CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Data Mining is the process of discovering actionable information from large sets of data. Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), a field at the intersection of computer science and statistics, is the process that attempts to discover patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use Aside from the raw analysis step, it involves database and data management aspects, data preprocessing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

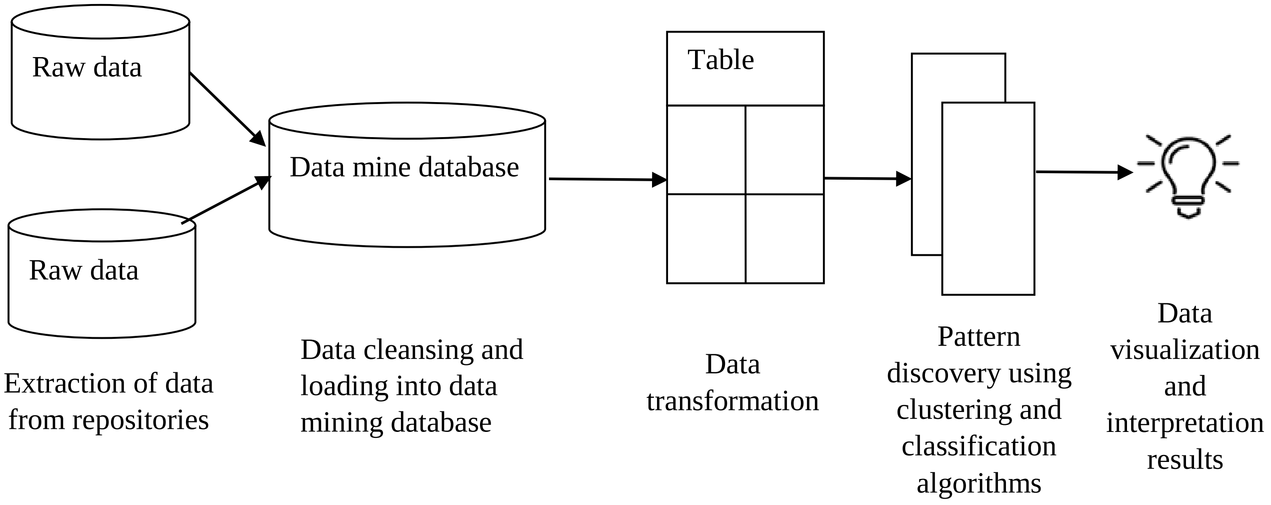

Raw data
Raw data
Extraction of data from repositories
Data mine database
Data cleansing and loading into data mining database
Table
Data transformation
Pattern discovery using clustering and classification algorithms
Data visualization and interpretation results

Fig 1.1: Process of Data Mining

1.1.1 Data

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- Operational or transactional data such as, sales, cost, inventory, payroll, and accounting
- Nonoperational data, such as industry sales, forecast data, and macro-economic data
- Meta data - data about the data itself, such as logical database design or data dictionary definitions.

1.1.2 Data Mining Elements

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.

- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table.

1.2 DATA MINING TECHNIQUES

1.2.1 Association

Association is one of the best known data mining technique. In association, a pattern is discovered based on a relationship of a particular item on other items in the same transaction. For example, the association technique is used in market basket analysis to identify what products that customers frequently purchase together. Based on this data businesses can have corresponding marketing campaign to sell more products to make more profit.

1.2.1.1 Apriori Algorithm

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

1.2.2 Classification

Classification is a classic data mining technique based on machine learning. Basically classification is used to classify each item in a set of data into one of predefined set of classes or groups. Classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics. In classification, make the software that can learn how to classify the data items into groups. For example, can apply classification in application that “given all past records of employees who left the company, predict which current employees are probably to leave in the future.” In this case, divide the employee’s records into two groups that are “leave” and “stay”.

1.2.2.1 Decision trees

Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.

1.2.2.2 Nearest neighbor method

A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k 1). Sometimes called the k-nearest neighbor technique.

1.2.2.3 Rule induction

The extraction of useful if-then rules from data based on statistical significance.

1.2.2.4 Data visualization

The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

1.2.3 Clustering

Clustering is a data mining technique that makes meaningful or useful cluster of objects that have similar characteristic using automatic technique. Different from classification, clustering technique also defines the classes and put objects in them, while in classification objects are assigned into predefined classes. To make the concept clearer, can take library as an example. In a library, books have a wide range of topics available. The challenge is how to keep those books in a way that readers can take several books in a specific topic without hassle.

There are several major data mining techniques have been developed and used in data mining projects recently including association, classification, clustering, prediction and sequential patterns. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

To accomplish these tasks, data miners use one or more of the following techniques:

- Association rule discovery: defining normal activity and enabling the discovery of anomalies.
- Classification predicting the category to which a particular record belongs.
- Clustering of the data into natural categories.

1.3 NEED FOR DATA MINING

Data mining is the procedure of capturing large sets of data in order to identify the insights and visions of that data. Nowadays, the demand of data industry is rapidly growing which has also increased the demands for Data analysts and Data scientists.

Since with this technique, we analyze the data and then convert that data into meaningful information. This helps the business to take accurate and better decisions in an organization.

Data mining helps to develop smart market decision, run accurate campaigns, predictions are taken and many more.

With the help of Data mining, we can analyze customer behaviors and their insights. This leads to great success and data-driven business.

1.4 DATA MINING AND ITS PROCESS

Data mining is an interactive process. Take a look at the following steps.

1.4.1 Requirement gathering

Data mining project starts with the requirement gathering and understanding. Data mining analysts or users define the requirement scope with the vendor business perspective. Once, the scope is defined we move to the next phase.

1.4.2 Data exploration

Here, in this step Data mining experts gather, evaluate and explore the requirement or project. Experts understand the problems, challenges and convert them to metadata. In this step, data mining statistics are used to identify and convert the data patterns.

1.4.3 Data preparations

Data mining experts convert the data into meaningful information for the modelling step. They use ETL process – extract, transform and load. They are also responsible for creating new data attributes. Here various tools are used to present data in a structural format without changing the meaning of data sets.

1.4.4 Modelling

Data experts put their best tools in place for this step as this plays a vital role in the complete processing of data. All modeling methods are applied to filter the data in an appropriate manner. Modelling and evaluation are correlated steps and are followed same time to check the parameters. Once the final modeling is done the final outcome is quality proven.

1.4.5 Evaluation

This is the filtering process after the successful modelling. If the outcome is not satisfied then it is transferred to the model again. Upon final outcome, the requirement is checked again with the vendor so no point is missed. Data mining experts judge the complete result at the end.

1.4.6 Deployment

This is the final stage of the complete process. Experts present the data to vendors in the form of spreadsheets or graphs.

1.5 ADVANTAGES OF DATA MINING

- With the help of Data mining- Marketing companies build data models and prediction based on historical data. They run campaigns, marketing strategy etc. This leads to success and rapid growth.
- The retail industry is also on the same page with marketing companies- With Data mining they believe in predictive based models for their goods and services. Retail stores can have better production and customer insights. Discounts and redemption are based on historical data.

- Data mining suggest banks regarding their financial benefits and updates. They build a model based on customer data and then check out the loan process which is truly based on data mining. In other ways also Data mining serves a lot to the banking industry.
- Manufacturing obtains benefits from Data mining in engineering data and detecting the faulty devices and products. This helps them to cut off the defected items from the list and then they can occupy the best services and products in place.
- It helps government bodies to analyze the financial data and transaction to model them to useful information.
- Data mining organization can improve planning and decision makings.
- New revenue streams are generated with the help of Data mining which results in organization growth.

CHAPTER 2

LITERATURE SURVEY

2.1 ARCHITECTURAL REPRESENTATION FOR INFERENCE RULES GENERATION FOR ASTROLOGICAL PREDICTIONS USING INDUCTION OF HOROSCOPE CHARTS. [Ref. No: 1]

This paper specifies the Architectural overview for Astrological Inference engine that uses person's Date of Birth using Horoscope Charts. Because of Astrology is having vast scope and so predictions happened in various aspects such as studies, profession, colour complexion, assets and liabilities, marriage and so on. Methodologies such as zeroR, Simple Cart, Decision Table, Case Based Reasoning were used to evaluate the accuracy. Machine learning algorithms aims at increasing automation, similarly astrological concepts automation using user's recommendations rather than manual computation improves the accuracy and efficiency as per the inference rules. Machine Learning algorithms such as Support Vector machines, Linear Regression, Logistic Regression and so on are useful to compute the Astrological predictions. This paper represents a sample architectural perspective for astrological predictions. This paper represents a sample architectural perspective for astrological predictions.

2.2 ASTROLOGICAL PREDICTION FOR PROFESSION USING CLASSIFICATION TECHNIQUES OF ARTIFICIAL INTELLIGENCE. [Ref. No: 2]

This paper concentrates on finding universal rules and validity of astrology using various scientific methods. This paper have performed prediction for profession of the person by using different classification techniques. This paper uses three methods for testing such as ZeroR, Simple Cart algorithm and Decision table algorithm for the performance of astrological prediction for prediction power, it determines a baseline performance. Simple Cart Algorithm constructs binary tree, hence each internal node has exactly two outgoing edges. Decision table algorithm is used as classifier based on the concept of simple lookup table. Weka (Waikato Environment for Knowledge Analysis) well known software of machine learning written in Java is used in the paper. This paper concluded that decision table algorithm with 12 fold cross validation produces the results with 50% accuracy.

