**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

***IJCSMC, Vol. 3, Issue. 5, May 2014, pg.464 – 471***

**RESEARCH ARTICLE**

# Implementation of Preprocessing Techniques in Datamining

**A. Abdullah[1], O. Fadhil [2]**

[1, 2] Department Computer engineering, Eastern Mediterranean University, North Cyprus / Turkey

[1] alharith_alkafije@yahoo.com, [2] osamahyassen@yahoo.com

***Abstract— carefully screened can produce misleading results. Thus, the raw data needs to pre-process before doing data mining. And often-times, this step can take considerable amount of processing time. Usually, data from experiments are not suitable for doing data mining tasks. Because of the raw data may contain out-of-range-values, impossible data combination or missing value etc. Analyzing data without being Data pre-processing includes cleaning, normalization, transformation, feature selection and extraction etc. The product of data pre-processing is the final training data set.***

***In our research, we do discretization, calculating similarity or distance between objects, normalization, and find a correlation between objects or attributes in a data set to gain better analyze before main pre-processing steps.***

***Keywords— Discretization; Correlation; Normalization; Euclidean distance; Cosine similarity***

## I. INTRODUCTION

Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Commonly used as a preliminary data mining practice, data preprocessing transforms the data into a format that will be more easily and effectively processed for the purpose of the user. There are a number of different tools and methods used for preprocessing, including: sampling, which selects a representative subset from a large population of data; transformation, which manipulates raw data to produce a single input; DE noising, which removes noise from data; normalization, which organizes data for more efficient access; and feature extraction, which pulls out specified data that is significant in some particular context. Before using these methods to preprocessing raw data, we should apply some preliminary operation on data, such as discretization, calculating similarity or distance between objects, normalization, and find a correlation between objects or attributes in a data set to gain better analyze before main preprocessing steps that mentioned above. In this research, basic operations applied on given data sets that include information about 181 genes with 500 attributes. Implementations of those algorithms were based on the approaches that are in a data mining.[1]

## II. PROGRAM STRUCTURE

For executing program we should load the main.m file to matlab and run it, other functions will call from this function based on user's choice this program made of some function and has a main menu to selecting what should it do? The first main menu is in this shape:

- Please enter proper number:
- For discretizing data:
- For Euclidean distance and cosine similarity between 10 gene

 

- For calculate correlation between 10 attributes
- For normalization original data and after that calculate correlation between the same attributes
- EXITE

This program will run until user enters the number 5. If number 1 imported by user, system will draw plots for discretization diagrams based on the number of column that will import by him for the next step. For number 2 firstly 10 genes indexes will generate randomly and after that system calculate the Euclidean distance and cosine similarity between each possible pairs of those genes. In the case of number 3 such as previous section, 10 attributes indexes will generate randomly and after that the correlation between each pair will calculate. Lastly if user imports number 4, original data will be normalized based on given formula and after that all proceeding steps will repeat again on new data. We used some functions to implementation of this program such as:

main(),dirstselect(data),secondselect(data),thirdselect(data),euclidean(x,y),cosine(x,y),correlation(x,y),normalization(data),firstdiscretize(data),discretize(data,num).

All of these functions will explain in a proper respective section. The last thing that seems necessary for mention here is, in this program we used two global variables to keeping the randomly generated indexes for genes and attributes to compare result on same data before normalization and after normalizationAll of these functions will explain in a proper respective section. The last thing that seems necessary for mention here is, in this program we used two global variables to keeping the randomly generated indexes for genes and attributes to compare result on same data before normalization and after normalization.

## III. DISCRETIZATION

When number 1 is entered by user in first menu, function first discretizes (data) will call for getting a column number from user to discretizing based on that number. After that system will call the discretize(data,num) function to apply equal-width and equal-frequency discretization. This function works on this way: first of all specified column drawn from the original data then we plot all of the genes based on that column's value for equal-width discretizing we calculate the maximum and minimum value of that column and calculate the interval width with this formula: w= (max-min)/number of intervals, finally we plot tree line based on that interval on the previous diagram. For equal-frequency firstly we draw a diagram for all genes based on that column number again after that we should calculate the frequency of data, for this reason we use one of the matlab functions : tiedrankes(newdata) [2],and we should partition data to 4 groups so we use this formula: partition=ceil(4*tiedrank(newdata)/max(size(newdata))) and finally for calculating intervals we calculate the mean of each group of genes in a partition and drew the interval's lines. [3]

