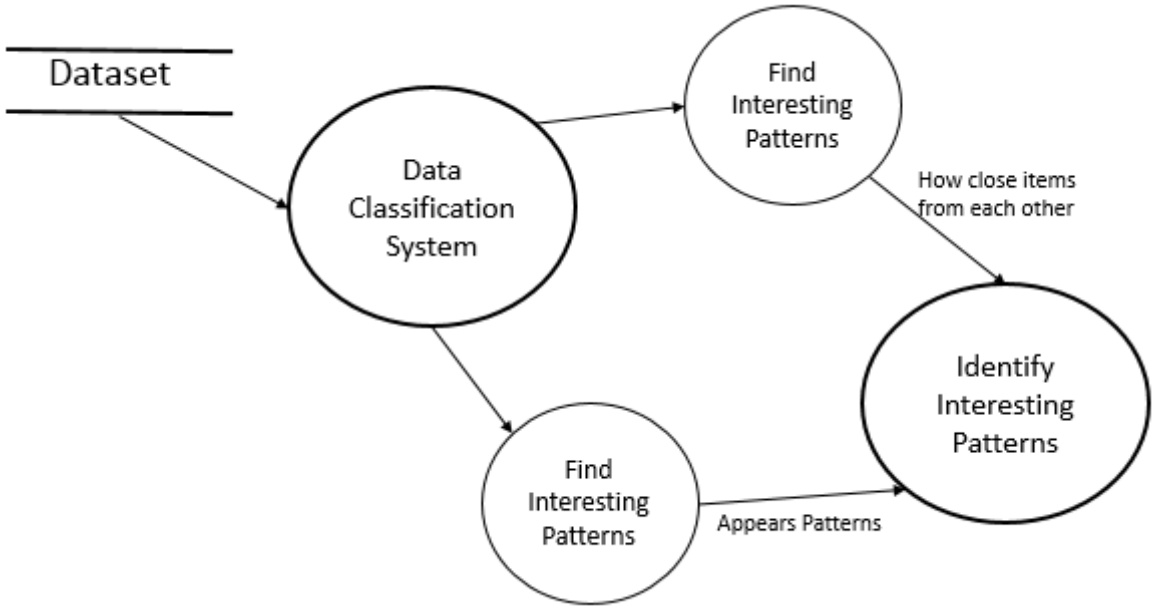

Figure 4.4: Level 2 DFD

Level 2 DFDs aim to give an detailing of the full system. They look at the system in more detail. Major processes are broken down into sub-processes.

4.3 UML Diagrams

This section illustrates the various UML diagrams of the project.The Unified Modeling Language (UML)is a standard visual modeling language intended to be used for modeling business and similar processes, analysis, design, and implementation of software-based systems.UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems. UML is a standard modeling language, not a software development process.

4.3.1 Use Case Diagram

A Use case diagram shows a set of use cases and actors and their relationships. Usecase diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. The Use Case diagram of the proposed system shows the basic functionality of system such as Provide dataset,measure Interesting PatternsIP,find classification rules,prune the rules,apply the classifier and display classification to user.

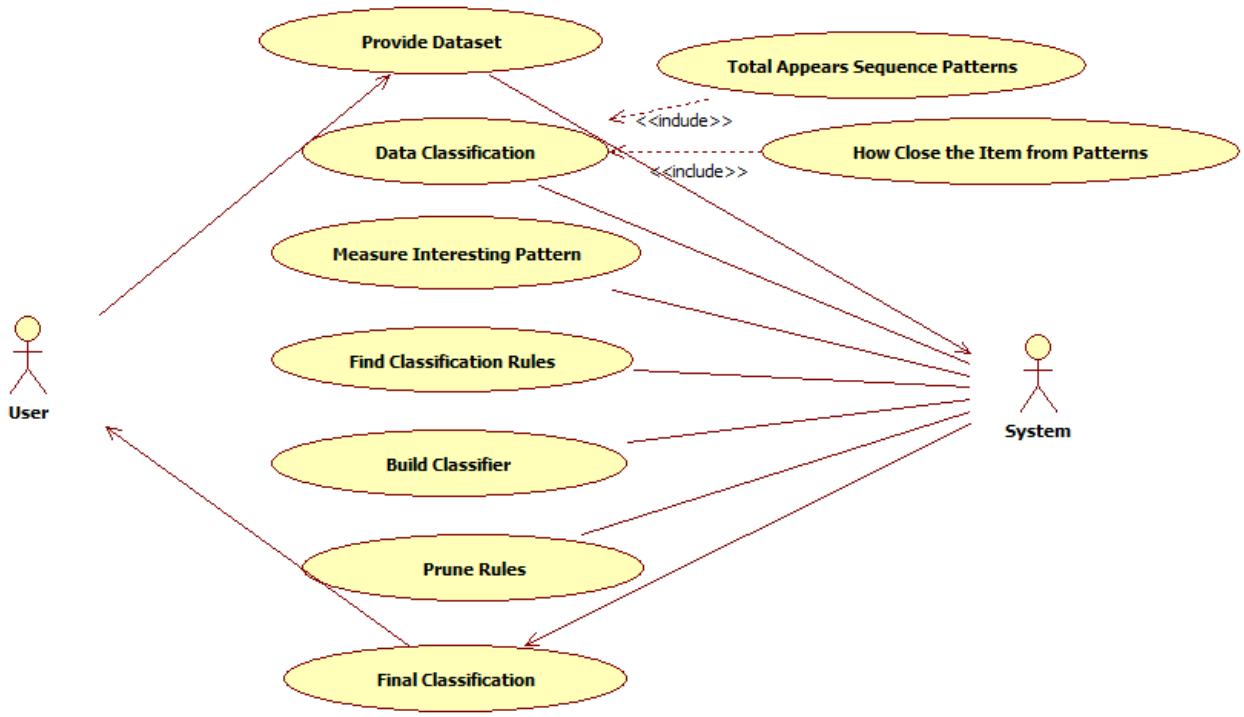

Figure 4.5: UseCase Diagram

4.3.2 Class Diagram

A Class diagram shows a set of classes, interfaces and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagram address the static design view of a system. Figure 4.6 shows the class diagram of system. Class diagram of the proposed system include various classes with different relationship. Classifier is depend on 'classifier rule' class.

