

An Efficient Rule-Mining for Medical Diagnosis: A Market-Basket Approach

¹AWODUN, Mojirade Adejumoke & ²ADEDARA, Raphael Oluwadare

¹Department of Computer Science, University of Ilorin, Ilorin.Kwara State, Nigeria.

² Department of Computer Science, College of Education, Ikere-Ekiti, Ekiti State, Nigeria.

Received: February 01, 2020

Accepted: March 03, 2020

ABSTRACT: *Medical diagnosis is an important and also delicate and time involving task that needs to be carried out in order to accurately identify a disease a patient might be suffering from. Due to the inaccurate diagnosis given to patients which could result in a wrong prescription that may lead into unwelcome consequences therefore to help in this direction an automated system is developed which is able to generate the association rules among different symptoms which will also assist the medical personnel in diagnosing patient's diseases accurately. In actualizing this aim, an analysis technique in data mining which is Market Basket Analysis using bitmap representation of Transaction table is adopted in revealing the association rule among the symptoms and diseases given a support and confidence level. The rules generated are compared with a doctor's prescription to determine its relevance in healthcare, the irrelevant ones are then filtered out to ascertain the ones which are acceptable for diagnosis of the various diseases.*

Key Words: Rule-Mining, Medical Diagnosis, Market-Basket and Market-Basket Approach

Introduction

Good medical care relies solely on well-trained doctors and nurses and on high-quality facilities and equipment. Good medical care also relies on adequate record keeping because without accurate, up-to-date and accessible patients case notes, medical personnel may not offer the best treatment or may misdiagnose a condition which can lead to a very severe consequences like death.

In our society today, a large number of deaths are recorded daily most especially in the underdeveloped and urban areas which is due to inadequate medical facilities. However in recent years especially this 21st century, some hospitals have introduced computer in their services. Notwithstanding, the use of computer in the health services is mainly on some housekeeping functions such as accounting, costing, store control, billing, medical record keeping, payroll and so on. Much emphasis has not been made in the area of medical diagnosis.

For several years now most especially in Nigeria, the idea of using a computer for medical diagnosis has been partially left out in the process of automating or computerizing health services. Medical activities in Nigeria are still mostly by manual method. "Health is wealth" goes a popular saying, and this is one of the reasons the medical aspect of life should not be taken with levity.

Misdiagnosis has been explained as a failure or delay in diagnosis. In diagnosis, errors can occur as a result of single or various flaws within the healthcare system such flaw can include:

- Patients inability to express themselves which leads to faulty data gathering;
- Wrong interpretation of results;
- Not considering all other possible diagnosis;
- Failure to verify diagnosis and gather new data e.t.c.

These and many other flaws have resulted in misdiagnosis which has caused inappropriate or inadequate treatment which in turn has led to different fatal consequences. It has been discovered that no healthcare in the world is free from the unfortunate issue of making wrong diagnosis of diseases.

Due to the increase of data in our society today, most organizations and big companies have large databases that contain a great amount of undiscovered and valuable information. The huge amount of data in the database makes it very difficult to examine and discover knowledge of interest with the manual process. This is due to the fact that the quantity of data in the world roughly doubles every year, but the astonishing fact is that the amount of relevant and useful information decreases rapidly which has now led to emergence of data mining technologies. Jackson (2002) opined that just like the real world mining, the "information gold nuggets" or "diamonds" can be automatically found by sifting out massive quantities of data-wreckage from our database through data mining with the computer.

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining is supported by a variety of techniques such as statistical methods, rule induction, neural networks, conceptual clustering, association discovery, etc. Information extraction can be enhanced by the cooperative use of several data mining techniques (Fawzi et al, 2011). Association rule mining which is a technique in data mining, finds interesting associations and/or correlation relationships among large set of data items. Association rules show attributes value conditions that occur frequently together in a given dataset. A typical and widely-used example of association rule mining is Market Basket Analysis.

Market-Basket Analysis is an advanced predictive model used to determine associations between items or events and deriving rules that indicate the likelihood of items occurring together in groups of specific types. It also identifies items that are unlikely to occur together. Apart from discovering the associating rules, Market-basket analysis also assist in visualizing the degree of attraction between items, view charts representing item associations, rank association rules and applying them to new data to produce recommendations. Market basket analysis is a data mining methods used in retail companies which focused on discovering purchasing patterns by extracting associations or co-occurrences from a store's transactional data.

In healthcare, data mining is becoming increasingly trendy and also indispensable. So many factors have motivated the use of data mining applications in health services. For example, the existence of medical insurance fraud and abuse has led numerous healthcare insurers to attempt to reduce their losses by using data mining tools to help them find and track offenders. Detecting fraud using data mining applications is prevalent in the commercial world, for example, by detecting fraudulent credit card transactions. The huge amounts of data generated by healthcare transactions are too complex and voluminous to be processed and analyzed by traditional methods.

Fawzi et al (2011) affirmed that there are various natural arguments among the traditional ways of data mining strategies and medicine in past researches. He also affirmed that Data Mining in the field of medical research starts with an assumptions thereby altering the outputs so as to be able to achieve the previous assumptions. This process is different in the case of standard data mining because here the datasets does not begins with assumptions this is because standard data mining lacks clarifying development and models and focuses mainly on relating items. He also stated that Healthcare institutions should utilize the process of data mining to be able to gain more facts and knowledge from the data that is present in their institution records.

The impact of data mining in health care nowadays is great which has made data mining essential. This is because the health organization generates and gather large amounts of data on daily basis. This gives room for data mining which makes it possible to extract data that helps in getting interesting rules, knowledge and regularities. The knowledge extracted in the data can help in improving the work efficiently and also enhance quality decision making. Data mining assist healthcare organization in predicting trends in patient conditions and their behaviors. It also helps in early detection of contagious diseases and their treatments. Data mining can be limited in data access this is owed to the fact that the unprocessed inputs for data mining frequently exist in different settings and systems (Borris, 2012).

Rani (2011) demonstrated in his research the use of neural networks in the analysis of Heart diseases dataset to generate efficient classification rules. The neural network is skilled using Back propagation algorithm to be suitable in performing classification task of medical data. It is also stated in his research that data mining algorithms can gain knowledge from previous examples in clinical records and model the often times non-linear relationships between the independent and dependent variables. Neural networks have been applied to various areas of healthcare like diagnosis areas, image analysis, drug application etc. It is also stated in the research that neural networks are more vigorous in noisy and incomplete data. As advantageous as the neural network is the structure of the neural network gotten for a particular problem cannot be specify due to the fact that the structure is often learnt by experiment and it is also difficult to understand how it works.

Mohammed et al (2013) asserted in a paper which focuses on analysis of different data mining techniques needed for medical data mining especially in discovering locally frequent diseases like heart ailments, cancer of the lungs and breast cancer etc. that data mining is the process of digging inside data for the purpose of discovering latent patterns which can be translated into useful information. The various data mining techniques which has been used in medical data are Apriori Algorithm, FP Growth, unsupervised neural networks, Decision trees, linear programming, Association rule mining. The latter (i.e Association rule mining) is used in discovering frequently occurring items in the given dataset. In his research the Apriori Algorithm is used for discovering frequent patterns for finding locally frequent disease and also

visualizing the experimental results in various view points. Medical mining yields required business intelligence to support well informed diagnosis and decisions making.