2.3 ASSOCIATION RULE MINING USING APRIORI ALGORITHM: A SURVEY. [Ref. No: 3]

This paper helps to find the association relationship among the large number of database items and its most typical application is to find the new useful rules. There are two phases in association rule mining. They are finding all frequent item sets and generating strong association rules from the frequent item sets. The key idea of Apriori algorithm is to make multiple passes over the database. This algorithm works on the following technique, firstly, separate every acquired data according to discretization of data items and count the data while scan the database, secondly, prune the acquired item sets.

2.4 PREDICTIVE ROLE OF CASE BASED REASONING FOR ASTROLOGICAL PREDICTIONS ABOUT PROFESSION: SYSTEM MODELING APPROACH. [Ref. No: 4]

This paper describes the research in the field of astrology using case based reasoning method of artificial intelligence. This research addresses the prediction based on finding the similarity relationship between data to be predicted and the data stored in the case base. This paper uses reasoning method for prediction because Case based reasoning method is based on the concept that the similar problem can be solved best using the similar solution hence the system learns from the planetary position of the previous cases stored in case base to identify the profession of the newly entered records.

In this system the user provides the information of the new case through Input Interface for the Inference Engine. The implementation was done successfully and the results obtained was only 70-80% accurate because the carrier of the person also depends on his knowledge level, family background and other factor.

2.5 ARCHETYPE OF ASTROLOGICAL PREDICTION SYSTEM ABOUT PROFESSION OF ANY PERSONS' USING CASE BASED REASONING. [Ref. No: 5]

This paper made an effort to develop a formal method of astrological prediction and birth chart analysis, using popular Artificial Intelligence Technique. Case Based Reasoning is used here to predict the profession of any person. This project is started by collecting the data of persons from different profession. The basic information collected was Time of Birth, Place of Birth, Date of Birth, Qualification, Specialization, Profession, type of sector, Number of years in the profession and gender. In this system the user provides the information through Input Interface to the Inference Engine. The Inference Engine uses the new case entered, cases stored in Case Based Storage. This paper implemented Case Based Reasoning computational technique for astrological prediction about profession of individual. The prediction of the profession is done successfully and the prediction was 70-80% accurate.

2.6 THE ROLE OF APRIORI ALGORITHM FOR FINDING THE ASSOCIATION DATA MINING. [Ref. No: 6]

This paper illustrates Apriori algorithm on simulated database and finds the association rules on different confidence value. Association rule mining is interested in finding frequent rules that define relations between unelated frequent items in databases. It has two main measurements. They are support and confidence values. Association rule mining proceeds on two main steps. The first step is to find all item sets with adequate

supports and the second step is to generate association rules by combining these frequent or large item sets. This paper is an attempt to use data mining as a tool used to
Find the hidden pattern of the frequently used item-sets.

2.7 Algorithm Selection for Classification Problems. [Ref. No: 7]

In this paper, algorithm selection is proposed for classification problems in data mining. This paper uses three types of data characteristics. They are simple, information theoretic, and statistical data characteristics are used. Results are generated using nine different algorithms on thirty eight benchmark dataset. The users are finding difficulty in choosing the models which can solve the problem and combining them if more than one model is required. This problem is solved by creating the meta-learning framework. The paper doesn't recommend a set of classifiers that are the best in general for algorithm selection and be used in all cases. But it guides novice users or researchers having little experience in algorithm selection. The algorithm is directly recommended for his problem.

2.8 A Study of Some Data Mining Classification Techniques. [Ref. No: 8]

This paper represent the study of various data mining classification techniques like Decision Tree, K- Nearest Neighbour, Support Vector Machines, Naive Bayesian Classifiers, and Neural Networks. Classification is the process of finding a model that describes and distinguishes data classes or concepts. Classification methods can handle both numerical and categorical attributes. There are several characteristics of classifiers. They are Data size, Correctness, Strength, Time and Extendibility. There are many classifier model. They are Decision Tree, K-Nearest Neighbour, Support Vector Machine, Naive Bayesian Classifiers and Neural Networks.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

This system consist of horoscope purporting to explain aspects of a person's personality and predict future events in their life based on the positions of the sun, moon, and other celestial objects at the time of their birth. This system offers recommendations such as Business, Health, Travel, Asset, and Auspicious events. Though this system is widely used by many users, this system is failed to predict the correct horoscope and birth chart. So the calculations of Dasaputhi will also be incorrect.

3.1.1 Disadvantages of Existing system

Some software such as epanchang are failed to generate an exact horoscope of a person which leads to wrong predictions.

3.2 PROPOSED SYSTEM

As modern people are computerized much fascinated to be aware of what may happen in the forth coming periods, we propose a system to find universal patterns to predict whether a person will become either a government servant or not. The planets positions of the users are collected and created as a dataset. Then the dataset are associated using Association rule mining. This Association Rule Mining generates several rules and these rules are matched to the predefined rules. The Training dataset are trained using the generated model. Then the test dataset are tested to the trained model. Then the trained model is compared to many algorithms. The applied algorithms are Logistic Regression, Linear regression and Support Vector Machine. It gives the accuracy and these accuracy are compared to get the best result.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

Hardware is the physical components of the computer like microprocessor, hard disks, RAM and motherboard. Hardware devices are the executors of the commands provided by software applications. Computer hardware as the electronic, magnetic, and electric devices that carry out the computing functions.

- Platform: Windows10
- Processor: INTEL Core i3
- RAM Capacity: 8GB
- Hard disk: 40GB

4.2 SOFTWARE REQUIREMENTS

Software includes all the various forms and roles that digitally stored data may have and play in a computer (or similar system), regardless of whether the data is used as code for a CPU, or other interpreter. Software thus encompasses a wide array of products that may be developed using different techniques such as ordinary programming languages, scripting languages and etc.

- Operating System : Windows10
- Tool : WEKA Tool

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

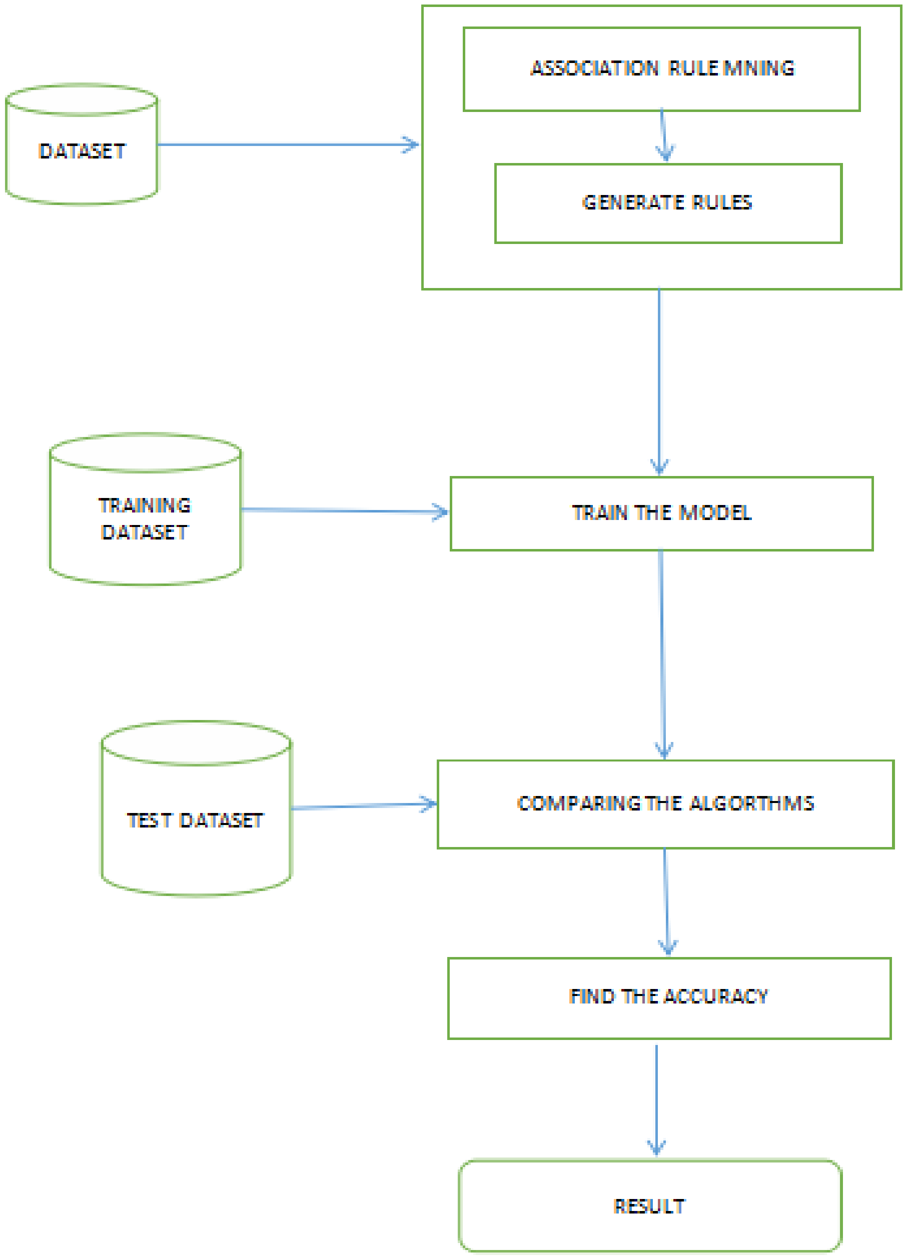


Fig 5.1: System Architecture

5.2 ARCHITECTURE DESCRIPTION

First the planets positions of the users are collected and created as a dataset. Then the dataset are associated using Association rule mining. This Association Rule Mining generates several rules and these rules are matched to the predefined rules. The Training dataset are trained using the generated model. Then the test dataset are tested to the trained model. Then the trained model is compared to many algorithms. The applied algorithms are Logistic Regression, Linear regression and Support Vector Machine. It gives the accuracy and these accuracy are compared to get the best result.