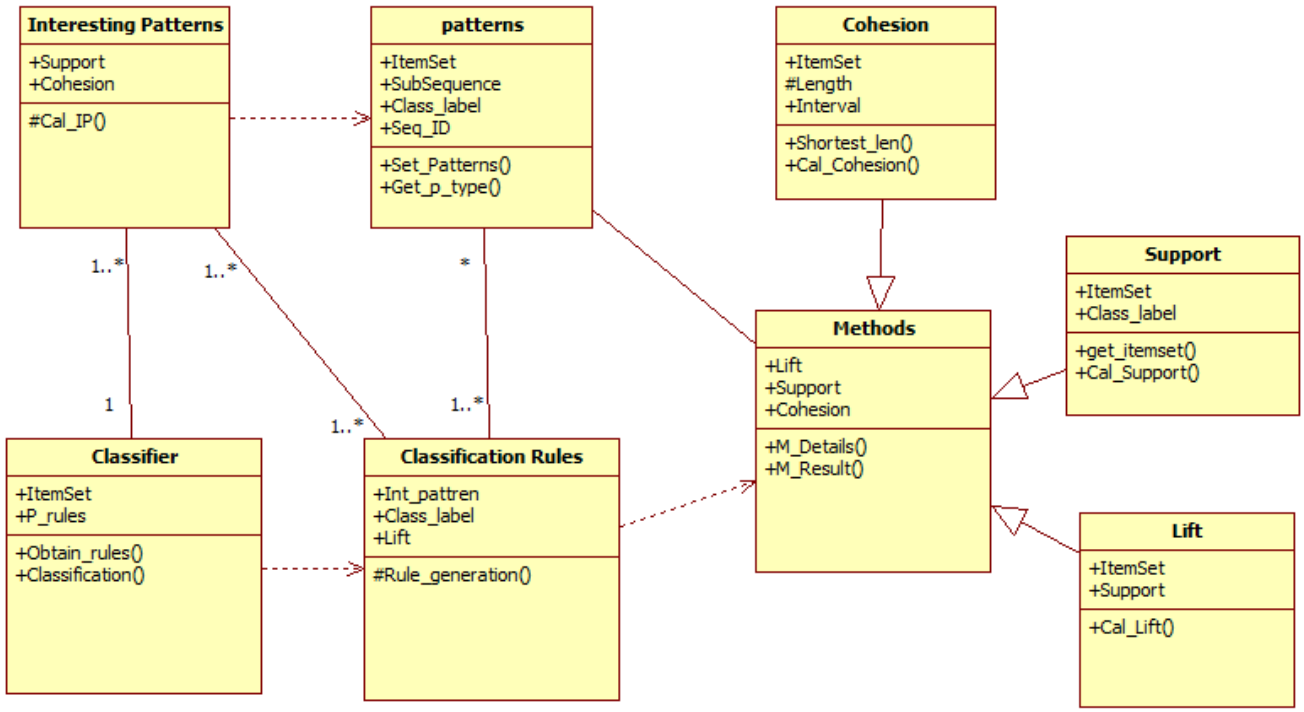

Figure 4.6: Class Diagram

4.3.3 Sequence Diagram

Both sequence diagrams and collaboration diagrams are kinds of interaction diagrams. An shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. Interaction diagrams address the dynamic view of a system. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other. Sequence diagram shows the Control flow of any system which content object with timeline. figure 4.7, 4.8, 4.9, 4.10 show sequence diagrams of system. All sequence diagrams shows the flow of the various usecases.

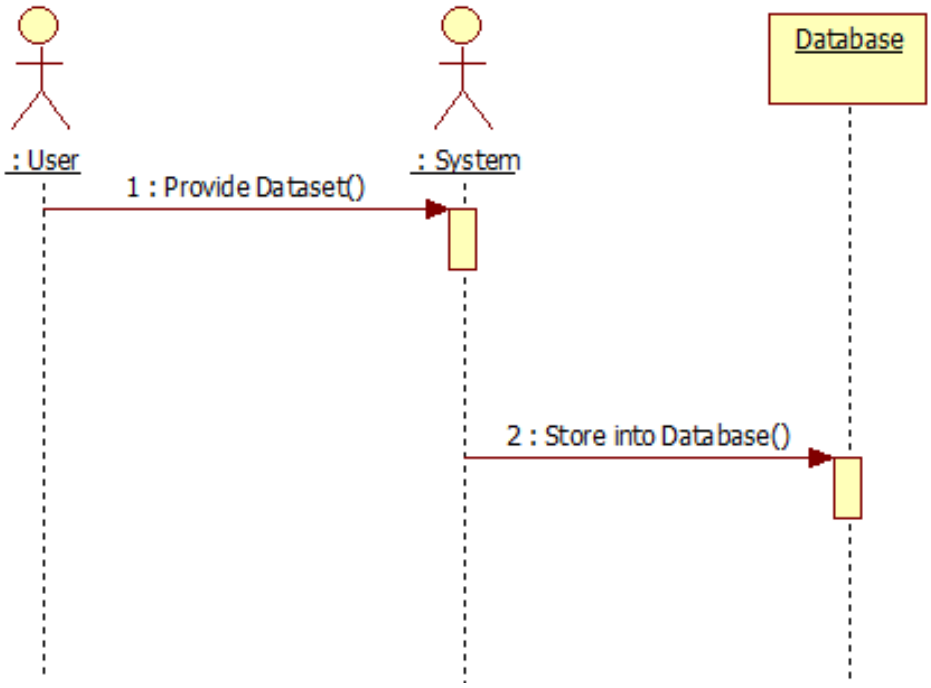

Figure 4.7: Sequence Diagram for UseCase 'Provide Dataset'