Sona et al (2014) affirmed that the Healthcare database consists of different records of patient information from the hospitals. When these data are mined properly, it reveals hidden patterns which can lead to better diagnosis of different diseases and also helps in medicines treatment. According to Jacqulin et al (2013), applying data mining on medical institution's existing data helps to discover new, useful and potentially life- saving knowledge which would have otherwise remained hidden in their databases. He also proposed a system which will help in detecting discrepancies in medical data. The data mined is cleaned to remove unwanted, inconsistent, noisy and corrupt data thereby making the data to be of high quality data which is potentially more accurate and also prevents occurrences of fraud in case of false health sale and billings. Data mining can also help in detecting medical fraud such as providing false and intentionally misleading statements to patients, falsify medical records or reports, unnecessary medical treatment or drug prescription and also detecting false medicine composition. Data mining has the ability to generate a knowledge rich environment which can help to significantly improve the quality of clinical decisions.

Srinivas, Rao and Govardhan (2012) affirmed that data mining aims at discovering knowledge out of data and presenting it in a form that is easily compressible to humans. It is a process that is developed to examine large amounts of data routinely collected. Similarly, Rani (2011) affirmed that data mining is an interdisciplinary field and its main aim is to uncover relationship in data and to predict outcomes. Diverse data mining problems can be solved adequately by soft computing techniques. These techniques are Clustering, Decision tree, fuzzy logic, neural network, genetic algorithms and rough sets which will lead to an intelligent, interpretable, low cost solution than traditional techniques.

Association Rule Mining in Health Care

Association Rule is one of the most important techniques of data mining that is used to detect the frequent patterns or rules, interesting relationships among a set of data items in the data repository. It is also known as market basket analysis due to its ability to discover the association among purchased item or unknown patterns of sales of customers in a transaction database. Association rule also plays a vital role in the healthcare field to detect the relationships among diseases, health state and symptoms. Association Rule is widely used by Healthcare organization for discovering relationships between various diseases and drugs. It is also used in detecting fraud and abuse in health insurance. It is also used with classification techniques to enhance the analysis capability of Data Mining.

Han et al (2006) proposed in his research that Association rule mining process could be divided into two main phases to enhance the implementation of the algorithm.

The phases are:

1. Frequent Item Generation: This is to find all the itemsets that satisfy the minimum support threshold. The itemsets are called frequent itemsets.

2. Rule Generation: This is to extract all the high confidence rules from the frequent itemsets found in the first step. These rules are called strong rules.

Reasons for using Support count and Confidence Rule

Support count is an important measure because a rule that has very low support may occur simply by chance. A rule with a low support count is likely to be uninteresting from a business perspective because it may not be profitable to promote items that are frequent together. Therefore, support count is often used to eliminate uninteresting rules. It also has a desirable property that can be exploited for the efficient discovery of association rules.

Confidence Rule on the other hand, measures the reliability of the inference made by a rule. For a given $A \rightarrow B$, the higher the confidence, the more likely it is for B to be present in transactions that contain A. confidence also proves an estimate of the conditional probability of B given A.

Importance and Uses of Data Mining in Medicine

Data Overload: The health record is rich of knowledge, but the data is so huge that it seems that one is being drowned inside the data and is actually starving for knowledge.

Evidence based medicine and prevention of hospital errors: Medical institutions that applies data mining on their existing data discover new, useful and potentially life saving knowledge that would have remain lifeless in the database if not discover.

Policy making in public health: GIS and data mining combined tools like (free, open source, java- based data mining tools), to analyze similarities between different aspect of the medical record.

Objectives of the Study

The objectives of this study are:

1. To study the healthcare data i.e. the various occurrences of diseases and the symptoms that comes with it;
2. To enhance the system used for diagnosis for better performance of medical personnel in the diagnosis of disease; and
3. To evaluate the efficacy of the system developed.

Misdiagnosis has been said to occur due to cognitive error, that is, failure to consider all likely diagnoses. With the increase in biomedical knowledge it is humanly impossible for a physician to remember all the various symptoms associated with all diagnostic possibilities. Diagnosis errors are usually errors of omission which are difficult to identify unlike most medical errors which are usually errors of commission. Sometimes misdiagnosis cases surface through malpractice litigation, morbidity and mortality conferences and uncertain feedbacks from patients and physicians, (Kenneth et al, 2013).

Misdiagnosis is a serious and common occurrence in the healthcare in which the repercussions can damage a patient's health and cost money, or even a life. Not all misdiagnosis are due to physician's negligence, skillful doctors can and do make diagnosis error even when using reasonable care.

In the Healthcare, many facts can be achieved from the past data stored in computers, since the data are huge in quantity it is very strenuous for an individual to examine the whole data and gain awareness through the traditional method of viewing files. Therefore, this research is applying a Market- basket analysis technique which is a technique in data mining into medical records to identify association between diseases and symptoms in order to assist in solving the problem of misdiagnosis.

The aim of this study was to design a system to assist the medical practitioners in the diagnosis of preventable disease. This was done by using market- basket analysis technique with the bitmap based algorithms which would help to analyze different diseases and determine patterns in the symptoms that always occurred together.

Methodology

The nature of data used is based on diseases and symptoms. The information is gathered under supervision of expertise doctors and from the patients records. The data used are based on different patients suffering from ten (10) specified diseases which are:

1. Influenza
2. Measles
3. Malaria
4. Typhoid
5. Cholera
6. Typhus
7. Yellow fever
8. Pneumonia
9. Gastroenteritis
10. Dengue fever

The data contain hundred (100) different cases of each diseases and their symptoms.

The Healthcare data consists of a set of patients' records or data such as the name, sex, address and some other categorical information. The diverse information contained in the patients record might not be suitable for data mining. So therefore, there is need for data preparation before the data mining process can begins. The data preparation process in this study includes: data cleaning and data transformation.

Data Management: the data gotten are grouped according to their types for easy accessed. The patients record used are stored differently due to their transaction type (i.e. disease classification). Ten (10) different diseases are used for this research and for each disease one hundred (100) different cases are chosen.

Data Mining: The association rule mining algorithm is applied on the medical data sets generated. Here the Basket analysis technique is applied to be able to identify the relationships between data. It detects significant and concise medical association rules to be used in medical diagnosis.

Association Rule mining technique works in two main steps:

1. It searches through the items to discover the ones having frequency count which surpass the user-defined support count. This set is referred to as Frequent itemset that is a set of items that has a support higher than a user-supplied minimum support.
2. It discover item sets (or patterns or rules) meeting the second user-defined frequency count. This are item set meeting the specified confidence level. The item set generated at this step are the final rules generated.