5.3 DATAFLOW DIAGRAMS

A data-flow diagram (DFD) is a way of representing a flow of a data of a processor a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

Level 0

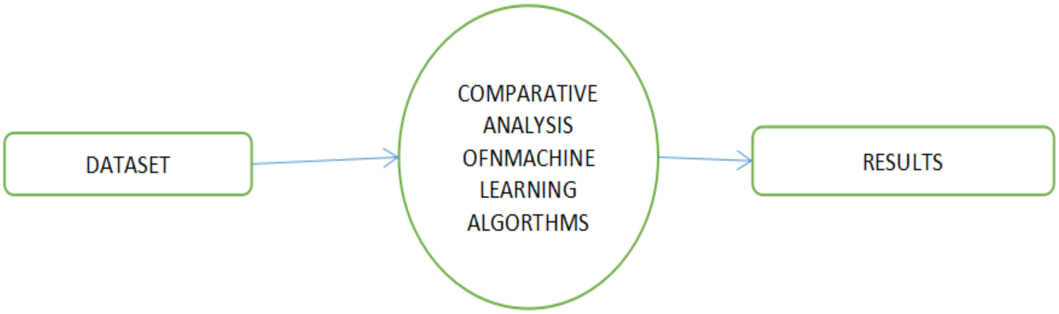


Fig 5.2: Level 0 DFD

The above level 0 diagram shows that the dataset is analyzed and compared using Machine Learning Algorithms and then the result is displayed.

Level 1

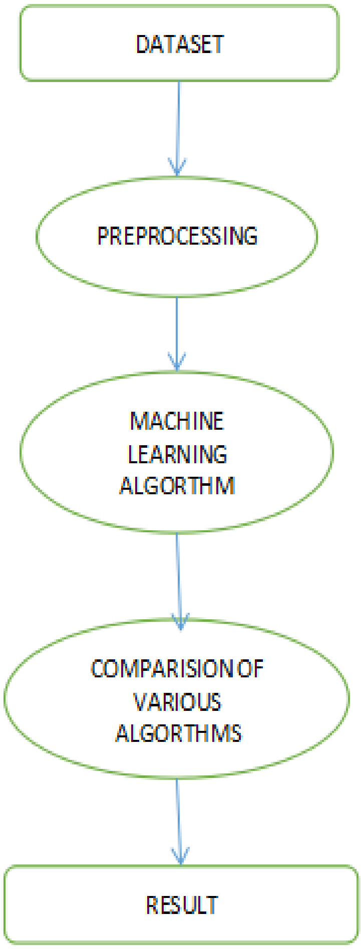


Fig 5.3: Level 1 DFD

This level 1 diagram explains that the dataset is preprocessed. Then the preprocessed dataset is analyzed using various Machine Learning Algorithm. By this analysis, it produces some accuracy for all applied algorithm. Then the accuracy of the algorithm is compared for better result.

LEVEL 2

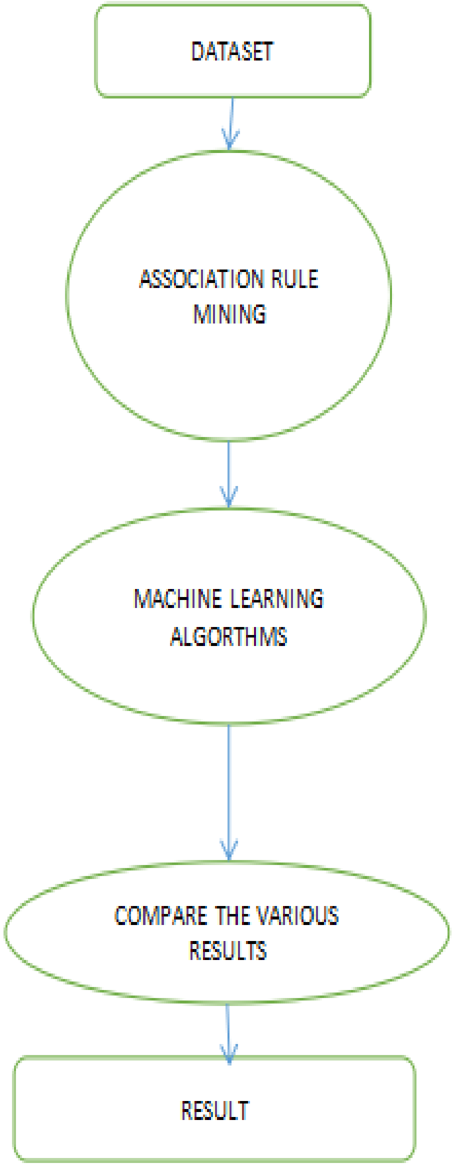
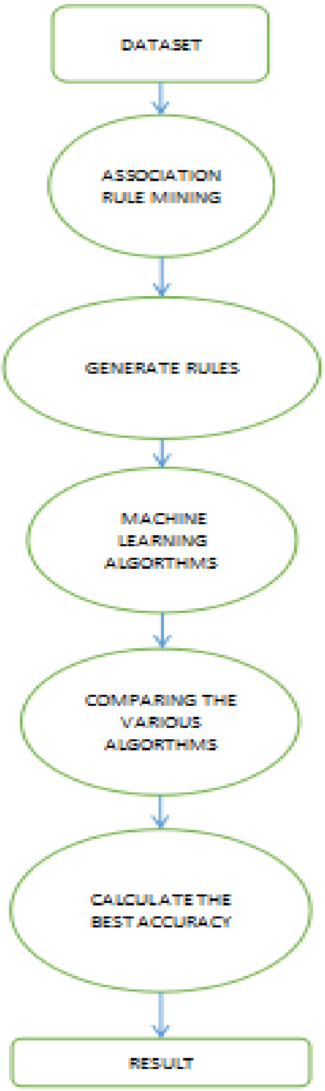


Fig 5.4: Level 2 DFD

The above shown diagram explains that the preprocessed dataset is associated using Apriori Algorithm to find frequent occurrence. Then it generates rules using Association Rule Mining Technique. The generated rules are applied to Machine Learning Algorithms and it gives accuracy of those generated rules. The accuracy of several algorithms are compared to get best results.

LEVEL 3


DATASET
ASSOCIATION RULE MINING
GENERATE RULES
MACHINE LEARNING ALGORTHMS
COMPARING THE VARIOUS ALGORTHMS
CALCULATE THE BEST ACCURACY
RESULT

This level 3 diagram explains that the dataset is associated using Apriori Algorithm and it generates rules using Association Rule Mining Technique. The generated rules are applied to Machine Learning Algorithms and it gives accuracy of those generated rules. The applied algorithm gives accuracy and the accuracy are compared, the higher accuracy is taken as the result.

CHAPTER 6

TECHNIQUES USED

6.1 APRIORI ALGORITHM

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

6.2 LINEAR REGRESSION

Linear Regression is one of the type of regression model which is used to study the relationship between dependent and independent variables. It is used to estimate the relationship between target and one or more predictors.

$$Y = a + bX$$

Where, Y - dependent variable

X - Independent variable

a - intercept of y

b – slope of the line

6.3 SUPPORT VECTOR MACHINE

Support vector machine is a supervised learning model. It creates hyper plane which attempts to maximize the margin between classes. This is achieved by selecting a small number of boundary instance and building a linear function which maximizes separation. Unlike some other linear classifier, svm is able to classify nonlinear class boundaries. SVM has the property of stability and does not change much when a small number of instance are added to the dataset and over fitting is unlikely to occur. Although SVM have many positive theoretical properties, training SVM can often be slow especially in the case of highly dimensional datasets. It analyses the data which is used for classification and regression. It performs well with a limited amount of data. The main purpose of using SVM is to classify the class from the given dataset.