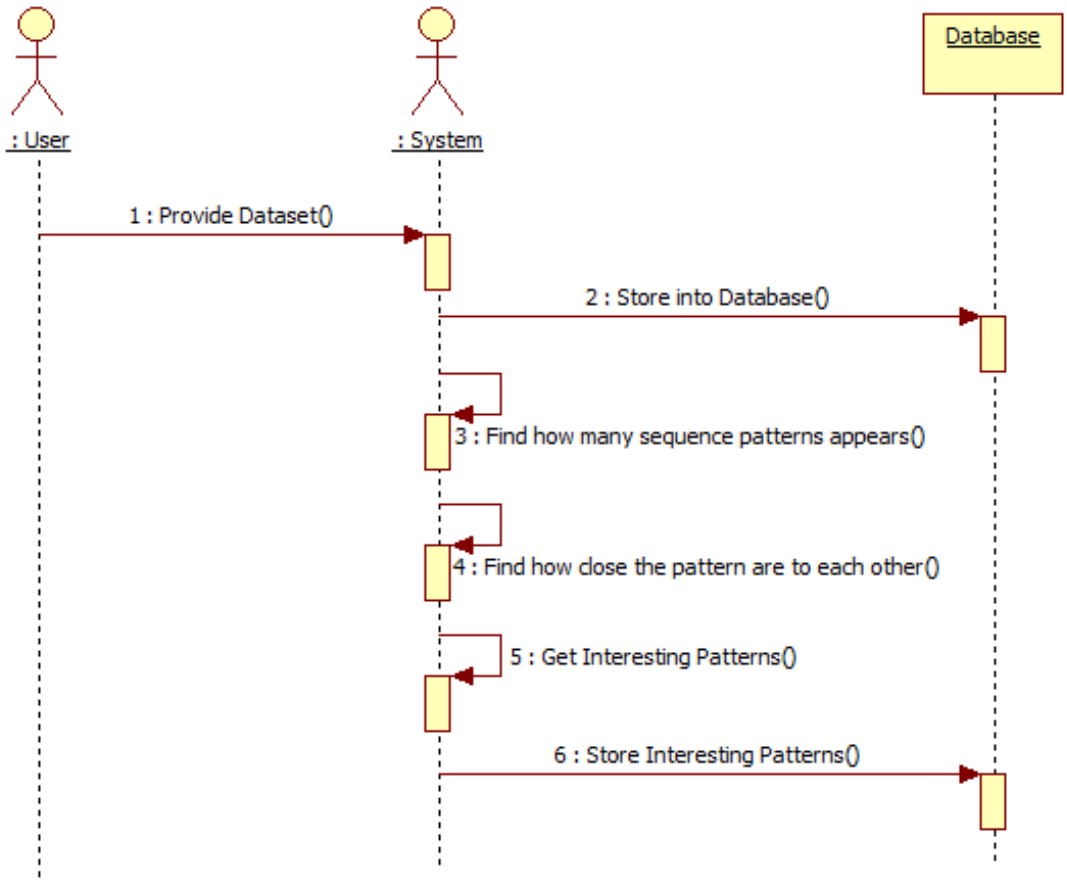

Figure 4.8: Sequence Diagram for UseCase 'Measure Interesting Patterns'

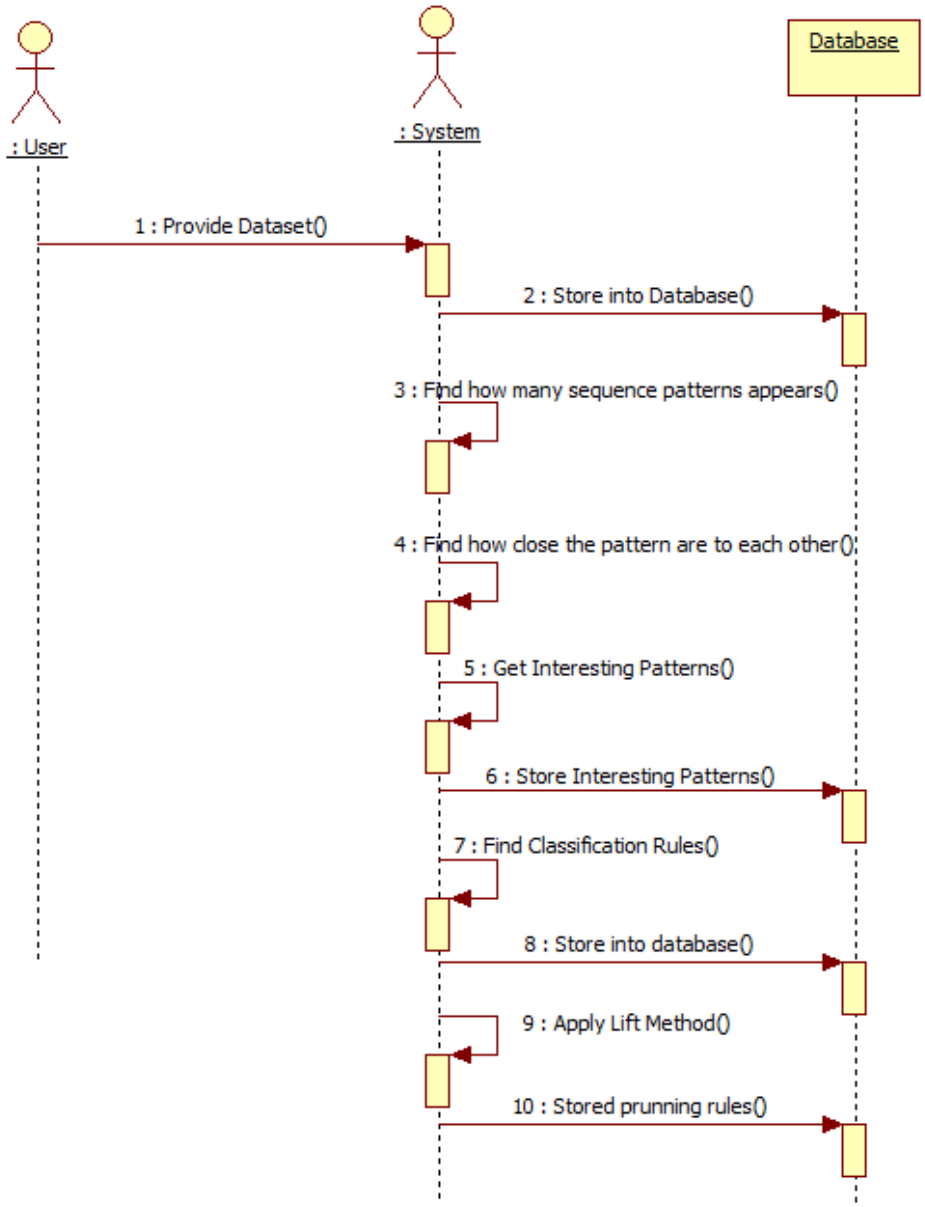

Figure 4.9: Sequence Diagram for UseCase 'Prune Rules'