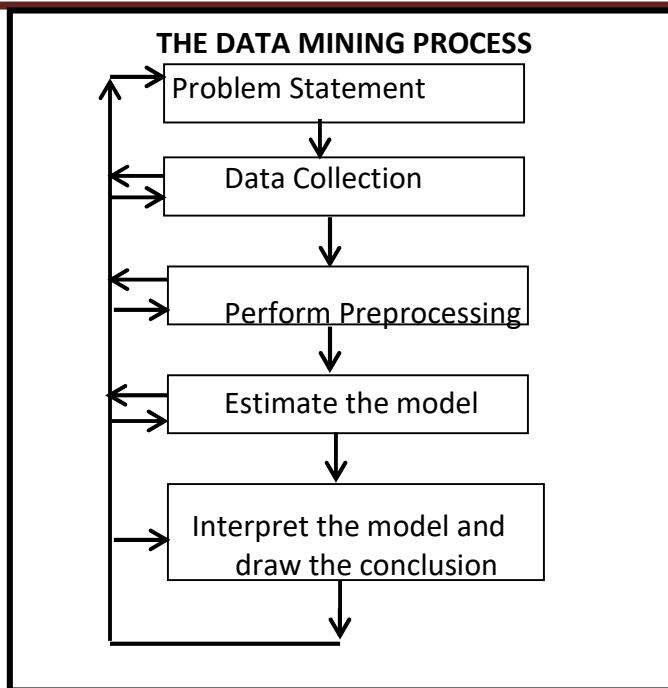


Fig. 1: The data mining Process

Apriori Algorithm

The main aspect of the Apriori algorithm which is the basis of this analysis is grounded on the rule of identifying associations between items in a transactional database by applying the rule of sufficient repeat occurrence of a particular item in transaction data set which is enough to sustain the minimum support count. The Algorithm works as follow:

Apriori Generate Algorithm

C_k : Candidate item set of size k

L_k : Frequent item set of size k

$L_1 = \{\text{Frequent items}\}$;

For ($k = 1$; $L_k \neq \emptyset$; $K++$) do begin

C_{k+1} = candidates generated from L_k ;

For each transaction t in database do increment the count of all candidates in C_{k-1} that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

End

return $U_k L_k$;

APRIORI ALGORITHM FLOW CHART

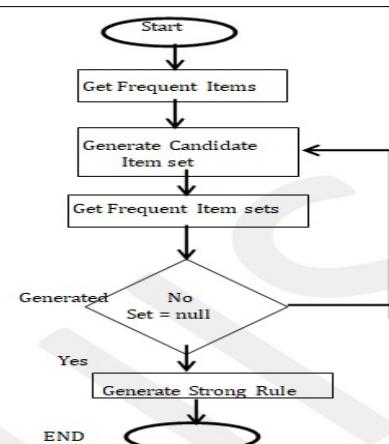


Fig 2: Apriori Flow chart

- STEP 1: The Algorithm first check through the transaction dataset to detect the number of items in singularity and the frequency of occurrence for the item.
- STEP 2: The Items are identified and the first candidate sets are made and frequency counts are also list down
- STEP 3: The candidate set is then checked with the minimum support count provided in order to identify the Items with the required support count.
- STEP 4: The items that fall below the requirements are eliminated, while the remaining items that meet up the requirement are process again to create another candidate set that will check their occurrence in different group (e.g. group of two, three and so on).
- STEP 5: Each time a candidate set is created it is check with the provided support count to see if its surpass the minimum threshold value and if it does not then the process of generating new candidate set is terminated
- STEP 6: The Association Rule generated from the Item set is then compared to the threshold value of the particular transactional Dataset to identify the Strong and Weak association rules.

Technique used for generating the Rules

Market Basket Analysis : This analysis method makes use of the horizontal database (i.e. each transaction has its items listed out in rows). The itemset are represented in a bitmap format (i.e. each item has a bit position assign to it to indicate the presence and absence of the items ("1" or "0") respectively. Items are therefore generated from the bitmap by performing logical AND operation on each column (itemset) and the support count of each column is calculated wherefore any column less than the given support count is dropped or discarded. This analysis method make use of algorithm which only scan the database once unlike the Apriori algorithm which takes more time in finding the frequent itemsets, because it needs to scan the database again and again which makes it to be a time consuming process.

The basic concepts of data mining can be explained using a supermarket data as an example.

No.	Items purchased
001	MILK, TEA, SUGAR, CORNFLAKES
002	MILK, CORNFLAKES, SUGAR
003	MILK, TEA, SUGAR
004	MILK, TEA, BREAD, CORNFLAKES
005	MILK, TEA, BREAD, CUSTARD

Table 1: a small set of transaction

The above table is used to show the transaction performed by five different customers. Each row of the table is referred to as a transaction. This table will be used in illustrating the concepts involved in data mining.

1. An **Association rule** is a pair of disjoint itemsets. If A and B denotes the two disjoint itemsets, the association rule is written as $A \rightarrow B$
2. The **support count** of the association rule $A \rightarrow B$ with respect to a transaction set "C" is the support of the itemsets $A \cup B$ with respect to C.
3. The **confidence** of the rule $A \rightarrow B$ with respect to a transaction set "C" is the ratio support $(A \rightarrow B) / \text{support}(C)$.

The given association rules $A \rightarrow B$ is consider significant if it has a high support count and high confidence. For a stated the support count, the rules with higher confidence values are more significant than those with lower confidence values. It can be seen from the given definition above that the confidence value is at most 1, therefore they are measured in percentage rather than ratio.

This analysis method is implemented in Java language. This algorithm divides the database into different units and after which the results of the different processing units are combine to produce a single results. In medical analysis, market basket analysis is applied on the data separately in order to get accurate results.

A preprocessing step is used to convert the raw data into a form similar to market basket data. According to the market basket data discussed above, the set of all the symptoms and disease diagnosis codes referred to the set of items. The set of the symptoms and disease codes for each patient is considered to be a transaction. In other words, each itemset constitute of one or more disease codes. Therefore, the support count of an itemset C is the number of patients (transaction) whose set of items corresponds to the items in C. The major goal of this research is to discover the relationships between diagnosed diseases and

their corresponding symptoms. A natural way of discovering this rule is finding the association rules of the form A \rightarrow B, where A is a set of symptoms codes and B is a set of disease diagnosis code. For this rule to be significant it must constitute a high support count and high confidence rule.

System Requirements

The system Requirement specifies the hardware and software requirements for carrying out this project. The following are the hardware and the software requirements.

- Hardware Requirements
 - a. Processor: Intel(R) core(TM) i7- 2640M CPU @ 2.80 GHz
 - b. RAM: 4GB
 - c. System type: 64-bit Operating S
- Software Requirement

The researcher made use of

 - a. Windows 7
 - b. 64- bit Operating System
 - c. Microsoft Office (Windows Excel)
 - d. Java Net bean version 8.02