6.4 LOGISTIC REGRESSION

Logistic Regression [4] is a form of parametric regression. It allows predicting a class or membership from a set of variables that may be continuous, discrete, dichotomous or a combination of it. Its details are presented in Agrestic. Logistic regression makes no assumption about the distribution of the independent variables. Logistic regression is used for the prediction of the probability of occurrence of an event by fitting the data to a logit function. The logit function z is defined [6] as a linear combination of regression coefficients bi, input variables Xi and intercept constant b0:

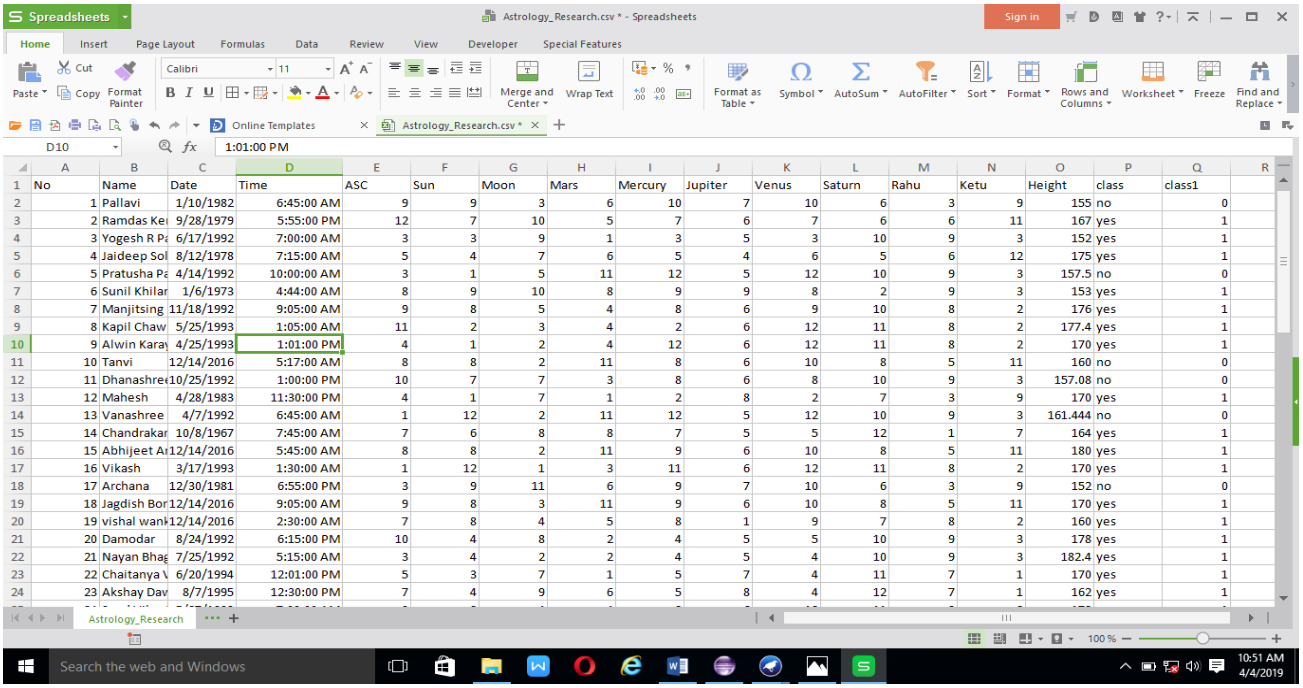
$$z = b0 + b1X1 + b2X2 + ... + bkXk$$

By rewriting the logit function, the probability of an event occurrence is defined as: P (event) =1/1 + e−z

CHAPTER 7

IMPLEMENTATION AND RESULTS

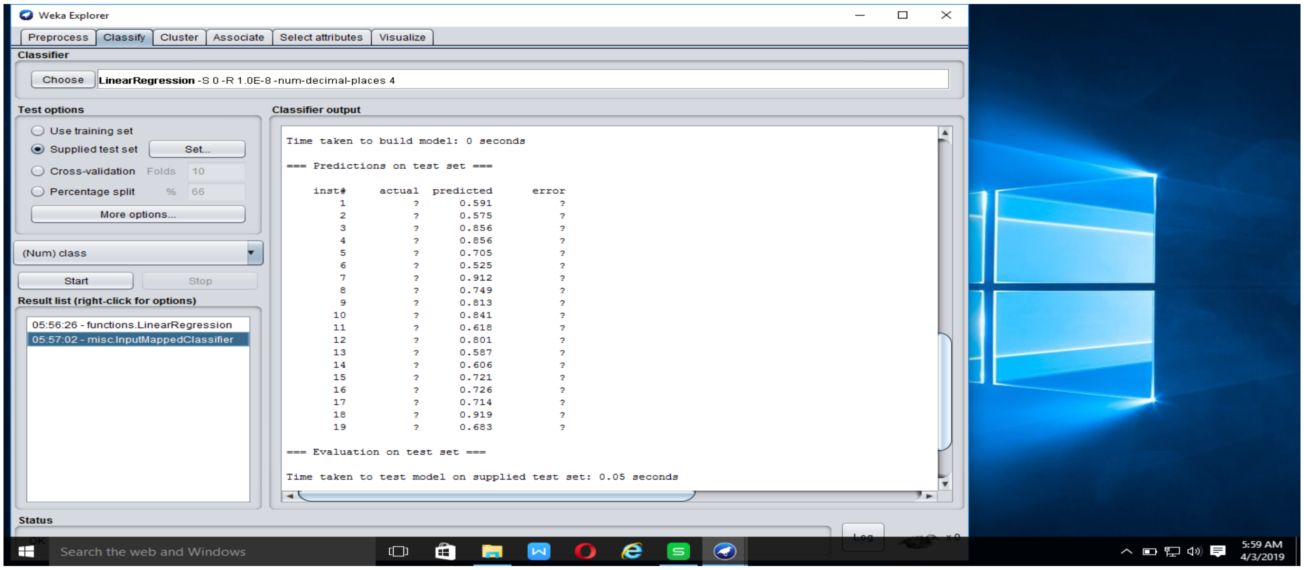
7.1 SCREENSHOTS



D10 | 1:01:00 PM

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	No	Name	Date	Time	ASC	Sun	Moon	Mars	Mercury	Jupiter	Venus	Saturn	Rahu	Ketu	Height	class	class1
2	1	Pallavi	1/10/1982	6:45:00 AM	9	9	3	6	10	7	10	6	3	9	155	no	0
3	2	Ramdas Kel	9/28/1979	5:55:00 PM	12	7	10	5	7	6	7	6	6	11	167	yes	1
4	3	Yogesh R Pa	6/17/1992	7:00:00 AM	3	3	9	1	3	5	3	10	9	3	152	yes	1
5	4	Jaideep Sol	8/12/1978	7:15:00 AM	5	4	7	6	5	4	6	5	6	12	175	yes	1
6	5	Pratusha Pa	4/14/1992	10:00:00 AM	3	1	5	11	12	5	12	10	9	3	157.5	no	0
7	6	Sunil Khilar	1/6/1973	4:44:00 AM	8	9	10	8	9	9	8	2	9	3	153	yes	1
8	7	Manjitsing	11/18/1992	9:05:00 AM	9	8	5	4	8	6	9	10	8	2	176	yes	1
9	8	Kapil Chaw	5/25/1993	1:05:00 AM	11	2	3	4	2	6	12	11	8	2	177.4	yes	1
10	9	Alwin Karay	4/25/1993	1:01:00 PM	4	1	2	4	12	6	12	11	8	2	170	yes	1
11	10	Tanvi	12/14/2016	5:17:00 AM	8	8	2	11	8	6	10	8	5	11	160	no	0
12	11	Dhanashree	10/25/1992	1:00:00 PM	10	7	7	3	8	6	8	10	9	3	157.08	no	0
13	12	Mahesh	4/28/1983	11:30:00 PM	4	1	7	1	2	8	2	7	3	9	170	yes	1
14	13	Vanashree	4/7/1992	6:45:00 AM	1	12	2	11	12	5	12	10	9	3	161.444	no	0
15	14	Chandrakar	10/8/1967	7:45:00 AM	7	6	8	8	7	5	5	12	1	7	164	yes	1
16	15	Abhijeet Ar	12/14/2016	5:45:00 AM	8	8	2	11	9	6	10	8	5	11	180	yes	1
17	16	Vikash	3/17/1993	1:30:00 AM	1	12	1	3	11	6	12	11	8	2	170	yes	1
18	17	Archana	12/30/1981	6:55:00 PM	3	9	11	6	9	7	10	6	3	9	152	no	0
19	18	Jagdish Bor	12/14/2016	9:05:00 AM	9	8	3	11	9	6	10	8	5	11	170	yes	1
20	19	vishal wank	12/14/2016	2:30:00 AM	7	8	4	5	8	1	9	7	8	2	160	yes	1
21	20	Damodar	8/24/1992	6:15:00 PM	10	4	8	2	4	5	5	10	9	3	178	yes	1
22	21	Nayan Bhag	7/25/1992	5:15:00 AM	3	4	2	2	4	5	4	10	9	3	182.4	yes	1
23	22	Chaitanya V	6/20/1994	12:01:00 PM	5	3	7	1	5	7	4	11	7	1	170	yes	1
24	23	Akshay Dav	8/7/1995	12:30:00 PM	7	4	9	6	5	8	4	12	7	1	162	yes	1