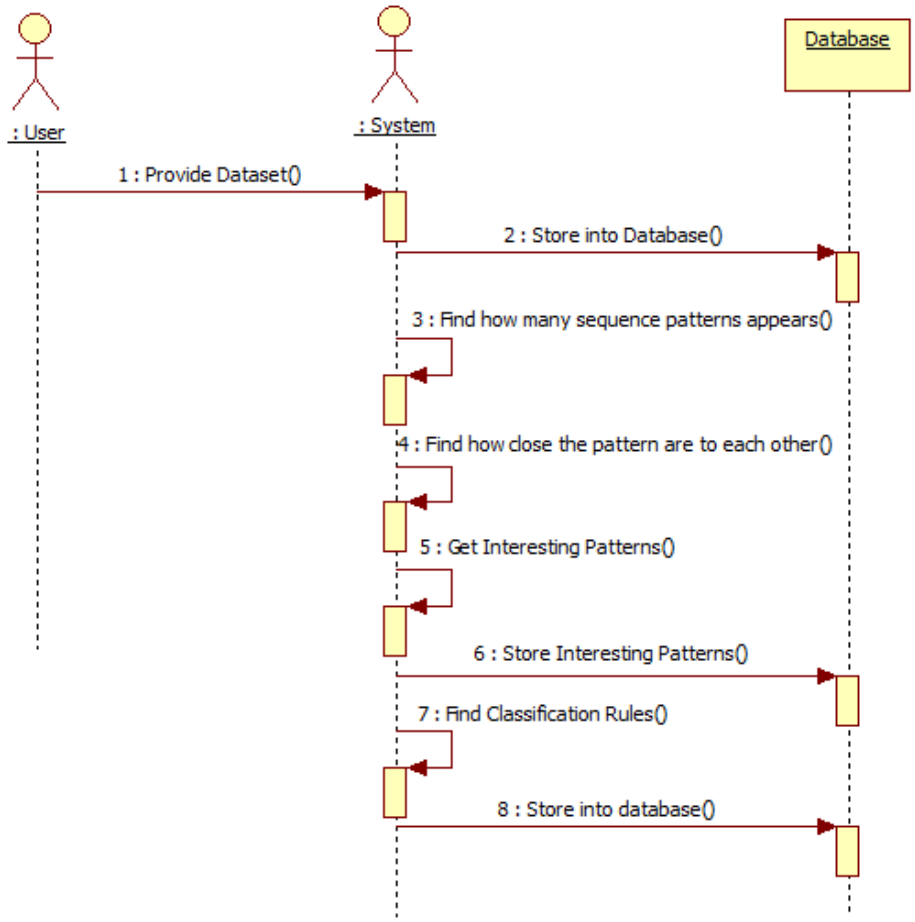

Figure 4.10: Sequence Diagram for UseCase 'Find Classification Rules'

4.3.4 Component Diagram

A component diagram shows the organizations and dependencies among a set of components. Component diagrams address the static implementation view of a system. They are related to class diagrams in that a component typically maps to one or more classes, interfaces, or collaborations. Component Diagram Contains Package Specification,Task specification,Object and dependency relationship amongst the object.Figure 4.12 shows the component diagram of system. proposed system Component diagram shows dependency of modules which are use in system such as system is depend on classifier and dataset again classifier depend upon classification rules and lift method.

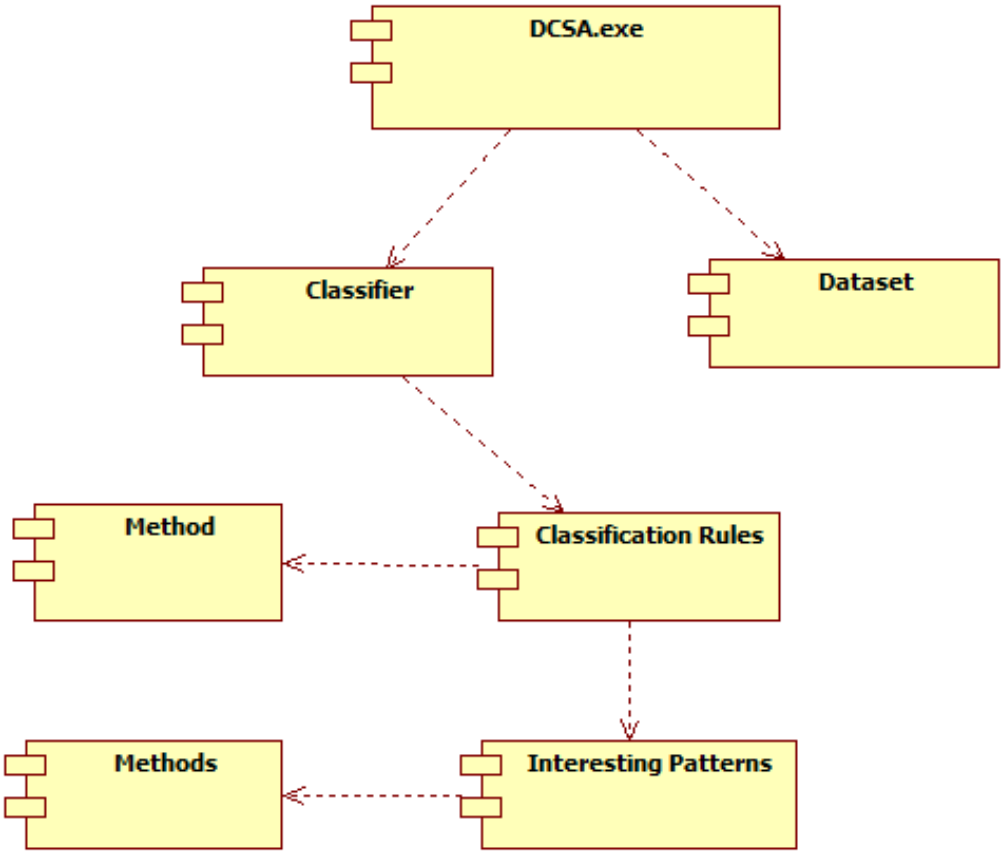

Figure 4.11: Component Diagram

4.3.5 Deployement Diagram

A deployment diagram shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of an architecture. They are related to component diagrams in that a node typically encloses one or more components. figure 4.13 shows the deployment diagram of proposed system. Deployment diagram include the physical component of the system such as user and printer.