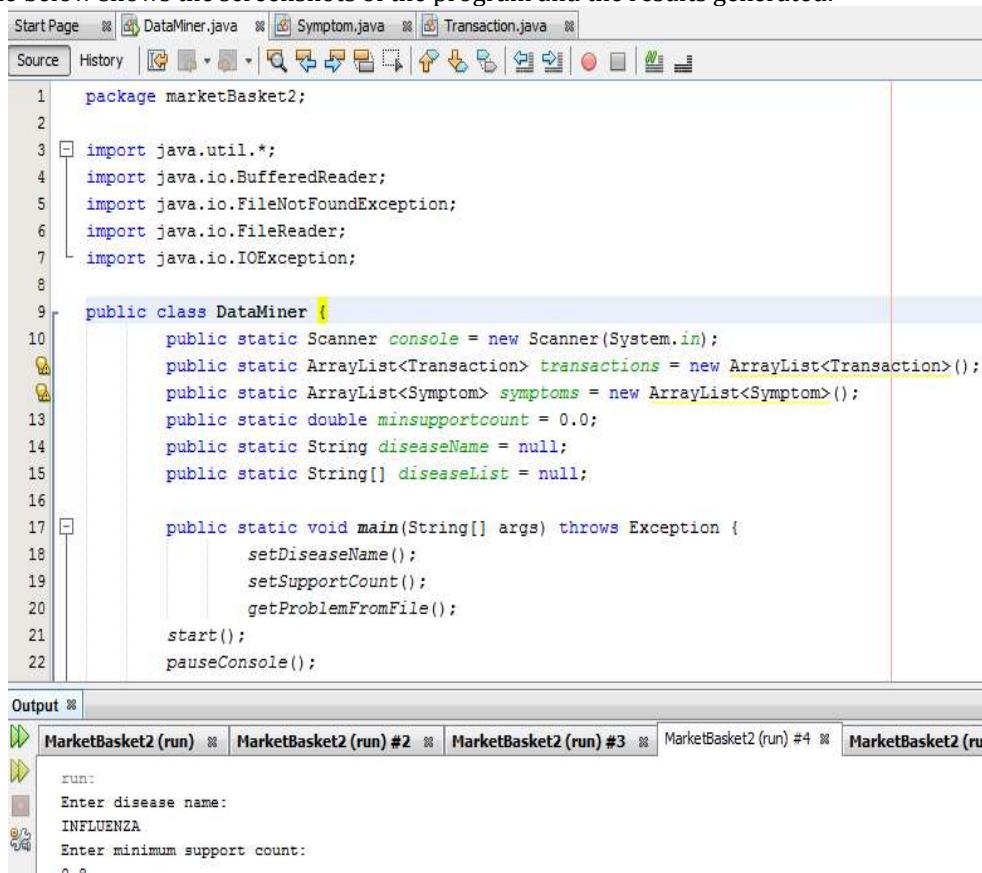
Results and Discussion

The data is represented in two formats; the generated format and the transformed format

- The generated format is the original format in which the data is generated.
- The transformed format is the transaction format (that is the bitmap mode) which makes it suitable for discovering the frequent items in them.

The Program is designed in such a way that, there is an enter items command which will request for the disease that the physician wants to get information about with the minimum support and minimum confidence threshold for the generation of the rules.

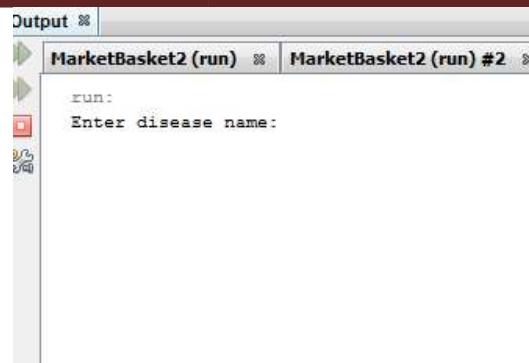
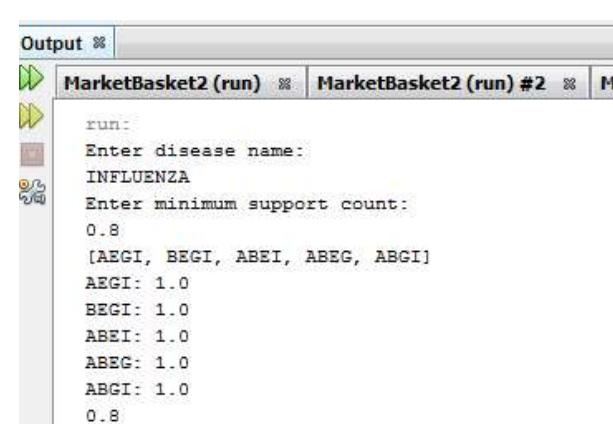
The below shows the screenshots of the program and the results generated:



The screenshot shows the Java code for a DataMiner class in NetBeans. The code imports various Java utilities and defines a DataMiner class with a main method. The output window shows the execution of the program, prompting for a disease name (INFLUENZA) and a minimum support count (0.0).

```

1 package marketBasket2;
2
3 import java.util.*;
4 import java.io.BufferedReader;
5 import java.io.FileNotFoundException;
6 import java.io.FileReader;
7 import java.io.IOException;
8
9 public class DataMiner {
10
11     public static Scanner console = new Scanner(System.in);
12     public static ArrayList<Transaction> transactions = new ArrayList<Transaction>();
13     public static ArrayList<Symptom> symptoms = new ArrayList<Symptom>();
14     public static double minsupportcount = 0.0;
15     public static String diseaseName = null;
16     public static String[] diseaseList = null;
17
18     public static void main(String[] args) throws Exception {
19         setDiseaseName();
20         setSupportCount();
21         getProblemFromFile();
22         start();
23         pauseConsole();
24     }
25
26     public static void pauseConsole() {
27         System.out.println("Press any key to continue . . .");
28         try {
29             System.in.read();
30         } catch (IOException e) {
31             e.printStackTrace();
32         }
33     }
34
35     public static void setDiseaseName() {
36         System.out.print("Enter disease name: ");
37         diseaseName = console.nextLine();
38     }
39
40     public static void setSupportCount() {
41         System.out.print("Enter minimum support count: ");
42         minsupportcount = Double.parseDouble(console.nextLine());
43     }
44
45     public static void getProblemFromFile() {
46         try {
47             BufferedReader reader = new BufferedReader(new FileReader("MarketBasket2.txt"));
48             String line;
49             while ((line = reader.readLine()) != null) {
50                 String[] items = line.split(",");
51                 Transaction transaction = new Transaction();
52                 for (String item : items) {
53                     transaction.addItem(item);
54                 }
55                 transactions.add(transaction);
56             }
57             reader.close();
58         } catch (FileNotFoundException e) {
59             e.printStackTrace();
60         } catch (IOException e) {
61             e.printStackTrace();
62         }
63     }
64
65     public static void start() {
66         System.out.println("Market Basket Analysis");
67         System.out.println("-----");
68         System.out.println("Disease Name: " + diseaseName);
69         System.out.println("Minimum Support Count: " + minsupportcount);
70         System.out.println("Transactions: " + transactions.size());
71         System.out.println("Frequent Itemsets: " + frequentItemsets);
72         System.out.println("Rules: " + rules);
73     }
74
75     public static void frequentItemsets() {
76         System.out.println("Frequent Itemsets: ");
77         for (String itemset : frequentItemsets) {
78             System.out.println(itemset);
79         }
80     }
81
82     public static void rules() {
83         System.out.println("Rules: ");
84         for (String rule : rules) {
85             System.out.println(rule);
86         }
87     }
88
89     public static void pauseConsole() {
90         System.out.println("Press any key to continue . . .");
91         try {
92             System.in.read();
93         } catch (IOException e) {
94             e.printStackTrace();
95         }
96     }
97
98     public static void main() {
99         DataMiner.main(null);
100    }
101
102    public static void main(String[] args) {
103        DataMiner.main();
104    }
105
106    public static void main() {
107        DataMiner.main();
108    }
109
110    public static void main() {
111        DataMiner.main();
112    }
113
114    public static void main() {
115        DataMiner.main();
116    }
117
118    public static void main() {
119        DataMiner.main();
120    }
121
122    public static void main() {
123        DataMiner.main();
124    }
125
126    public static void main() {
127        DataMiner.main();
128    }
129
130    public static void main() {
131        DataMiner.main();
132    }
133
134    public static void main() {
135        DataMiner.main();
136    }
137
138    public static void main() {
139        DataMiner.main();
140    }
141
142    public static void main() {
143        DataMiner.main();
144    }
145
146    public static void main() {
147        DataMiner.main();
148    }
149
150    public static void main() {
151        DataMiner.main();
152    }
153
154    public static void main() {
155        DataMiner.main();
156    }
157
158    public static void main() {
159        DataMiner.main();
160    }
161
162    public static void main() {
163        DataMiner.main();
164    }
165
166    public static void main() {
167        DataMiner.main();
168    }
169
170    public static void main() {
171        DataMiner.main();
172    }
173
174    public static void main() {
175        DataMiner.main();
176    }
177
178    public static void main() {
179        DataMiner.main();
180    }
181
182    public static void main() {
183        DataMiner.main();
184    }
185
186    public static void main() {
187        DataMiner.main();
188    }
189
190    public static void main() {
191        DataMiner.main();
192    }
193
194    public static void main() {
195        DataMiner.main();
196    }
197
198    public static void main() {
199        DataMiner.main();
200    }
201
202    public static void main() {
203        DataMiner.main();
204    }
205
206    public static void main() {
207        DataMiner.main();
208    }
209
210    public static void main() {
211        DataMiner.main();
212    }
213
214    public static void main() {
215        DataMiner.main();
216    }
217
218    public static void main() {
219        DataMiner.main();
220    }
221
222    public static void main() {
223        DataMiner.main();
224    }
225
226    public static void main() {
227        DataMiner.main();
228    }
229
230    public static void main() {
231        DataMiner.main();
232    }
233
234    public static void main() {
235        DataMiner.main();
236    }
237
238    public static void main() {
239        DataMiner.main();
240    }
241
242    public static void main() {
243        DataMiner.main();
244    }
245
246    public static void main() {
247        DataMiner.main();
248    }
249
250    public static void main() {
251        DataMiner.main();
252    }
253
254    public static void main() {
255        DataMiner.main();
256    }
257
258    public static void main() {
259        DataMiner.main();
260    }
261
262    public static void main() {
263        DataMiner.main();
264    }
265
266    public static void main() {
267        DataMiner.main();
268    }
269
270    public static void main() {
271        DataMiner.main();
272    }
273
274    public static void main() {
275        DataMiner.main();
276    }
277
278    public static void main() {
279        DataMiner.main();
280    }
281
282    public static void main() {
283        DataMiner.main();
284    }
285
286    public static void main() {
287        DataMiner.main();
288    }
289
290    public static void main() {
291        DataMiner.main();
292    }
293
294    public static void main() {
295        DataMiner.main();
296    }
297
298    public static void main() {
299        DataMiner.main();
300    }
301
302    public static void main() {
303        DataMiner.main();
304    }
305
306    public static void main() {
307        DataMiner.main();
308    }
309
310    public static void main() {
311        DataMiner.main();
312    }
313
314    public static void main() {
315        DataMiner.main();
316    }
317
318    public static void main() {
319        DataMiner.main();
320    }
321
322    public static void main() {
323        DataMiner.main();
324    }
325
326    public static void main() {
327        DataMiner.main();
328    }
329
330    public static void main() {
331        DataMiner.main();
332    }
333
334    public static void main() {
335        DataMiner.main();
336    }
337
338    public static void main() {
339        DataMiner.main();
340    }
341
342    public static void main() {
343        DataMiner.main();
344    }
345
346    public static void main() {
347        DataMiner.main();
348    }
349
350    public static void main() {
351        DataMiner.main();
352    }
353
354    public static void main() {
355        DataMiner.main();
356    }
357
358    public static void main() {
359        DataMiner.main();
360    }
361
362    public static void main() {
363        DataMiner.main();
364    }
365
366    public static void main() {
367        DataMiner.main();
368    }
369
370    public static void main() {
371        DataMiner.main();
372    }
373
374    public static void main() {
375        DataMiner.main();
376    }
377
378    public static void main() {
379        DataMiner.main();
380    }
381
382    public static void main() {
383        DataMiner.main();
384    }
385
386    public static void main() {
387        DataMiner.main();
388    }
389
390    public static void main() {
391        DataMiner.main();
392    }
393
394    public static void main() {
395        DataMiner.main();
396    }
397
398    public static void main() {
399        DataMiner.main();
400    }
401
402    public static void main() {
403        DataMiner.main();
404    }
405
406    public static void main() {
407        DataMiner.main();
408    }
409
410    public static void main() {
411        DataMiner.main();
412    }
413
414    public static void main() {
415        DataMiner.main();
416    }
417
418    public static void main() {
419        DataMiner.main();
420    }
421
422    public static void main() {
423        DataMiner.main();
424    }
425
426    public static void main() {
427        DataMiner.main();
428    }
429
430    public static void main() {
431        DataMiner.main();
432    }
433
434    public static void main() {
435        DataMiner.main();
436    }
437
438    public static void main() {
439        DataMiner.main();
440    }
441
442    public static void main() {
443        DataMiner.main();
444    }
445
446    public static void main() {
447        DataMiner.main();
448    }
449
450    public static void main() {
451        DataMiner.main();
452    }
453
454    public static void main() {
455        DataMiner.main();
456    }
457
458    public static void main() {
459        DataMiner.main();
460    }
461