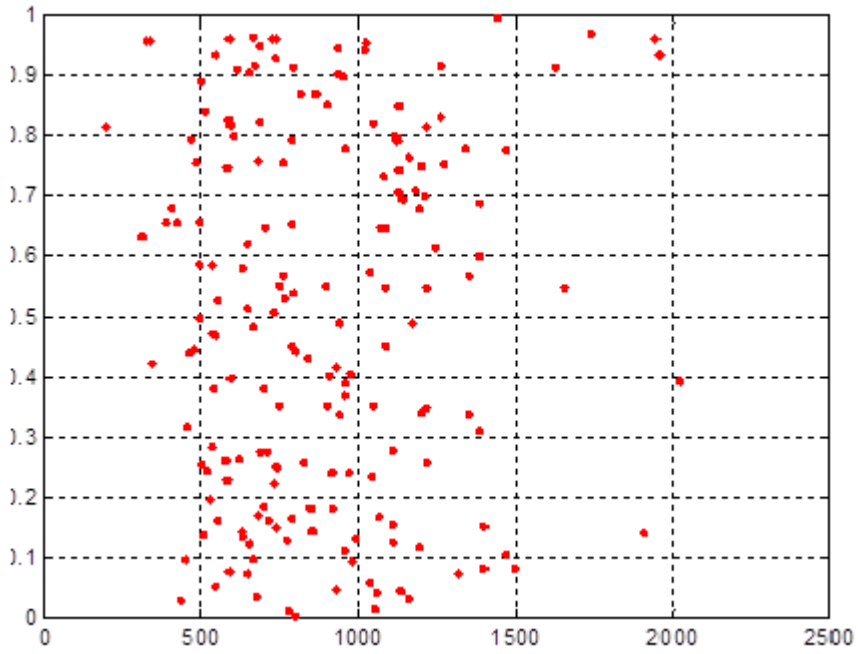


Fig 1: Equal width discretization attributes 10

 