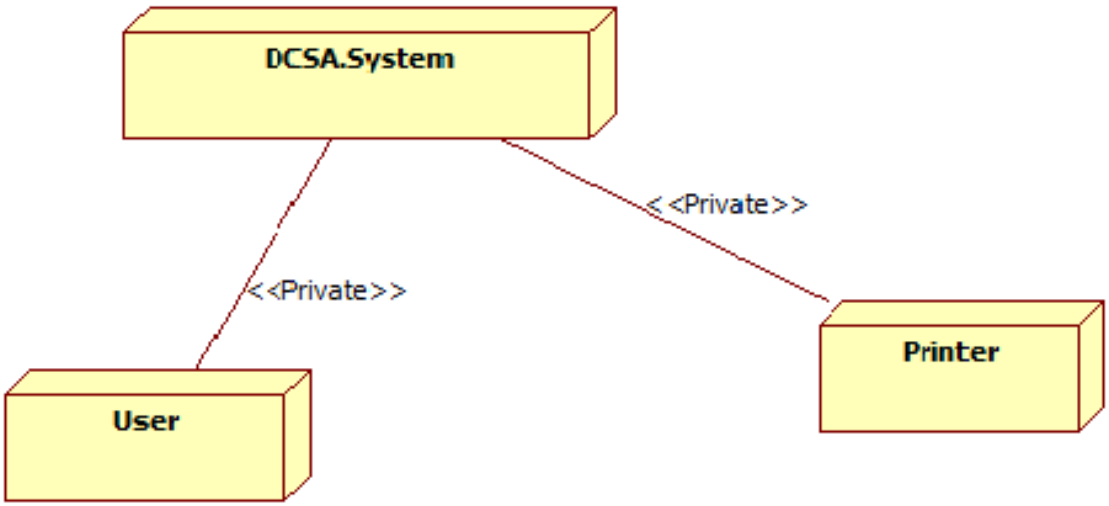

Figure 4.12: Deployment Diagram

4.3.6 Statechart Diagram

Statechart diagram is use to show behaviour of system in case of some external events. In statechart diagram, a state is a condition during the life of an object or an interaction during which it satisfies some condition, performs some action, or waits for some event. An initial is a kind of pseudo state that represents the starting point in a region of a state machine. It has a single outgoing transition to the default state of the enclosing region, and has no incoming transitions. There can be one (and only one) initial state in any given region of a state machine. It is not itself a state but acts as a marker. A final state represents the last or "final" state of the enclosing composite state. There may be more than one final state at any level signifying that the composite state can end in different ways or conditions. When a final state is reached and there are no other enclosing states it means that the entire state machine has completed its transitions and no more transitions can occur. Figure 4.13 shows the state chart diagram of proposed system. Statechart diagram include various state such as check pattern types,measure interesting pattern,apply classifier etc.

4.3.7 Activity Diagram

Activity diagram shows the basic activities between two immediate states of statechart diagram. Activity diagram shows the flow from activity to activity. An activity is an ongoing non atomic execution within a state machine. Activity ultimately results in some action which is made up of executable atomic computation that result in a change in state of system or the return of value. The activity diagram is a collection of vertices and arcs. It also contains FORKING and JOINING as shown in figure. Activity diagram commonly contains activity states and action states, transitions and objects. Figure 4.11 shows the Activity diagram of system.Activity Diagram provide detail between two state such as Sequential patterns and measure interesting patterns.

4.4 Summary

In this chapter, system design is described. In the next chapter, system implementation is presented.