462    public static void main() {
463        DataMiner.main();
464    }
465
466    public static void main() {
467        DataMiner.main();
468    }
469
470    public static void main() {
471        DataMiner.main();
472    }
473
474    public static void main() {
475        DataMiner.main();
476    }
477
478    public static void main() {
479        DataMiner.main();
480    }
481
482    public static void main() {
483        DataMiner.main();
484    }
485
486    public static void main() {
487        DataMiner.main();
488    }
489
490    public static void main() {
491        DataMiner.main();
492    }
493
494    public static void main() {
495        DataMiner.main();
496    }
497
498    public static void main() {
499        DataMiner.main();
500    }
501
502    public static void main() {
503        DataMiner.main();
504    }
505
506    public static void main() {
507        DataMiner.main();
508    }
509
510    public static void main() {
511        DataMiner.main();
512    }
513
514    public static void main() {
515        DataMiner.main();
516    }
517
518    public static void main() {
519        DataMiner.main();
520    }
521
522    public static void main() {
523        DataMiner.main();
524    }
525
526    public static void main() {
527        DataMiner.main();
528    }
529
530    public static void main() {
531        DataMiner.main();
532    }
533
534    public static void main() {
535        DataMiner.main();
536    }
537
538    public static void main() {
539        DataMiner.main();
540    }
541
542    public static void main() {
543        DataMiner.main();
544    }
545
546    public static void main() {
547        DataMiner.main();
548    }
549
550    public static void main() {
551        DataMiner.main();
552    }
553
554    public static void main() {
555        DataMiner.main();
556    }
557
558    public static void main() {
559        DataMiner.main();
560    }
561
562    public static void main() {
563        DataMiner.main();
564    }
565
566    public static void main() {
567        DataMiner.main();
568    }
569
570    public static void main() {
571        DataMiner.main();
572    }
573
574    public static void main() {
575        DataMiner.main();
576    }
577
578    public static void main() {
579        DataMiner.main();
580    }
581
582    public static void main() {
583        DataMiner.main();
584    }
585
586    public static void main() {
587        DataMiner.main();
588    }
589
590    public static void main() {
591        DataMiner.main();
592    }
593
594    public static void main() {
595        DataMiner.main();
596    }
597
598    public static void main() {
599        DataMiner.main();
600    }
601
602    public static void main() {
603        DataMiner.main();
604    }
605
606    public static void main() {
607        DataMiner.main();
608    }
609
610    public static void main() {
611        DataMiner.main();
612    }
613
614    public static void main() {
615        DataMiner.main();
616    }
617
618    public static void main() {
619        DataMiner.main();
620    }
621
622    public static void main() {
623        DataMiner.main();
624    }
625
626    public static void main() {
627        DataMiner.main();
628    }
629
630    public static void main() {
631        DataMiner.main();
632    }
633
634    public static void main() {
635        DataMiner.main();
636    }
637
638    public static void main() {
639        DataMiner.main();
640    }
641
642    public static void main() {
643        DataMiner.main();
644    }
645
646    public static void main() {
647        DataMiner.main();
648    }
649
650    public static void main() {
651        DataMiner.main();
652    }
653
654    public static void main() {
655        DataMiner.main();
656    }
657
658    public static void main() {
659        DataMiner.main();
660    }
661
662    public static void main() {
663        DataMiner.main();
664    }
665
666    public static void main() {
667        DataMiner.main();
668    }
669
670    public static void main() {
671        DataMiner.main();
672    }
673
674    public static void main() {
675        DataMiner.main();
676    }
677
678    public static void main() {
679        DataMiner.main();
680    }
681
682    public static void main() {
683        DataMiner.main();
684    }
685
686    public static void main() {
687        DataMiner.main();
688    }
689
690    public static void main() {
691        DataMiner.main();
692    }
693
694    public static void main() {
695        DataMiner.main();
696    }
697
698    public static void main() {
699        DataMiner.main();
700    }
701
702    public static void main() {
703        DataMiner.main();
704    }
705
706    public static void main() {
707        DataMiner.main();
708    }
709
710    public static void main() {
711        DataMiner.main();
712    }
713
714    public static void main() {
715        DataMiner.main();
716    }
717
718    public static void main() {
719        DataMiner.main();
720    }
721
722    public static void main() {
723        DataMiner.main();
724    }
725
726    public static void main() {
727        DataMiner.main();
728    }
729
730    public static void main() {
731        DataMiner.main();
732    }
733
734    public static void main() {
735        DataMiner.main();
736    }
737
738    public static void main() {
739        DataMiner.main();
740    }
741
742    public static void main() {
743        DataMiner.main();
744    }
745
746    public static void main() {
747        DataMiner.main();
748    }
749
750    public static void main() {
751        DataMiner.main();
752    }
753
754    public static void main() {
755        DataMiner.main();
756    }
757
758    public static void main() {
759        DataMiner.main();
760    }
761
762    public static void main() {
763        DataMiner.main();
764    }
765
766    public static void main() {
767        DataMiner.main();
768    }
769
770    public static void main() {
771        DataMiner.main();
772    }
773
774    public static void main() {
775        DataMiner.main();
776    }
777
778    public static void main() {
779        DataMiner.main();
780    }
781
782    public static void main() {
783        DataMiner.main();
784    }
785
786    public static void main() {
787        DataMiner.main();
788    }
789
790    public static void main() {
791        DataMiner.main();
792    }
793
794    public static void main() {
795        DataMiner.main();
796    }
797
798    public static void main() {
799        DataMiner.main();
800    }
801
802    public static void main() {
803        DataMiner.main();
804    }
805
806    public static void main() {
807        DataMiner.main();
808    }
809
810    public static void main() {
811        DataMiner.main();
812    }
813
814    public static void main() {
815        DataMiner.main();
816    }
817
818    public static void main() {
819        DataMiner.main();
820    }
821
822    public static void main() {
823        DataMiner.main();
824    }
825
826    public static void main() {
827        DataMiner.main();
828    }
829
830    public static void main() {
831        DataMiner.main();
832    }
833
834    public static void main() {
835        DataMiner.main();
836    }
837
838    public static void main() {
839        DataMiner.main();
840    }
841
842    public static void main() {
843        DataMiner.main();
844    }
845
846    public static void main() {
847        DataMiner.main();
848    }
849
850    public static void main() {
851        DataMiner.main();
852    }
853
854    public static void main() {
855        DataMiner.main();
856    }
857
858    public static void main() {
859        DataMiner.main();
860    }
861
862    public static void main() {
863        DataMiner.main();
864    }
865
866    public static void main() {
867        DataMiner.main();
868    }
869
870    public static void main() {
871        DataMiner.main();
872    }
873
874    public static void main() {
875        DataMiner.main();
876    }
877
878    public static void main() {
879        DataMiner.main();
880    }
881
882    public static void main() {
883        DataMiner.main();
884    }
885
886    public static void main() {
887        DataMiner.main();
888    }
889
890    public static void main() {
891        DataMiner.main();
892    }
893
894    public static void main() {
895        DataMiner.main();
896    }
897
898    public static void main() {
899        DataMiner.main();
900    }
901
902    public static void main() {
903        DataMiner.main();
904    }
905
906    public static void main() {
907        DataMiner.main();
908    }
909
910    public static void main() {
911        DataMiner.main();
912    }
913
914    public static void main() {
915        DataMiner.main();
916    }
917
918    public static void main() {
919        DataMiner.main();
920    }
921
922    public static void main() {
923        DataMiner.main();
924    }
925
926    public static void main() {
927        DataMiner.main();
928    }
929
930    public static void main() {
931        DataMiner.main();
932    }
933
934    public static void main() {
935        DataMiner.main();
936    }
937
938    public static void main() {
939        DataMiner.main();
940    }
941
942    public static void main() {
943        DataMiner.main();
944    }
945
946    public static void main() {
947        DataMiner.main();
948    }
949
950    public static void main() {
951        DataMiner.main();
952    }
953
954    public static void main() {
955        DataMiner.main();
956    }
957
958    public static void main() {
959        DataMiner.main();
960    }
961
962    public static void main() {
963        DataMiner.main();
964    }
965
966    public static void main() {
967        DataMiner.main();
968    }
969
970    public static void main() {
971        DataMiner.main();
972    }
973
974    public static void main() {
975        DataMiner.main();
976    }
977
978    public static void main() {
979        DataMiner.main();
980    }
981
982    public static void main() {
983        DataMiner.main();
984    }
985
986    public static void main() {
987        DataMiner.main();
988    }
989
990    public static void main() {
991        DataMiner.main();
992    }
993
994    public static void main() {
995        DataMiner.main();
996    }
997
998    public static void main() {
999        DataMiner.main();
1000    }
1001
1002    public static void main() {
1003        DataMiner.main();
1004    }
1005
1006    public static void main() {
1007        DataMiner.main();
1008    }
1009
1010    public static void main() {
1011        DataMiner.main();
1012    }
1013