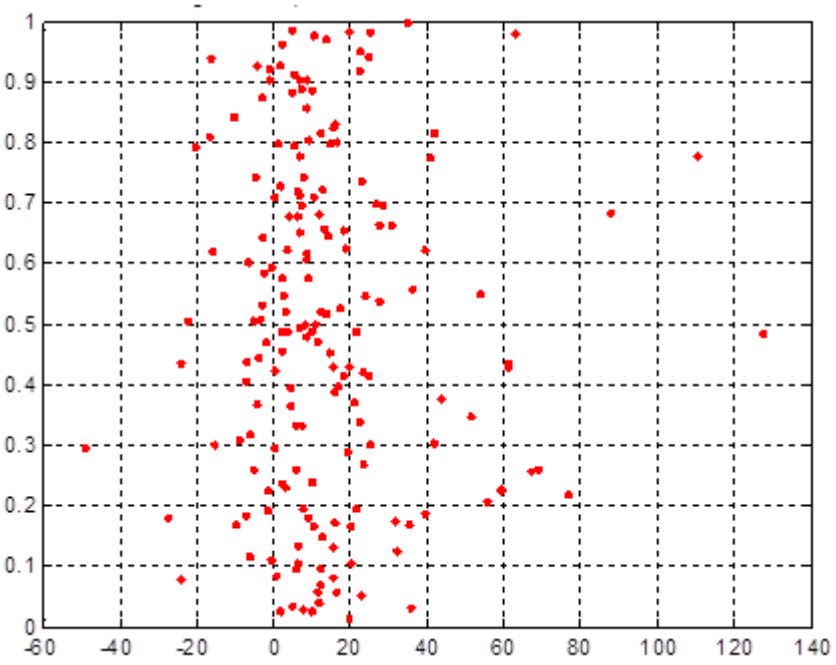


Fig 2 Equal width discretization attributes 100

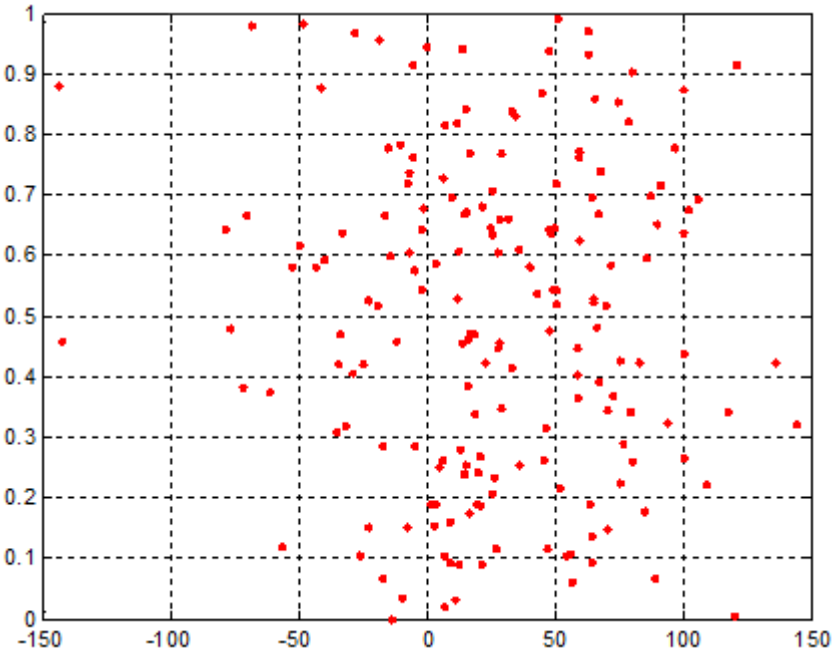


Fig 3 Equal width discretization attributes 200

 

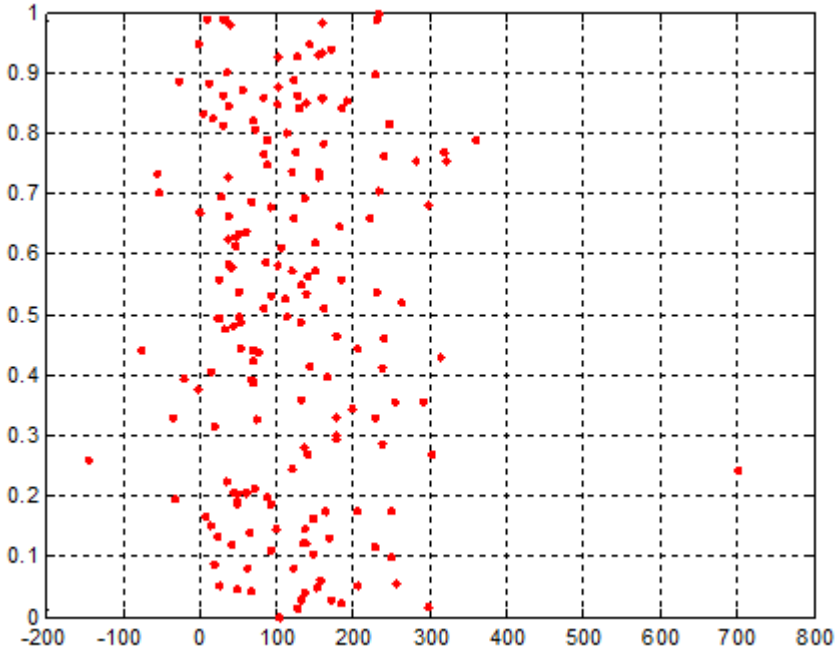


Fig 4 Equal width discretization attributes 300

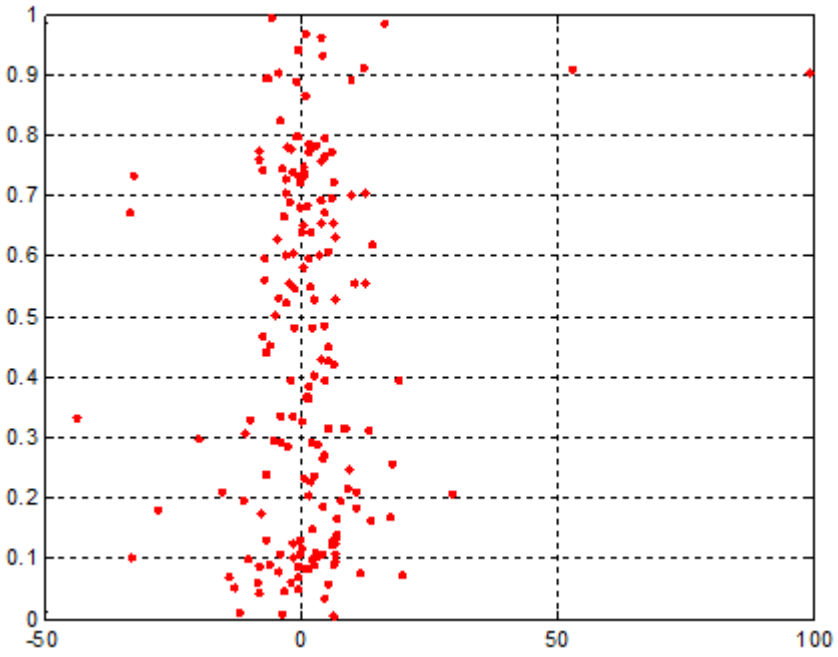


Fig 5: Equal width discretization attributes 450

 