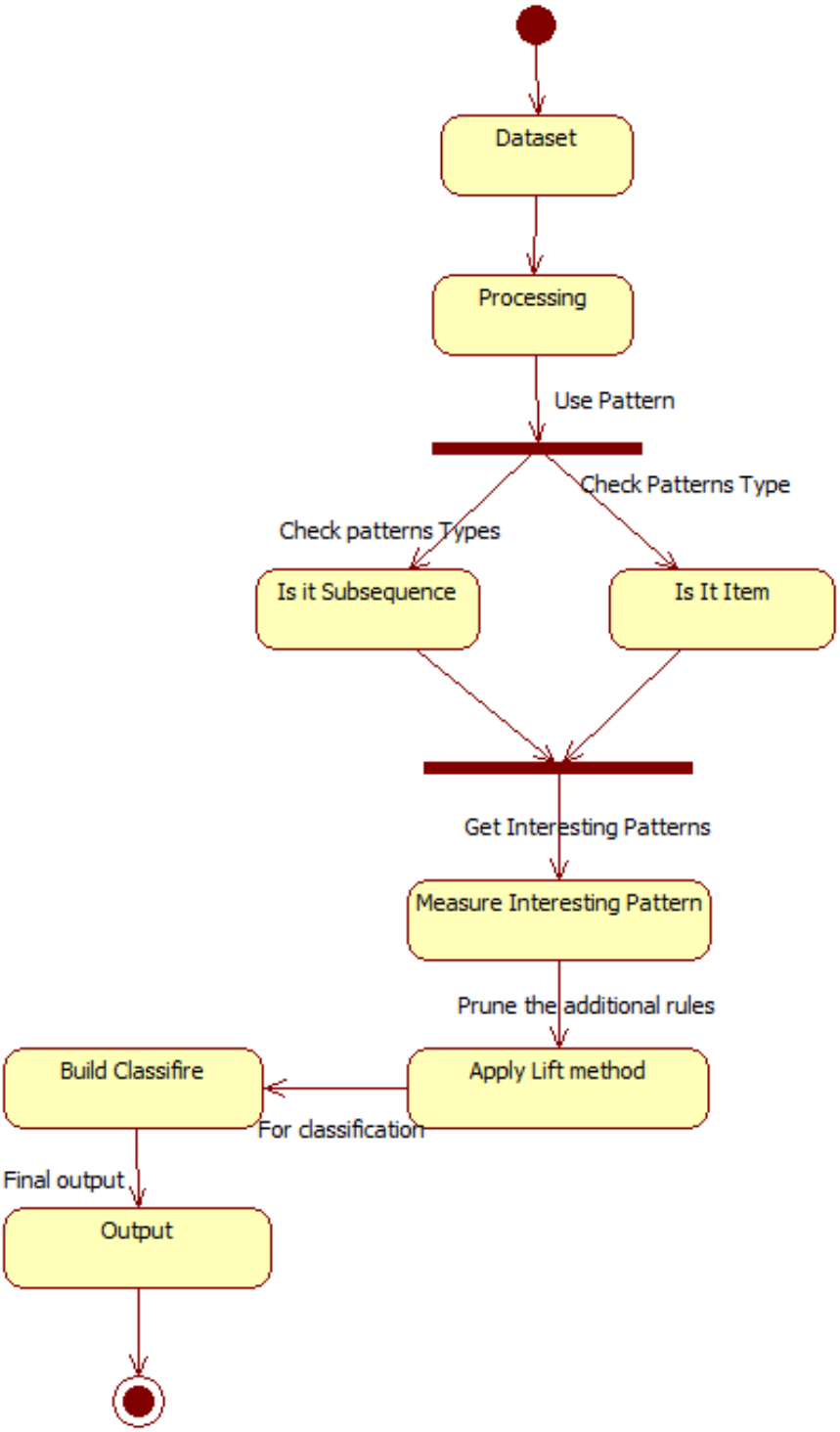

Figure 4.13: StateChart Diagram

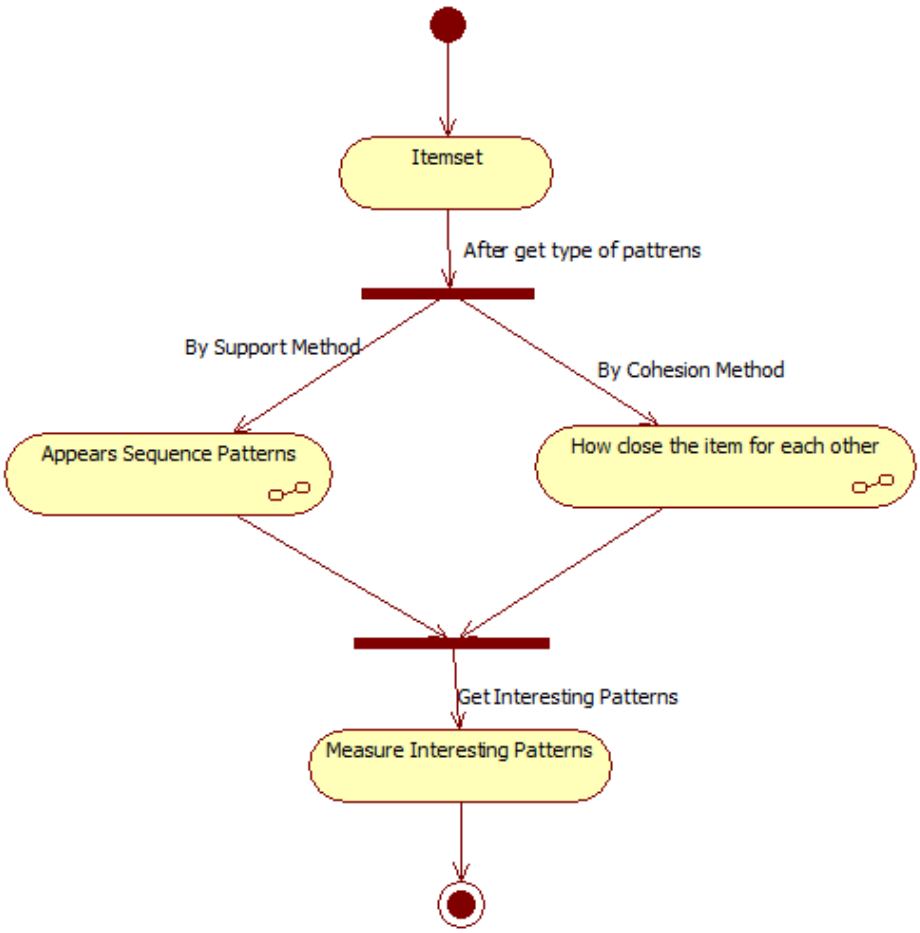

Figure 4.14: Activity Diagram

Chapter 5

Implementation

Project implementation is the phase where visions and plans become reality. This is the logical conclusion, after evaluating, deciding, visioning, planning, applying for funds and finding the financial resources of a project.

In section 5.1 implementation detail is described. Implementation environment is described in Section 5.2. In Section 5.3 flow of system development is described. Finally summary is presented in the last section.

5.1 Implementation details

In the Proposed system it is important to understand which methods are more appropriate than others so as to implement a faster market basket classification system. Interesting pattern find using support and cohesion methods. Lift technique is used to prune the classification rules.

5.1.1 Interesting Pattern

The pattern interestingness is depends on two methods: support and cohesion. The support method count of a pattern is defined as the number of sequences in which the pattern occurs. Cohesion method find how close pattern appear to each other.

1. **Support**

   Support is an indication of how frequently the item-set appears in the dataset. Suppose item set α, Denote the set of sequence that contain all item if α as T (α). Denote the set of sequence that contain all item of α labelled by class label C_K. The support of a pattern P in a given class of sequences Q_K can now be defined as,

$$T_K(P) = \frac{|T_K(P)|}{|Q_K|}, where Pisitemset. \tag{5.1}$$

2. **Cohesion**

Cohesion refers to the degree to which the elements of an item set belong together. Cohesion measures how close the items making up the pattern are to each other on average, using the lengths of the shortest intervals containing the pattern in different sequences. An item set α in a sequence $\beta \in \mathrm{T}(\alpha)$ as $\mathrm{W}(\alpha,\beta) = \min \{t_2 - t_1 + 1$, where $t_1 \quad \mathrm{t} \quad t_2\}$. In order to compute the cohesion of a pattern P within class k, we now compute the average length of such shortest intervals in,

$$T_K(P) = W_K(P) = \frac{\sum_{\beta \in T_K(P)} W(P,\beta)}{|T_K(P)|}, Pisitemset \tag{5.2}$$

Here, $\overline{W_K}(\mathrm{P})$ is greater than or equal to the number of item in P, denote as P. Furthermore, for a fully cohesive pattern, $\overline{W_K}(\mathrm{P})$=P. Therefore, define cohesion of P in $T_K(\mathrm{P})$ as,

$$\overline{A}(P) = \frac{|P|}{\overline{W_K}(P)} \tag{5.3}$$

All Pattern containing just one item are fully cohesion, that is,

$$\overline{A_K}(P) = 1If|P| = 1 \tag{5.4}$$

The cohesion of P in a single sequence γ is defined as

$$\overline{A}(P,\gamma) = \frac{|P|}{W(P,\gamma)}. \tag{5.5}$$

3. **Interestingness**

In a class of sequence Q_K, now a define the interestingness of a pattern P as,

$$I_K(P) = \Pi_K(P) \times \overline{A_K}(P), WherePis\alpha. \tag{5.6}$$

Minimum support threshold min_sup and a Minimum interestingness threshold min_int, a pattern P is considered interesting in a set of sequences labelled by class label C_K, if

$$\Pi_P \geq min_sup, \Pi_P \geq min_int \tag{5.7}$$

5.1.2 Lift Method

There are various technique used for finding classification rules but the problem is that there are redundant rules, which occupy a lot of space and take up CPU time for processing. To avoid such a scenario, pruning of rules is to be done. This pruning of rules is done is using

"Lift" method. Lift interestingness measure defines the number of transaction that contain the item used to find interesting patterns. The Lift is denoted by Lift(X = Y) as follows:

$$Lift(X \Rightarrow Y) = \frac{sup(X \cup U)}{sup(X) * sup(Y)} \tag{5.8}$$

The lift of a rule is defined as, the ratio of the observed support to that expected if X and Y were independent. If some rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events. If the lift is greater 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in given dataset.

5.2 Implementation environment

For implementing the project, Window 8 operating system is used. For the front end, Python was used. More specifically, version 3.5.2 is used. The project does not has any back end. Project is also compatible on other Windows operating system and also on Ubuntu.