Fig 7.1: Dataset



LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

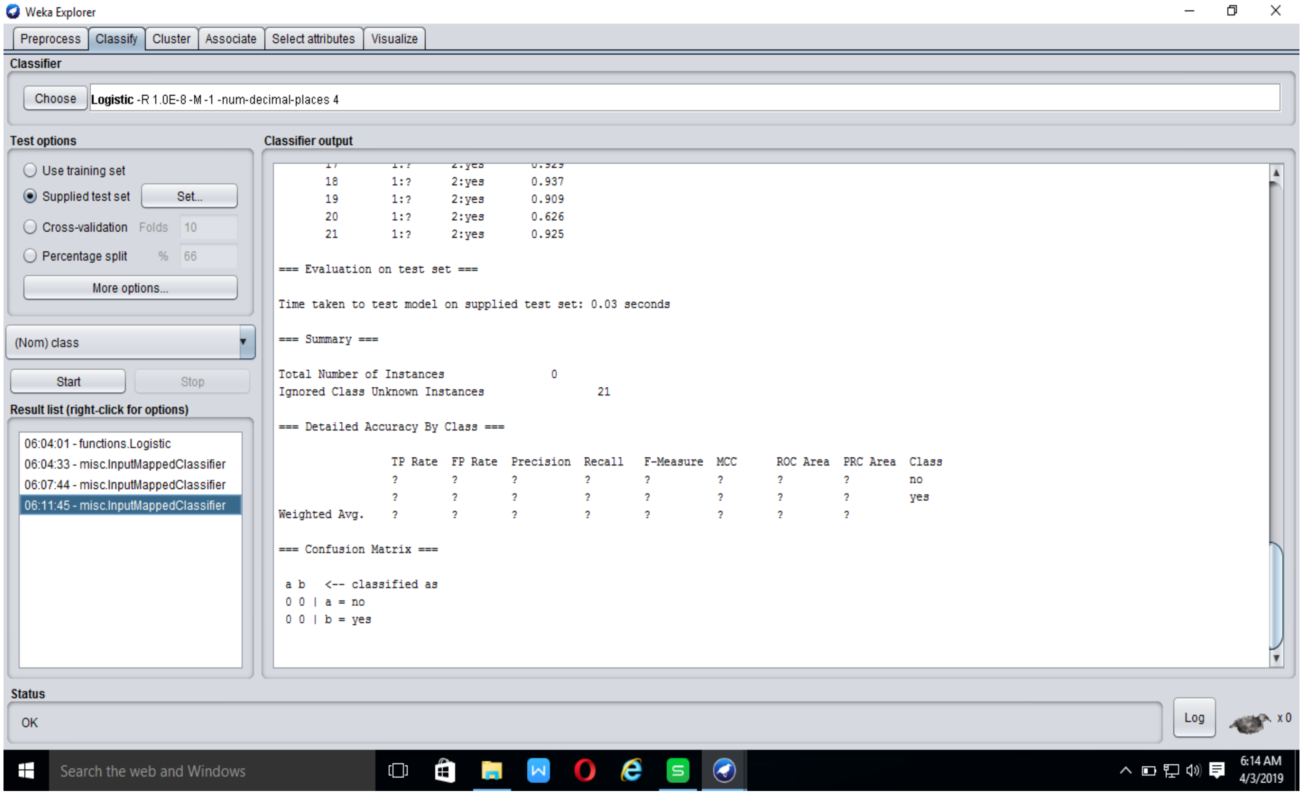
○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Num) class

05:56:26 - functions.LinearRegression
05:57:02 - misc.InputMappedClassifier

```
Time taken to build model: 0 seconds

=== Predictions on test set ===

    inst#     actual  predicted      error
        1          ?      0.591          ?
        2          ?      0.575          ?
        3          ?      0.856          ?
        4          ?      0.856          ?
        5          ?      0.705          ?
        6          ?      0.525          ?
        7          ?      0.912          ?
        8          ?      0.749          ?
        9          ?      0.813          ?
       10          ?      0.841          ?
       11          ?      0.618          ?
       12          ?      0.801          ?
       13          ?      0.587          ?
       14          ?      0.606          ?
       15          ?      0.721          ?
       16          ?      0.726          ?
       17          ?      0.714          ?
       18          ?      0.919          ?
       19          ?      0.683          ?

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.05 seconds
```

Fig 7.2: Linear regression-test1



Classifier: Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options

○ Use training set

● Supplied test set

○ Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

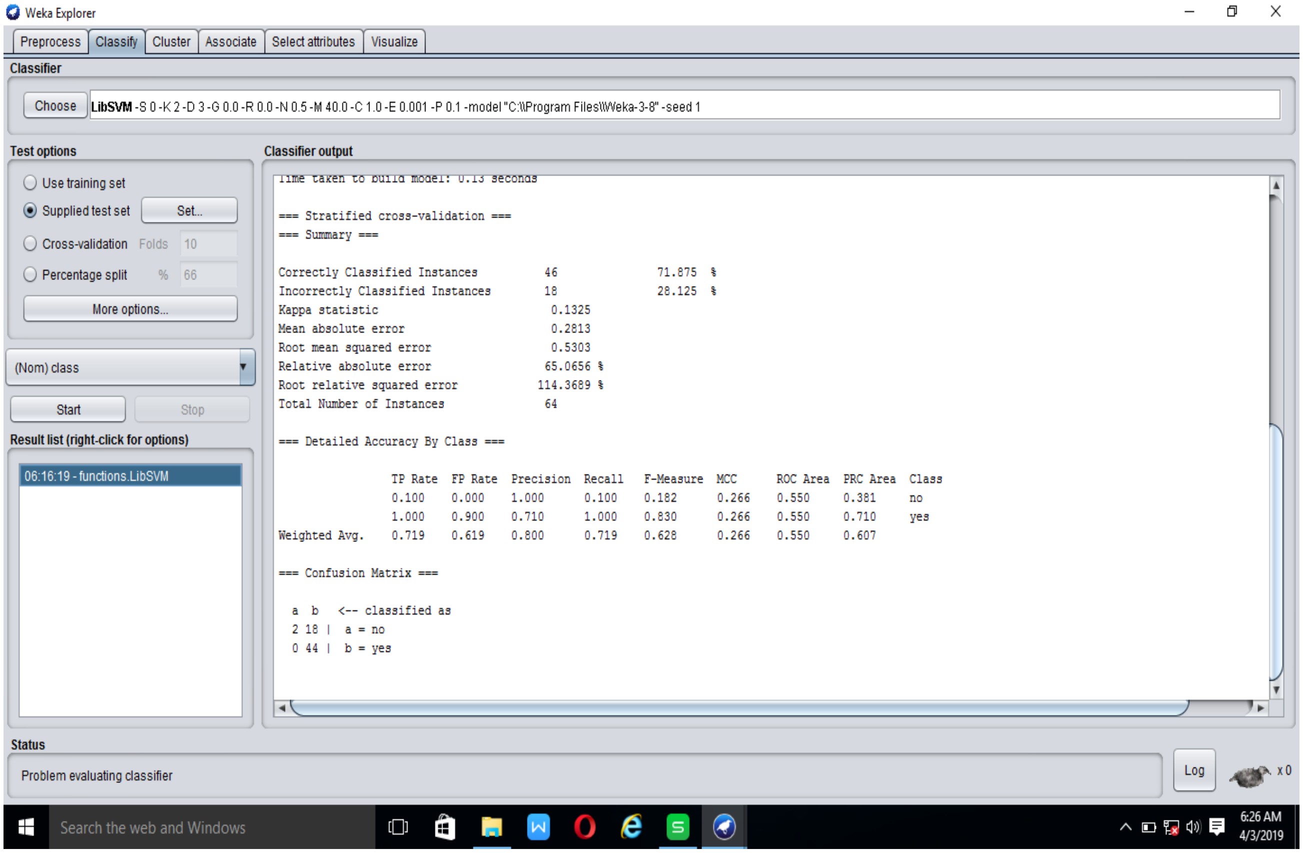
06:04:01 - functions.Logistic
06:04:33 - misc.InputMappedClassifier
06:07:44 - misc.InputMappedClassifier
06:11:45 - misc.InputMappedClassifier

Classifier output

```
      17        1:?      2:yes      0.929
      18        1:?      2:yes      0.937
      19        1:?      2:yes      0.909
      20        1:?      2:yes      0.626
      21        1:?      2:yes      0.925

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.03 seconds

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                21

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 ?        ?        ?          ?        ?          ?        ?         ?         no
                 ?        ?        ?          ?        ?          ?        ?         ?         yes
Weighted Avg.    ?        ?        ?          ?        ?          ?        ?         ?

=== Confusion Matrix ===

 a b   <-- classified as
 0 0 | a = no
 0 0 | b = yes
```