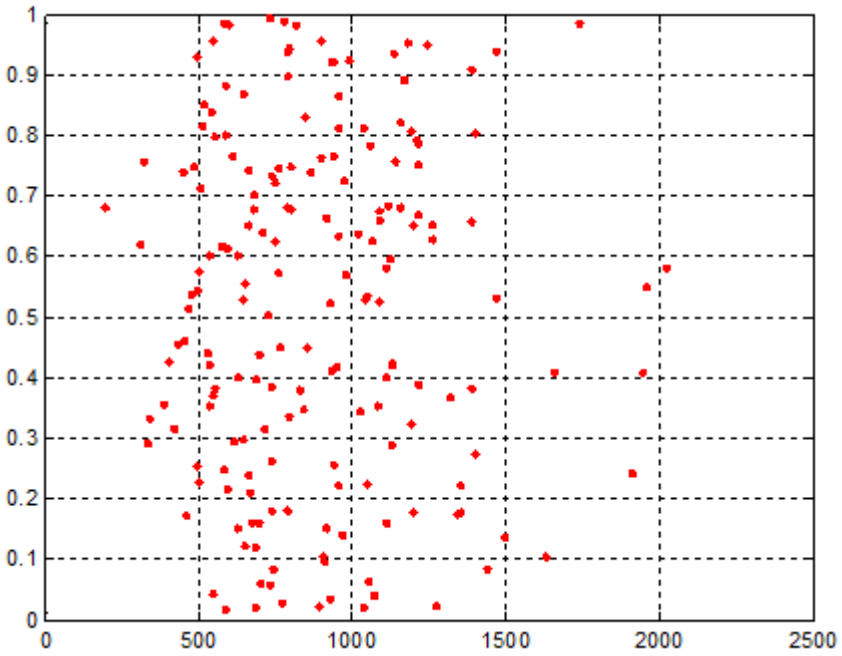


Fig 6 Equal frequency discretization attributes 10

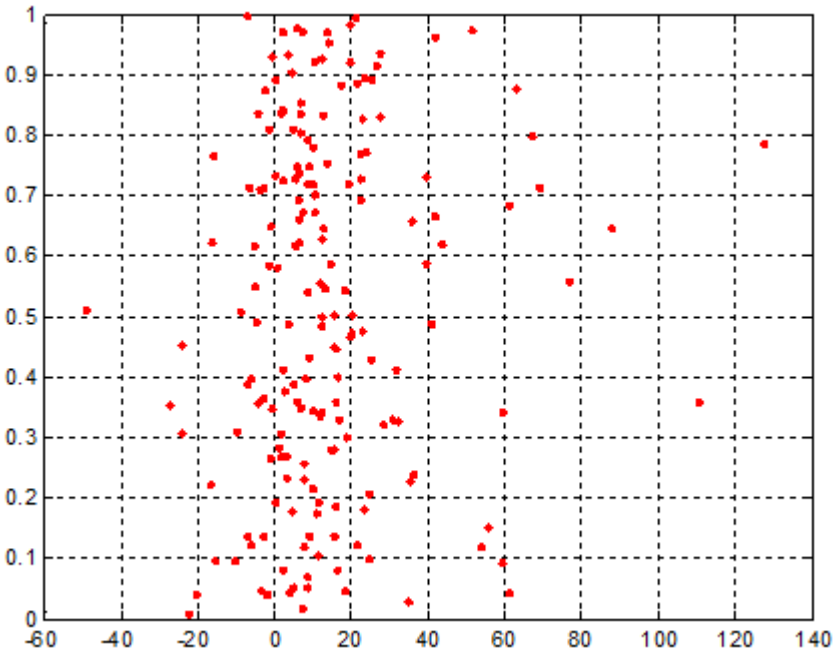


Fig.7 Equal frequency discretization attributes 100

 