5.3 Flow of system development

The flow of proposed system is shown in the Figure 5.1. In this system, grocery dataset use for classification. Firstly for finding interesting patterns (IP), apply two methods namely support and cohesion. The product of Support and Cohesion will provide us with Interestingness measure over a class of sequential patterns. i.e.

$$IP = support * cohesion \tag{5.9}$$

Once discovered all interesting patterns, the next classification of data. Define, $\Gamma = \rho \Rightarrow \beta$ as a classification rule where ρ is an interesting pattern and β is a class label. ρ is the antecedent of the rule and β is the consequent of the rule. The redundant rules occupy extra space and take more time for classification. After identify all classification rules, pruning of rules is to be done for redundant rules. This pruning of rules is done is using "Lift" Method. The lift of a rule is defined as, the ratio of the observed support to that expected if the antecedent and consequent were independent. If some rule had a lift of less than 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events. If the lift is greater than 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those

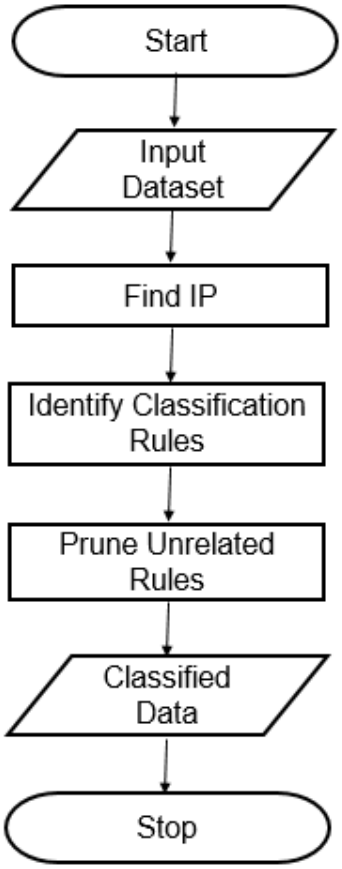

Figure 5.1: Flowchart of Proposed System

rules potentially useful for predicting the consequent in data sets. In the next step SCIP-MA (Sequence classification based on interesting pattern-Matching cohesion rules based classifier) algorithm use for build the classifier. In system, as a result the grocery dataset is labeled into four categories: Dairy Product, Bakery Product, Fruits and Vegetables. These types are classified based on classification rules.

Algorithm 1: Lift Method

Input ⇒Interesting Pattern, set of values calculated by lift. For each rule R in IP.
Output ⇒A set of prune rules PR.
Step 1: Choose co-relation of association rule based on 'Lift'.

1. If (L>1), positive correlation α (1+α)
2. If (L<1), negative correlation β (1-β)
3. If (L=1), antecedent and consequent are independent of each.

Step 2: Prune the rules having value 1.
Step 3: Store the values of α and β for each rule.
Step 4: Scan the data set.
Step 5: Classify the rule on the basis of the values of α and β.
Step 6: Store the rules on the basis of priority.

Algorithm 2: SCIP-MA Classifier

Input: PR, $default_r$, a new unclassified data object d = (s, $L_?$)
Output: a class label
Step 1: N=0
Step 2: foreach rule r in PR do

1. r matches d then store r into NP

Step 3:if N >0 then
Step 4:foreachrule r: P = C_K in N do
Step 5:use $SCIP_MA$ then
Step 6:r.value = r.lift $*$ C(P,d·s)
Step 7:sort rules in N by descending r.value
Step 8:CR = the top rules in sorted N
Step 9: Score = a new array of size —L—
Step 10:foreach rule r : P= C_K in CR do
Step 11:score[K] = score[K] + r.value
Step 12:return the class label C_K with largest score[K]
Step 13:else return the class label of default r.

5.4 Summary

In this chapter, system implementation is described. In the next chapter, system testing is presented.

Chapter 6

System Testing

This chapter gives testing of proposed system.The document that describes the steps to be taken in running a set of tests and specifies the executable order of the tests is called a test procedure. Testing plays a critical role for quality assurance and for ensuring the reliability of the software. Its basic function is to detect the errors. After the coding phase, testing is done to test the proper working of the new system. Testing is the process of executing a program with the intention of finding errors. It is a complete verification to determine whether the objectives are met and the user requirements are satisfied. The testing phase involves testing of a system using various test data. Preparation of the test data plays a vital role in the system testing. After preparing the test data, the system under study is testing using those test data. Errors were found and corrected by using the following testing steps and corrections are recorded for future references. Thus, a series of testing is performed on the system before it is ready for coding. Since code is the only product that can be executed frequently whose actual behavior can be observed, this phase is so important for the successful implementation of the software product. Thus, the goal of testing is to uncover the requirements, design and coding errors in the program [6].

In section 6.1 implementation of testing is described. The test cases and test results are described in section 6.2. In the last section summary is presented.

6.1 How to implement testing

The document that describes the steps to be taken in running a set of tests and specifies the executable order of the tests is called a test procedure in IEEE 829, and is also known as a test script. When test Procedure Specification is prepared then it is implemented and is called Test implementation. Test script is also used to describe the instructions to a test execution tool. An automation script is written in a programming language that the tool can understand.

Test cases are planned in accordance to the test process and documented with detailed

test descriptions. These test cases use cases based on projected operational mission scenarios.The testing process also includes stress and load testing for stability purpose process thoroughly tests the interfaces and modules. Software testing includes a traceable white box testing, black box testing and other test processes verifying implemented software against design documentation and requirements specified.

The testing has to be implemented on level by level basis, starting from the lowest level wherein only individual modules are to be tested, then testing has to be performed on integrated modules and finally the entire system has to be tested.

6.1.1 Unit Testing

Unit testing refers to testing an individual module. The unit testing corresponding to the project tests proposed system can be accepted data set. The next test is support,cohesion and lift methods provide correct value according to the formula. The next test is to check whether the system can be prune classification rules in proper way depend on lift value or not.

6.1.2 Integration testing

The integrated testing involves integration of some modules and then testing them. Referring to the project, the integration testing refers to whether interesting patterns are related to dataset or not, form product of support and cohesion.Interestingness is finding after getting the product value of support and cohesion methods. The next test is to check whether the system can be prune classification rules in proper way depend on lift value or not. Next after checking all pruning classification rules, in last classification is tested.

6.1.3 Manual Testing

It is the oldest and most rigorous types of testing it is performed by human sitting in front of a computer carefully going through application screens, trying various usage and input combinations, comparing the results to the expected behavior and recording and observations about project. The manual testing corresponding to project is to test whether that system give accurate classification of dataset. In manual testing firstly different test cases are written. Based on different test cases project is tested to check accurate classification.

6.1.4 White box testing

A level of white box test coverage is specified that is appropriate for the software being tested. The white box and other testing uses automated tools to instrument the software

to measure test coverage. The program code is examined for defects. It based on design and implementation structures. The white box testing corresponding to project is to test internal logic and structure of the code.