Status: OK

Fig 7.3: Linear Regression-test2



Classifier: LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1

Test options

○ Use training set

● Supplied test set

○ Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

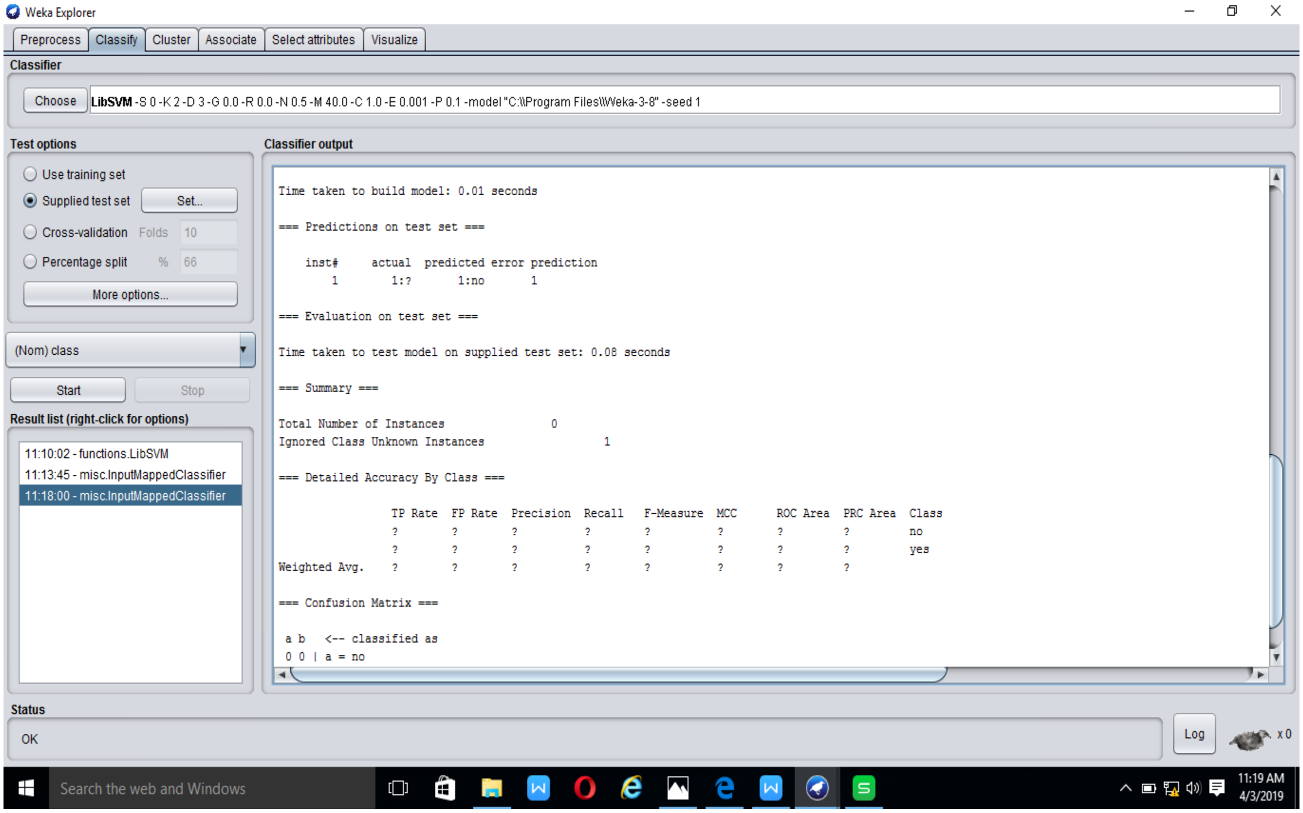
06:16:19 - functions.LibSVM

Classifier output

```
Time taken to build model: 0.13 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          46               71.875  %
Incorrectly Classified Instances        18               28.125  %
Kappa statistic                          0.1325
Mean absolute error                      0.2813
Root mean squared error                  0.5303
Relative absolute error                 65.0656 %
Root relative squared error            114.3689 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.100    0.000    1.000      0.100    0.182      0.266    0.550     0.381     no
                 1.000    0.900    0.710      1.000    0.830      0.266    0.550     0.710     yes
Weighted Avg.    0.719    0.619    0.800      0.719    0.628      0.266    0.550     0.607

=== Confusion Matrix ===

  a  b   <-- classified as
  2 18 |  a = no
  0 44 |  b = yes
```

Status: Problem evaluating classifier

Fig 7.4: LibSVM



Classifier

LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1

Test options

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Result list (right-click for options)

11:10:02 - functions.LibSVM
11:13:45 - misc.InputMappedClassifier
11:18:00 - misc.InputMappedClassifier

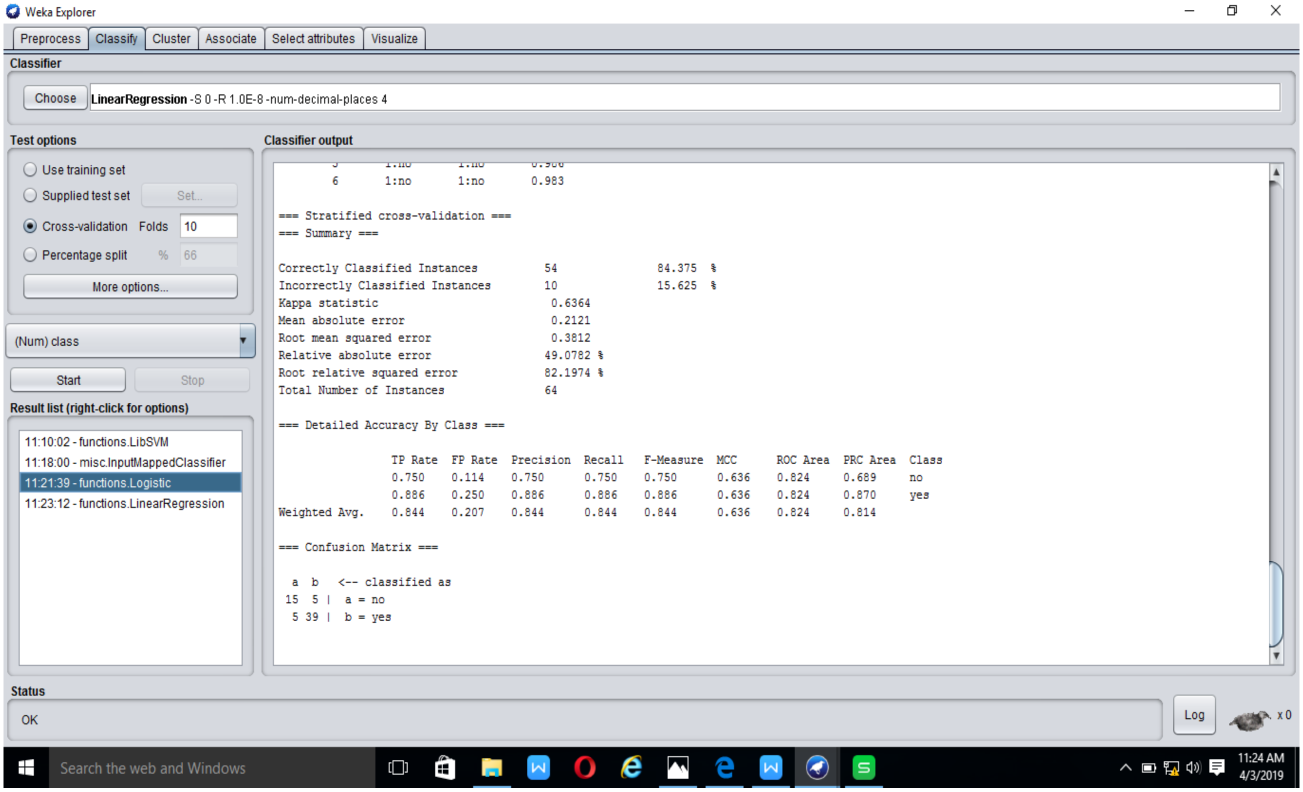
Classifier output

```
Time taken to build model: 0.01 seconds

=== Predictions on test set ===

    inst#     actual  predicted error prediction
        1        1:?       1:no       1

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.08 seconds

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                  1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 ?        ?        ?          ?        ?          ?        ?         ?         no
                 ?        ?        ?          ?        ?          ?        ?         ?         yes
Weighted Avg.    ?        ?        ?          ?        ?          ?        ?         ?

=== Confusion Matrix ===

 a b   <-- classified as
 0 0 | a = no
```

Status

OK

Fig 7.5: LibSVM-test



Classifier

LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Num) class

Result list (right-click for options)

11:10:02 - functions.LibSVM
11:18:00 - misc.InputMappedClassifier
11:21:39 - functions.Logistic
11:23:12 - functions.LinearRegression

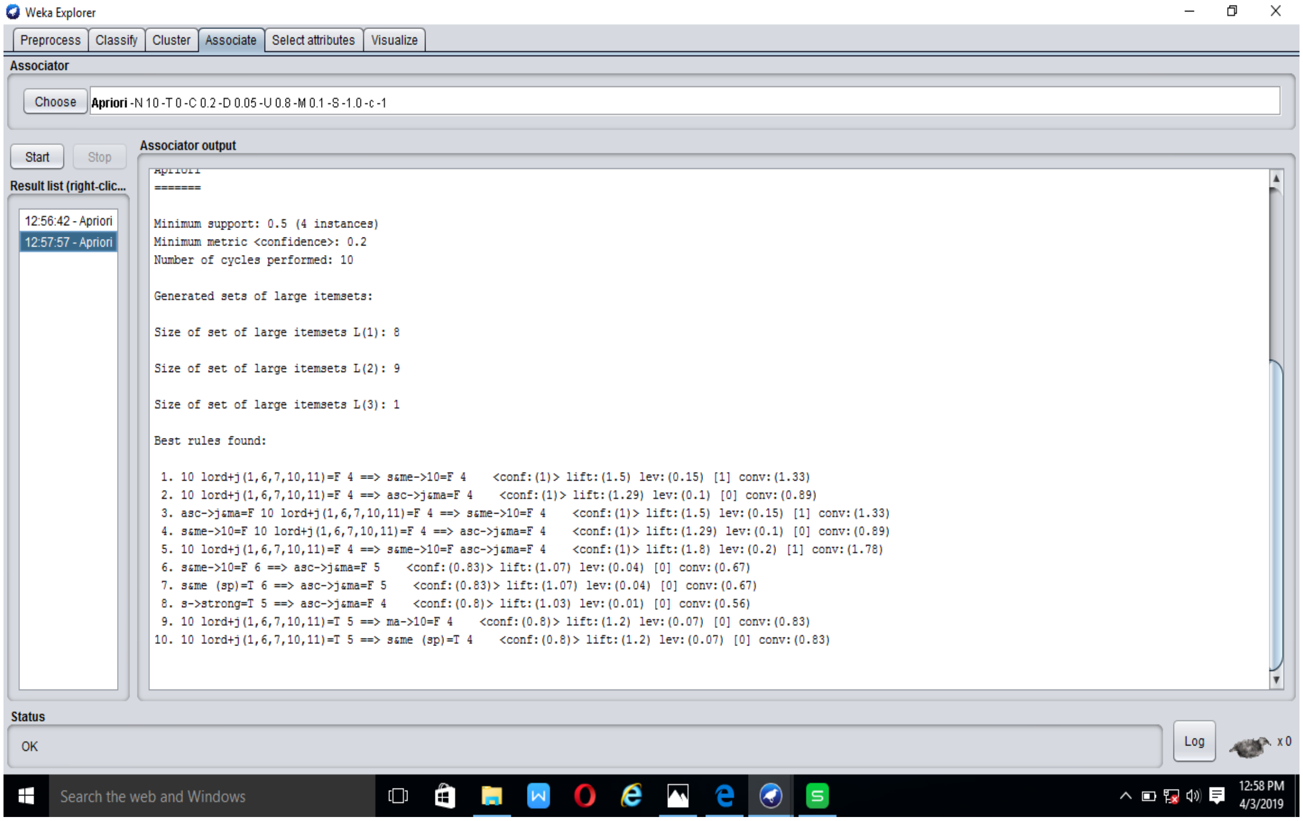
Classifier output

```
        6       1:no       1:no       0.983

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          54               84.375  %
Incorrectly Classified Instances        10               15.625  %
Kappa statistic                          0.6364
Mean absolute error                      0.2121
Root mean squared error                  0.3812
Relative absolute error                 49.0782 %
Root relative squared error             82.1974 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.750    0.114    0.750      0.750    0.750      0.636    0.824     0.689     no
                 0.886    0.250    0.886      0.886    0.886      0.636    0.824     0.870     yes
Weighted Avg.    0.844    0.207    0.844      0.844    0.844      0.636    0.824     0.814

=== Confusion Matrix ===

  a  b   <-- classified as
 15  5 |  a = no
  5 39 |  b = yes
```