1014    public static void main() {
1015        DataMiner.main();
1016    }
1017
1018    public static void main() {
1019        DataMiner.main();
1020    }
1021
1022    public static void main() {
1023        DataMiner.main();
1024    }
1025
1026    public static void main() {
1027        DataMiner.main();
1028    }
1029
1030    public static void main() {
1031        DataMiner.main();
1032    }
1033
1034    public static void main() {
1035        DataMiner.main();
1036    }
1037
1038    public static void main() {
1039        DataMiner.main();
1040    }
1041
1042    public static void main() {
1043        DataMiner.main();
1044    }
1045
1046    public static void main() {
1047        DataMiner.main();
1048    }
1049
1050    public static void main() {
1051        DataMiner.main();
1052    }
1053
1054    public static void main() {
1055        DataMiner.main();
1056    }
1057
1058    public static void main() {
1059        DataMiner.main();
1060    }
1061
1062    public static void main() {
1063        DataMiner.main();
1064    }
1065
1066    public static void main() {
1067        DataMiner.main();
1068    }
1069
1070    public static void main() {
1071        DataMiner.main();
1072    }
1073
1074    public static void main() {
1075        DataMiner.main();
1076    }
1077
1078    public static void main() {
1079        DataMiner.main();
1080    }
1081
1082    public static void main() {
1083        DataMiner.main();
1084    }
1085
1086    public static void main() {
1087        DataMiner.main();
1088    }
1089
1090    public static void main() {
1091        DataMiner.main();
1092    }
1093
1094    public static void main() {
1095        DataMiner.main();
1096    }
1097
1098    public static void main() {
1099        DataMiner.main();
1100    }
1101
1102    public static void main() {
1103        DataMiner.main();
1104    }
1105
1106    public static void main() {
1107        DataMiner.main();
1108    }
1109
1110    public static void main() {
1111        DataMiner.main();
1112    }
1113
1114    public static void main() {
1115        DataMiner.main();
1116    }
1117
1118    public static void main() {
1119        DataMiner.main();
1120    }
1121
1122    public static void main() {
1123        DataMiner.main();
1124    }
1125
1126    public static void main() {
1127        DataMiner.main();
1128    }
1129
1130    public static void main() {
1131        DataMiner.main();
1132    }
1133
1134    public static void main() {
1135        DataMiner.main();
1136    }
1137
1138    public static void main() {
1139        DataMiner.main();
1140    }
1141
1142    public static void main() {
1143        DataMiner.main();
1144    }
1145
1146    public static void main() {
1147        DataMiner.main();
1148    }
1149
1150    public static void main() {
1151        DataMiner.main();
1152    }
1153
1154    public static void main() {
1155        DataMiner.main();
1156    }
1157
1158    public static void main() {
1159        DataMiner.main();
1160    }
1161
1162    public static void main() {
1163        DataMiner.main();
1164    }
1165
1166    public static void main() {
1167        DataMiner.main();
1168    }
1169
1170    public static void main() {
1171        DataMiner.main();
1172    }
1173
1174    public static void main() {
1175        DataMiner.main();
1176    }
1177
1178    public static void main() {
1179        DataMiner.main();
1180    }
1181
1182    public static void main() {
1183        DataMiner.main();
1184    }
1185
1186    public static void main() {
1187        DataMiner.main();
1188    }
1189
1190    public static void main() {
1191        DataMiner.main();
1192    }
1193
1194    public static void main() {
1195        DataMiner.main();
1196    }
1197
1198    public static void main() {
1199        DataMiner.main();
1200    }
1201
1202    public static void main() {
1203        DataMiner.main();
1204    }
1205
1206    public static void main() {
1207        DataMiner.main();
1208    }
1209
1210    public static void main() {
1211        DataMiner.main();
1212    }
1213
1214    public static void main() {
1215        DataMiner.main();
1216    }
1217
1218    public static void main() {
1219        DataMiner.main();
1220    }
1221
1222    public static void main() {
1223        DataMiner.main();
1224    }
1225
1226    public static void main() {
1227        DataMiner.main();
1228    }
1229
1230    public static void main() {
1231        DataMiner.main();
1232    }
1233
1234    public static void main() {
1235        DataMiner.main();
1236    }
1237
1238    public static void main() {
1239        DataMiner.main();
1240    }
1241
1242    public static void main() {
1243        DataMiner.main();
1244    }
1245
1246    public static void main() {
1247        DataMiner.main();
1248    }
1249
1250    public static void main() {
1251        DataMiner.main();
1252    }
1253
1254    public static void main() {
1255        DataMiner.main();
1256    }
1257
1258    public static void main() {
1259        DataMiner.main();
1260    }
1261
1262    public static void main() {
1263        DataMiner.main();
1264    }
1265
1266    public static void main() {
1267        DataMiner.main();
1268    }
1269
1270    public static void main() {
1271        DataMiner.main();
1272    }
1273
1274    public static void main() {
1275        DataMiner.main();
1276    }
1277
1278    public static void main() {
1279        DataMiner.main();
1280    }
1281
1282    public static void main() {
1283        DataMiner.main();
1284    }
1285
1286    public static void main() {
1287        DataMiner.main();
1288    }
1289
1290    public static void main() {
1291        DataMiner.main();
1292    }
1293
1294    public static void main() {
1295        DataMiner.main();
1296    }
1297
1298    public static void main() {
1299        DataMiner.main();
1300    }
1301
1302    public static void main() {
1303        DataMiner.main();
1304    }
1305
1306    public static void main() {
1307        DataMiner.main();
1308    }
1309
1310    public static void main() {
1311        DataMiner.main();
1312    }
1313
1314    public static void main() {
1315        DataMiner.main();
1316    }
1317
1318    public static void main() {
1319        DataMiner.main();
1320    }
1321
1322    public static void main() {
1323        DataMiner.main();
1324    }
1325
1326    public static void main() {
1327        DataMiner.main();
1328    }
1329
1330    public static void main() {
1331        DataMiner.main();
1332    }
1333
1334    public static void main() {
1335        DataMiner.main();
1336    }
1337
1338    public static void main() {
1339        DataMiner.main();
1340    }
1341
1342    public static void main() {
1343        DataMiner.main();
1344    }
1345
1346    public static void main() {
1347        DataMiner.main();
1348    }
1349
1350    public static void main() {
1351        DataMiner.main();
1352    }
1353
1354    public static void main() {
1355        DataMiner.main();
1356    }
1357
1358    public static void main() {
1359        DataMiner.main();
1360    }
1361
1362    public static void main() {
1363        DataMiner.main();
1364    }
1365
1366    public static void main() {
1367        DataMiner.main();
1368    }
1369
1370    public static void main() {
1371        DataMiner.main();
1372    }
1373
1374    public static void main() {
1375        DataMiner.main();
1376    }
1377
1378    public static void main() {
1379        DataMiner.main();
1380    }
1381
1382    public static void main() {
1383        DataMiner.main();
1384    }
1385
1386    public static void main() {
1387        DataMiner.main();
1388    }
1389
1390    public static void main() {
1391        DataMiner.main();
1392    }
1393
1394    public static void main() {
1395        DataMiner.main();
1396    }
1397
1398    public static void main() {
1399        DataMiner.main();
1400    }
1401
1402    public static void main() {
1403        DataMiner.main();
1404    }
1405
1406    public static void main() {
1407        DataMiner.main();
1408    }
1409
1410    public static void main() {
1411        DataMiner.main();
1412    }
1413
1414    public static void main() {
1415        DataMiner.main();
1416    }
1417
1418    public static void main() {
1419        DataMiner.main();
1420    }
1421
1422    public static void main() {
1423        DataMiner.main();
1424    }
1425
1426    public static void main() {
1427        DataMiner.main();
1428    }
1429
1430    public static void main() {
1431        DataMiner.main();
1432    }
1433
1434    public static void main() {
1435        DataMiner.main();
1436    }
1437
1438    public static void main() {
1439        DataMiner.main();
1440    }
1441
1442    public static void main() {
1443        DataMiner.main();
1444    }
1445
1446    public static void main() {
1447        DataMiner.main();
1448    }
1449
1450    public static void main() {
1451        DataMiner.main();
1452    }
1453
1454    public static void main() {
1455        DataMiner.main();
1456    }
1457
1458    public static void main() {
1459        DataMiner.main();
1460    }
1461
1462    public static void main() {
1463        DataMiner.main();
1464    }
1465
1466    public static void main() {
1467        DataMiner.main();
1468    }
1469
1470    public static void main() {
1471        DataMiner.main();
1472    }
1473
1474    public static void main() {
1475        DataMiner.main();
1476    }
1477
1478    public static void main() {
1479        DataMiner.main();
1480    }
1481
1482    public static void main() {
1483        DataMiner.main();
1484    }
1485
1486    public static void main() {
1487        DataMiner.main();
1488    }
1489
1490    public static void main() {
1491        DataMiner.main();
1492    }
1493
1494    public static void main() {
1495        DataMiner.main();
1496    }
1497
1498    public static void main() {
1499        DataMiner.main();
1500    }
1501
1502    public static void main() {
1503        DataMiner.main();
1504    }
1505
1506    public static void main() {
1507        DataMiner.main();
1508    }
1509
1510    public static void main() {
1511        DataMiner.main();
1512    }
1513
1514    public static void main() {
151
```

```

C:\Windows\system32\cmd.exe
C:\Users\User\Desktop>cd "C:\Users\User\Documents\NetBeansProjects\MarketBasket2"
C:\Users\User\Documents\NetBeansProjects\MarketBasket2>javac src\marketBasket2/*
.java -d .
C:\Users\User\Documents\NetBeansProjects\MarketBasket2>java marketBasket2.DataMiner
Enter disease name:
cholera
Enter minimum support count:
0.8
[AIK, FAI, FAK, FIK, AHK, HIK, AHI, FHK]
AIK: 1.0
FAI: 1.0
FAK: 1.0
FIK: 1.0
AHK: 1.0
HIK: 1.0
AHI: 1.0
FHK: 1.0