6.2 Test cases and Test Results

Project Name: Data Classification Using SCIP-MA Algorithm.
Test case Template
Pre-condition: User should give the dataset(grocery dataset) as input.
Post-condition: Type of classification.

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. Different test cases for different modules are identified and their results are rendered in tabular form.The test cases for various modules and their results are presented in the Table 6.1.

Table 6.1: Test Cases and Result

Test ID	Test Case	Expected Result	Actual Result	Test Result
1	Support value	value must be as per formula	Provide expected result	Pass
2	Cohesion value	Value must be as per formula	Provide expected result	Pass
3	Interestingness	Value must be Product of support and cohesion	Provide expected result	Pass
4	Lift value	value must be as per formula	Provide expected result	Pass
5	Classification	Classification must be according to rules	Provide expected result	Pass

6.3 Summary

In this chapter, system testing is described. In the next chapter, result and analysis is presented.

Chapter 7

Results and Analysis

Results that illustrate how the system designed by you works in practice, and how it is intended to be used. Analysis summarizes the qualitative and quantitative analysis that explains why results are relevant. The chapter focuses on results generated by the system and the analysis of the results.

From result of proposed system it is observed that the system is more versatile than existing technique. Consider the example of grocery (Market Basket Analysis) dataset: fruits, banana, orange, apple, grape, pineapple, and mango, peach here, fruit is a class label and banana, orange etc. are all items in sequence. First finding interestingness of itemset using support and cohesion terms. Consider the itemset X= banana, orange then the support value of X is 0.120690 and 2 is cohesion value. Product of support and cohesion provide an interestingness value of given itemset i.e. interestingness value of above itemset is 0.241379. In the next step find the classification rule from interestingness and then prune unrelated classification rules with the help of lift method. The lift value for the given itemset is 2.197802. Lift value of X is greater than 1 hence make rule potentially useful for predicting the consequent in data sets. Depend on lift value the classification rules are prune and only related classification rules are used for data classification. In existing technique the confident method id used for identifying classification rule, which contain independent classification rules. Due to independent rules system take more time for classification while using lift, classification time is reduced. Hence classification using lift method improve the proposed system performance.

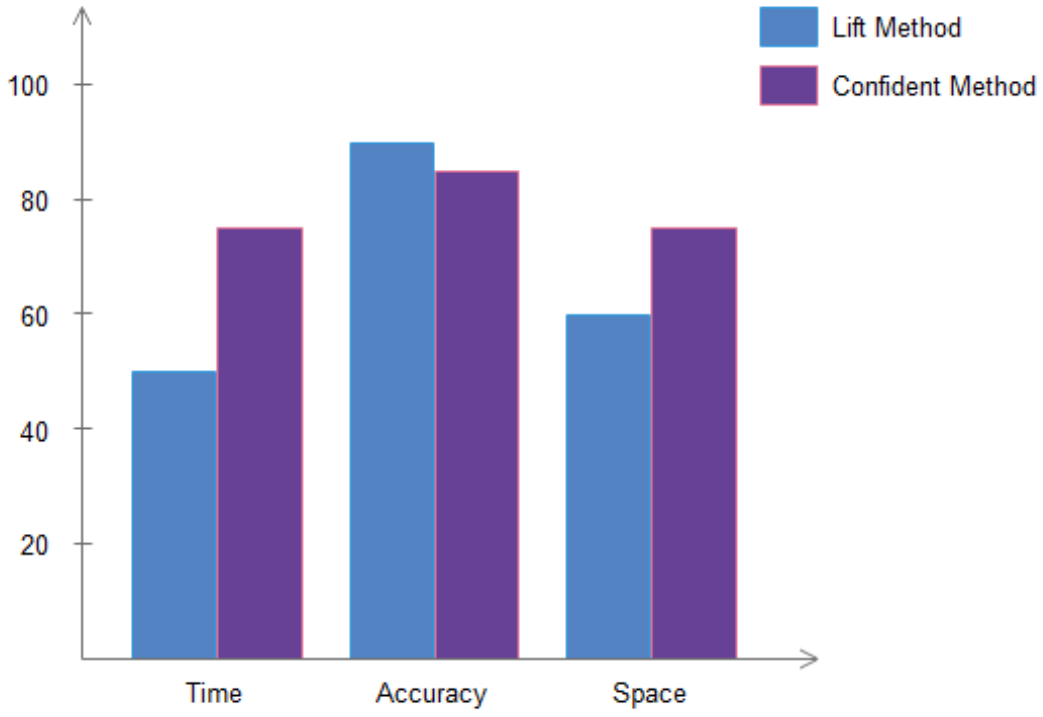

Figure 7.1: Result Analysis

Chapter 8

Conclusion and Future Scope

In the proposed work, a classification is done by using Sequence Classification based on Interesting Pattern Matching cohesion rule based classifier (SCIP-MA).For classification pruned rules are used. Classification rules generated by interesting pattern also contain unrelated rules. The lift method is used to prune the unrelated classification rule. It is observed that lift decreases the time required for classification of data as compared to existing system. The proposed work is useful in various data mining application such as Market Basket Analysis.

For future work other technique can be applied for identify classification rules.

Bibliography

[1] R. Agrawal and R. Srikant. Mining sequential patterns. *in Proceedings of the 11th International Conference on Data Engineering,*, pages 3–14, 1995.

[2] B. C. Cheng Zhou and B. Goethals. pattern based sequence classification.

[3] J. G. J. Ayres, J. Flannick and T. Yiu. Sequential pattern mining using a bitmap representation. *in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM*, pages 429–435, 2002.

[4] J. H. J Pei and M. C. H. et. Mining sequential patterns by patterngrowth: The prefixspan approach. *IEEE Trans. Knowl. Data Eng*, 16(11):1424–1440, 2004.

[5] M. J. Z. N. Lesh and M. Ogihara. Scalable feature mining for sequential data. *IEEE Intell. Syst*, 15(2):4856, 2000.

[6] R. S. Pressman. Software engineering - a practitioners approach. *McGraw-Hill Series in Computer Science*, 2001.

[7] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. *in Proceedings of the 5th International Conference on Extending Database Technology.Springer-Verlag*, pages 3–17, 1996.

[8] C. P. T. P. Exarchos, M. G. Tsipouras and D. I. Fotiadis. a two-stage methodology for sequence classification based on sequential pattern mining and optimization. *Data and Knowledge Engineering*, 66(3):467–487, 2008.

[9] V. S. Tseng and C.-H. Lee. Effective temporal data classification by integrating sequential pattern mining and probabilistic induction. *Expert Systems with Application*, 36(5):95249532, 2009.

[10] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.