Status

OK

Fig 7.6: Logistic Regression



Apriori -N 10 -T 0 -C 0.2 -D 0.05 -U 0.8 -M 0.1 -S -1.0 -c -1

```
Apriori
=======

Minimum support: 0.5 (4 instances)
Minimum metric <confidence>: 0.2
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 9

Size of set of large itemsets L(3): 1

Best rules found:

 1. 10 lord+j(1,6,7,10,11)=F 4 ==> s&me->10=F 4    <conf:(1)> lift:(1.5) lev:(0.15) [1] conv:(1.33)
 2. 10 lord+j(1,6,7,10,11)=F 4 ==> asc->j&ma=F 4    <conf:(1)> lift:(1.29) lev:(0.1) [0] conv:(0.89)
 3. asc->j&ma=F 10 lord+j(1,6,7,10,11)=F 4 ==> s&me->10=F 4    <conf:(1)> lift:(1.5) lev:(0.15) [1] conv:(1.33)
 4. s&me->10=F 10 lord+j(1,6,7,10,11)=F 4 ==> asc->j&ma=F 4    <conf:(1)> lift:(1.29) lev:(0.1) [0] conv:(0.89)
 5. 10 lord+j(1,6,7,10,11)=F 4 ==> s&me->10=F asc->j&ma=F 4    <conf:(1)> lift:(1.8) lev:(0.2) [1] conv:(1.78)
 6. s&me->10=F 6 ==> asc->j&ma=F 5    <conf:(0.83)> lift:(1.07) lev:(0.04) [0] conv:(0.67)
 7. s&me (sp)=T 6 ==> asc->j&ma=F 5    <conf:(0.83)> lift:(1.07) lev:(0.04) [0] conv:(0.67)
 8. s->strong=T 5 ==> asc->j&ma=F 4    <conf:(0.8)> lift:(1.03) lev:(0.01) [0] conv:(0.56)
 9. 10 lord+j(1,6,7,10,11)=T 5 ==> ma->10=F 4    <conf:(0.8)> lift:(1.2) lev:(0.07) [0] conv:(0.83)
10. 10 lord+j(1,6,7,10,11)=T 5 ==> s&me (sp)=T 4    <conf:(0.8)> lift:(1.2) lev:(0.07) [0] conv:(0.83)
```

Fig 7.7: Apriori Algorithm

CHAPTER 8

PERFORMANCE METRICS

Accuracy

Formula

- (TP + TN)/(TP + FP + TN + FN)

Accuracy seems like it could be the best method.

Precision/Recall

These two performance metrics are often use in conjunction.

Precision

Formula:

- TP / (TP + FP)

With precision, we are evaluating our data by its performance of ‘positive’ predictions.

Recall (also called sensitivity)

Formula

- TP / (TP + FN)

With recall, we are evaluating our data by its performance of the ground truths for positive outcomes. Meaning, we are judging how well predicted positive when the result was Positive.

Specificity

Specificity is the opposite of Sensitivity or Recall. Hence, the formula is TN/(TN+FP).

F1 Score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

Formula

- (2 * (Precision * Recall))/(Precision + Recall)

The F1-Score is the weighted average (harmonic mean) of precision and recall.

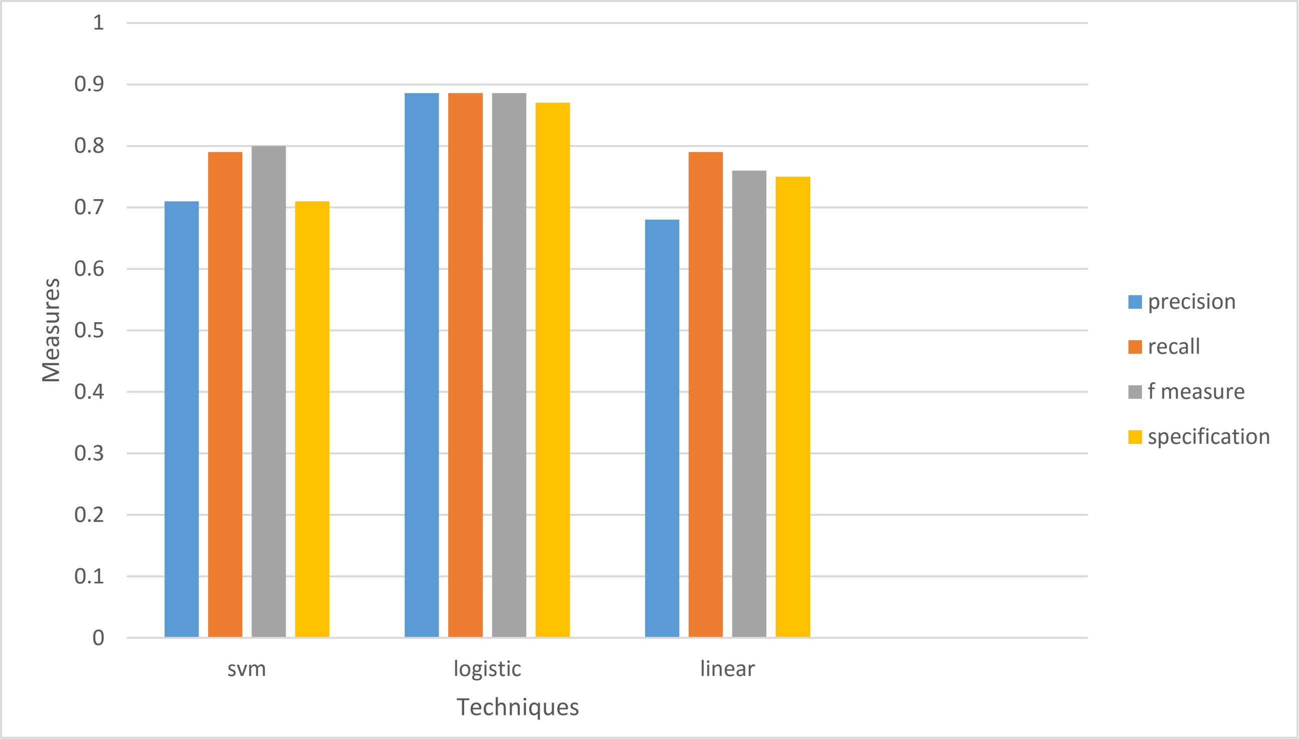


Fig 8.1: Comparison of performance metrics

Fig.8.1 describes the comparison of applied classification techniques with performance metrics. The performance metrics are Precision, Recall, F-Measure and specificity. By this comparison it is observed that Logistic Regression is best with accuracy 84% than Support Vector Machine and Linear Regression.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCENENTS

9.1 CONCLUSION

Astrological prediction is not a theoretical concept but was practical. Some Classification techniques like Support Vector Machine, Logistic Regression and Linear Regression are used for prediction. Hence we used these techniques to predict whether the person will become either a government servant or not. And also we generated the Universal patterns using Apriori algorithm based on the Association between the Planets position, then the patterns are classified by gaining the accuracy 84%.

9.2 FUTURE ENHANCEMENT

The existing Apriori algorithm is not fully capable to analyze the horoscope data in the context of astrology. For that we need to upgrade the Apriori algorithm as a Conditional Apriori Algorithm (CAA).