```

Original Format

One hundred cases of patients suffering from the selected diseases are gotten with the different symptoms which each of them is suffering from.

NOTE: The digits in the symptoms column each represents a particular symptoms which the patients is suffering from.

Discussion

The above tables show the different results given by the system after which the irrelevant rules are filtered out. After running the program on the disease table with the support count given that is 0.8 which is equivalent to 80% the confidence level is set to be able to ascertain the rules which meet up with the set count. The rules are then validated with the medical personnel in order to establish the relevant ones.

Most of the rules in the result indicated a justify correspondence between a set of symptoms and disease diagnosis. According to doctor's prescription, the above rules are 80% acceptable in the healthcare system and can assist the medical personnel in diagnosis. It will also reduce stress on medical personnel and also assist patients to get adequate attention and care.

The results obtained also shows the list of the unaccepted rules. However, the doctors contacted states that this rules cannot be accepted because some important symptoms are missing which can be as a result of the data collected. However, it is possible for two different diseases to have more than one similar symptom, but their degree of intensity may differ for example in the case of malaria and typhoid fever, headache is a major symptom, it occurs once in a while and mostly slightly in malaria. But in the case of typhoid, the intensity of headache is severe. In order to ascertain these two diseases; each symptom should be assigned a weighing factor which has value between zero and one. It is also advice that in the case of these two diseases; blood test can be carried out to have adequate result.

For all its peril, medical data mining can be the most rewarding. This is due to the fact that Thousands of data elements can be gathered in order to find solutions to some scientific questions. In the process of finding an answer to a medical problem, a life can be extended and comfort can be given to an ill person.

Summary

Owning to the danger surrounding misdiagnosis in Healthcare System, as medical diagnosis is essential and also delicate task that if not carried out effectively and carefully can lead to death. Therefore a system needs to be developed to safeguard against the inaccurate prescription given to patients which sometimes leads to misdiagnosis.

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. There is a lot of hidden data available within the healthcare system which makes the health care environment information rich but knowledge poor. There are many algorithms available for generating association rules over time, and one of the techniques of association rule mining is Market- basket analysis which is meant for identifying association between items in a transactional database.

The research work has study the manual operation of diagnosis and treatments of some selected diseases affecting mankind. It explained the general diagnosis procedures and techniques involves.

In this research, the analysis technique (Market- basket analysis) is applied on medical data (i.e. Symptoms and diseases) in other to detect the association between a group of symptoms and a disease. Each patient's data is referred to as a transaction. The rules generated in the research provide a method of measuring joint frequencies for different symptoms and the corresponding diseases.

Recommendations

The result generated in this research warrant further application in the Health System and also a data mining technique for medical informatics can be developed. Further research may be carried out on how to use some knowledge of the underlying domain to decide the types of association rules which benefit the medical personnel. A research can also be carried out on a set of restricted data which includes a specific group of diseases (Such as Cancer) and their different diagnosis procedures and treatments using data mining technique. This type of study can provide useful information concerning the effectiveness of a group of method in diagnosing a particular disease in the data and a set of treatments which is very accurate in curing the diseases.

This market basket technique can also be applied in detecting fraudulent behavior among physicians and patients, such as Association of medical personnel with frequent prescription of expensive and un-intruding diagnosis of patients. This study serves as advancement toward the technological development of our medical services.

Conclusion

Since the ability of computer system is known; one can conclude on the best method which favors the diagnosis of diseases in healthcare. The system will support and supplement human judgment, that is the physician, by reproducing the logical usage of data to the skilled observed. And also make diagnosis very fast and accurate. Doctors and Nurses should be encouraged to learn how to use computer so as to ease the work of system developer.

REFERENCES

1. Boris Milovic and Milan Milovic. (2012). Prediction and Decision Making in Health Care Using Data Mining. Kuwait Chapter of Arabian Journal of Business and Management Review, 1(12), 126-136.
2. Fawzi Elias Bekri, Govardhan A. (2011). Association of Data Mining and Healthcare Domain: Issues and Current State of the Art. Global Journal of Computer Science and Technology, 11(21), 1- 9.

3. Jackson, J. (2002). Data Mining: A Conceptual Overview. *Communications of the Association for Information Systems*, 8, 267-296.
4. Kenneth H. Falchuk, Evan Falchuk (2013): The Misdiagnosis epidemic; Five root causes and the growing demand for more patient- centric care. *International Journal for Healthcare Management*. Volume 5, Issue 1, Pages 61-65.
5. Mohammed Al-Maolegi and Bassam Arkok. (2014). An Improved Apriori Algorithm for Association Rules. *International Journal on ANtural language Computing*, 3(1), 21-29.
6. Rani, K. U. (2011). Analysis Of Heart Diseases Dataset Using Neural Network Approach. *International Journal of Data Mining and Knowledge Management Process*, 1-8.
7. Sona Baby and Ariya T.K. (2014). A survey paper of data mining in medical diagnosis. *International Journal of Research in Computer and Communication Technology*, 98- 101.
8. Srinivas, K., Rao, G. R., & Govardhan, A. (2012). Mining Association rules from Large Datasets Towards Disease Prediction. *2012 International Conference on Information and Computer Networks*, 27, 22- 26.