APPENDIX A

```
package astro;
import java.io.*;
import java.util.*;

public class AlgoAprioriHT extends Observable
 {
public static void main(String[] args) throws Exception
 {
 new AlgoAprioriHT (args);
}
 private List<int[]> itemsets ;
        private String transaFile;
        private int numItems;
       private int numTransactions;
       private double minSup;

        private boolean usedAsLibrary = false;



    public AlgoAprioriHT(String[] args, Observer ob) throws Exception
    {
          usedAsLibrary = true;
          configure(args);
          this.addObserver(ob);
          go();
    }
    public AlgoAprioriHT(String[] args) throws Exception
    {
        configure(args);
        go();
    }

      private void go() throws Exception {

        long start = System.currentTimeMillis();



        createItemsetsOfSize1();
        int itemsetNumber=1; //the current itemset being looked at
        int nbFrequentSets=0;

        while (itemsets.size()>0)
        {

            calculateFrequentItemsets();

            if(itemsets.size()!=0)
            {
                nbFrequentSets+=itemsets.size();
                log("Found "+itemsets.size()+" frequent itemsets of size " + itemsetNumber
+ " (with support "+(minSup*100)+"%)");;
                createNewItemsetsFromPreviousOnes();
      }

      itemsetNumber++;
    }


    long end = System.currentTimeMillis();
    log("Execution time is: "+((double)(end-start)/1000) + " seconds.");
    log("Found "+nbFrequentSets+ " frequents sets for support "+(minSup*100)+"%
(absolute "+Math.round(numTransactions*minSup)+")");
    log("Done");
  }


  private void foundFrequentItemSet(int[] itemset, int support) {
     if (usedAsLibrary) {
      this.setChanged();
      notifyObservers(itemset);
     }
     else {System.out.println(Arrays.toString(itemset) + "  ("+ ((support / (double)
numTransactions))+" "+support+")");}
  }

  private void log(String message) {
     if (!usedAsLibrary) {
          System.err.println(message);
     }
  }
```

```
private void configure(String[] args) throws Exception
{

  if (args.length!=0) transaFile = args[0];
  else transaFile = "E:\\data.csv"; // default


   if (args.length>=2) minSup=(Double.valueOf(args[1]).doubleValue());
   else minSup = .8;// by default
   if (minSup>1 || minSup<0) throw new Exception("minSup: bad value");



   numItems = 0;
   numTransactions=0;
   BufferedReader data_in = new BufferedReader(new FileReader(transaFile));
   while (data_in.ready()) {
        String line=data_in.readLine();
        if (line.matches("\\s*")) continue; // be friendly with empty lines
        numTransactions++;
        StringTokenizer t = new StringTokenizer(line," ");
        while (t.hasMoreTokens()) {
             int x = Integer.parseInt(t.nextToken());
             //log(x);
             if (x+1>numItems) numItems=x+1;
        }
   }
```

```
        outputConfig();
data_in.close();
    }



          private void outputConfig() {

                     log("Input configuration: "+numItems+" items, "+numTransactions+"
transactions, ");
                     log("minsup = "+minSup+"%");
          }

          private void createItemsetsOfSize1() {
                    itemsets = new ArrayList<int[]>();
        for(int i=0; i<numItems; i++)
        {
          int[] cand = {i};
          itemsets.add(cand);
        }
          }



    private void createNewItemsetsFromPreviousOnes()
    {
                    int currentSizeOfItemsets = itemsets.get(0).length;
          log("Creating itemsets of size "+(currentSizeOfItemsets+1)+" based on
"+itemsets.size()+" itemsets of size "+currentSizeOfItemsets);
```

```
  HashMap<String, int[]> tempCandidates = new HashMap<String, int[]>();

        for(int i=0; i<itemsets.size(); i++)
{
    for(int j=i+1; j<itemsets.size(); j++)
    {
        int[] X = itemsets.get(i);
        int[] Y = itemsets.get(j);

        assert (X.length==Y.length);



        int [] newCand = new int[currentSizeOfItemsets+1];
        for(int s=0; s<newCand.length-1; s++) {
            newCand[s] = X[s];
        }

        int ndifferent = 0;

        for(int s1=0; s1<Y.length; s1++)
        {
            boolean found = false;

            for(int s2=0; s2<X.length; s2++) {
            if (X[s2]==Y[s1]) {
                      found = true;
                      break;
            }
```

```
                    }
                    if (!found){ // Y[s1] is not in X
                              ndifferent++;
                                                            newCand[newCand.length -1] = Y[s1];
                    }

                    }

                // we have to find at least 1 different, otherwise it means that we have two
times the same set in the existing candidates
                assert(ndifferent>0);


                if (ndifferent==1) {
                   // HashMap does not have the correct "equals" for int[] :-(
                   // I have to create the hash myself using a String :-(
                    // I use Arrays.toString to reuse equals and hashcode of String
                    Arrays.sort(newCand);
                    tempCandidates.put(Arrays.toString(newCand),newCand);
                }
            }
        }

        //set the new itemsets
        itemsets = new ArrayList<int[]>(tempCandidates.values());
          log("Created "+itemsets.size()+" unique itemsets of size
"+(currentSizeOfItemsets+1));
```

```
    }



    private void line2booleanArray(String line, boolean[] trans) {
              Arrays.fill(trans, false);
              StringTokenizer stFile = new StringTokenizer(line, " "); //read a line from the file to the tokenizer
              //put the contents of that line into the transaction array
              while (stFile.hasMoreTokens())
              {

                  int parsedVal = Integer.parseInt(stFile.nextToken());
                              trans[parsedVal]=true; //if it is not a 0, assign the value to true
              }
    }



    private void calculateFrequentItemsets() throws Exception
    {

        log("Passing through the data to compute the frequency of " + itemsets.size()+ " itemsets of size "+itemsets.get(0).length);

        List<int[]> frequentCandidates = new ArrayList<int[]>(); //the frequent candidates for the current itemset

        boolean match; //whether the transaction has all the items in an itemset
```

```
   int count[] = new int[itemsets.size()]; //the number of successful matches,
initialized by zeros



        // load the transaction file
        BufferedReader data_in = new BufferedReader(new
InputStreamReader(new FileInputStream(transaFile)));

        boolean[] trans = new boolean[numItems];

        // for each transaction
        for (int i = 0; i < numTransactions; i++) {

            // boolean[] trans = extractEncoding1(data_in.readLine());
            String line = data_in.readLine();
            line2booleanArray(line, trans);

            // check each candidate
            for (int c = 0; c < itemsets.size(); c++) {
                match = true; // reset match to false
                // tokenize the candidate so that we know what items need to
be
                // present for a match
                int[] cand = itemsets.get(c);
                //int[] cand = candidatesOptimized[c];
                // check each item in the itemset to see if it is present in the
                // transaction
                for (int xx : cand) {
```

```
                                        if (trans[xx] == false) {
                                                match = false;
                                                break;
                                        }
                                }
                                if (match) { // if at this point it is a match, increase the count
                                        count[c]++;
                                        //log(Arrays.toString(cand)+" is contained in trans
"+i+" ("+line+")");
                                }}
                }

                data_in.close();

                for (int i = 0; i < itemsets.size(); i++) {

                        if ((count[i] / (double) (numTransactions)) >= minSup) {
                                foundFrequentItemSet(itemsets.get(i),count[i]);
                                frequentCandidates.add(itemsets.get(i));
                        }
                        //else log("-- Remove candidate: "+
Arrays.toString(candidates.get(i)) + "  is: "+ ((count[i] / (double) numTransactions)));
                }

        //new candidates are only the frequent candidates
        itemsets = frequentCandidates} }
```

REFERENCES

[1] CNVBR Sri Gowrinath1, B Srinivasa S P Kumar, Architectural Representation for Inference rules generation for Astrological Predictions using induction of Horoscope Charts- ISSN 0973-4562 Volume 13, Number 20 (2018) pp. 14495-14497.

[2] Chaplot, N., Dhyani, P. and Rishi, O.P., 2015, May. Astrological prediction for profession using classification techniques of artificial intelligence. In Computing, Communication & Automation (ICCCA), 2015 International Conference on (pp. 233-236). IEEE.

[3] Charanjeet Kaur, ASSOCIATION RULE MINING USING APRIORI ALGORITHM: A SURVEY- IJRITCC, July 2015, Volume 3 Issue 7, ISSN: 23218169PP : 4431-4436.

[4] O.P. Rishi and NeelamChaplot, "Predictive role of case based reasoning for astrological predictions about profession: System modeling approach" in International Conference on Communication and Computational Intelligence (INCOCCI), 2010, pp 313 – 317.

[5] O.P.Rishi, Neelam Chaplot, Archetype of Astrological Prediction System about Profession of any Persons' using Case Based Reasoning, International Conference on Communication and Computational Intelligence – 2010, pp.373-377.

[6] Jugendra Dongre, Gend Lal Prajapati, S.V. Tokekar, The Role of Apriori Algorithm for Finding the Association Rules in Data Mining- 978-1-4799-2900-9/14/$31.00 ©2014 IEEE.

[7] Nitin Pise, Parag Kulkarni, Algorithm Selection for Classification Problems- 5.3 (2014): 2767-2771.

[8] Vikas K Vijayan, Bindu K.R, Latha Parameswaran, A Comprehensive Study of Text Classification Algorithms- 978-1-5090-6367-3/17/$31.00 ©2017 IEEE.

[9] H. Mahgoub,"Mining association rules from unstructured documents" in Proc. 3rd Int. Conf. on Knowledge Mining, ICKM, Prague, Czech Republic, Aug. 25-27, 2006, pp. 167-172

[10] Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of the 20thVLDB conference, pp 4S7-499 660