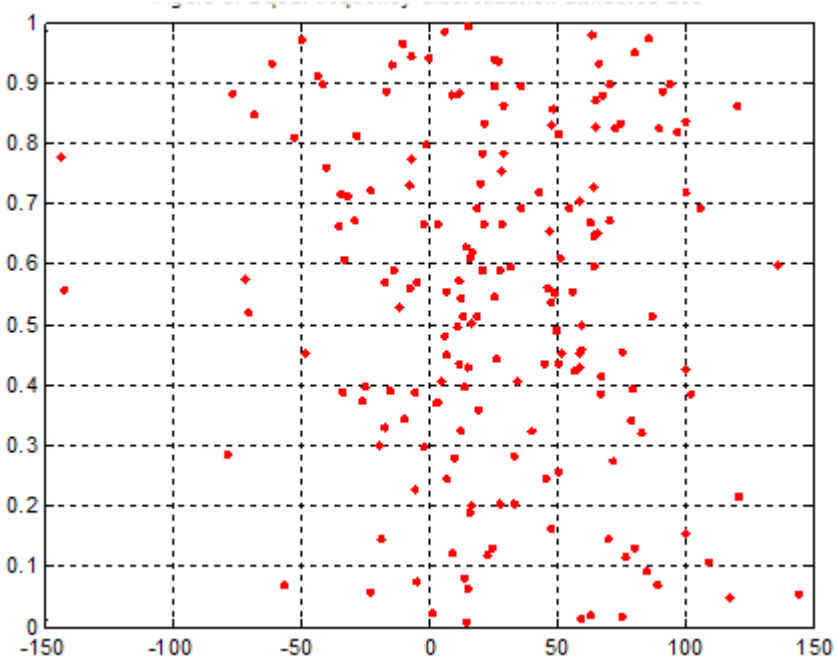


Fig.8: Equal frequency discretization attributes 200

## IV. EUCLIDEAN DISTANCE AND COSINE SIMULARITY

This part of program will start with calling Euclidean(x,y) and cosine(x,y) functions respectively to calculating Euclidean distance and cosine similarity between two genes. Because we should normalize data on next steps and compare the results of calculating these functions before normalizing and after normalizing on the same genes we use a global variable to keep the number of genes that produced with random function **inp=randi(181,1,10)** in main program. For calculating distance and similarity we implement above functions based on their definitions in the course text book. Because Euclidean distance and cosine similarity have symmetry property, so we calculate distance and similarity just for one pair. for example because cosine(g1,g2)=cosine(g2,g1) and Euclidean(g1,g2)=Euclidean(g2,g1) , we calculate distance and similarities for one side of those equality. Totally for 10 genes we should apply those functions on 45 pairs of genes.[3] For example output of this section for each pair will be in this mode:

- Euclidean distance between gene with number 166 and gene with number 43 is:17693.86.
- Cosine similarity between gene with number 171 and gene with number 13 is: 0.92

## V. CORRLEATION

When number 3 is entered in main menu, firstly function secondselect(data) will be called. Inside this function, we define a global variable **inp2** to keeping number of attributes that take its value from the main program randomly. We used global variable, because we should apply this function on the same indexes of attributes after normalization. For calculating correlation we implement correlation function with two input parameter based on formula given in text book. Same as the previous section we pass the 10 attributes indexes from the main program to the secondselect(data) via the global **inp2** variable. Then we calculate correlation for each pair of attributes and put the attribute1, and attribute2 and their correlation's value into the matrix **matz** we used this matrix for finding 10 most positively and negatively correlated pairs, also the most positive correlation and the most negative correlation will be calculate.[4,5] Parts of output for this section on random attributes number have been shown below:

Attributes indexes are: 79 486 479 243 401 71 211 458 397 480
Correlation between attribute with number 486 and attribute with number 458 is: -0.025511
Correlation between attribute with number 486 and attribute with number 397 is: -0.076498
The most positively correlation is between attribute with number 79 and number 480 with correlation value: 0.318557
The most negatively correlation is between attribute with number 79 and number 71 with correlation value: -0.180

 

A) 10 most positive correlation

| number1 | number2 | corvalue |
|---|---|---|
| 79.0000 | 486.0000 | 0.3186 |
| 79.0000 | 479.0000 | 0.2843 |
| 79.0000 | 243.0000 | 0.2675 |
| 79.0000 | 401.0000 | 0.2402 |
| 79.0000 | 71.0000 | 0.2236 |
| 79.0000 | 211.0000 | 0.1996 |
| 79.0000 | 458.0000 | 0.1928 |
| 79.0000 | 397.0000 | 0.1712 |
| 79.0000 | 480.0000 | 0.1238 |
| 486.0000 | 479.0000 | 0.1234 |

B) 10 most negative correlation

| number1 | number2 | corvalue |
|---|---|---|
| 397.0000 | 480.0000 | -0.1810 |
| 458.0000 | 480.0000 | -0.1800 |
| 458.0000 | 397.0000 | -0.1535 |
| 211.0000 | 480.0000 | -0.1498 |
| 211.0000 | 397.0000 | -0.1126 |
| 211.0000 | 458.0000 | -0.0765 |
| 71.00 | 480.0000 | -0.0450 |
| 71.0000 | 397.0000 | -0.0355 |
| 71.0000 | 458.0000 | -0.0279 |
| 71.0000 | 211.0000 | -0.0255 |

Because in given data, we don't have any pairs of attributes that their correlations are zero, so we don't have uncorrelated attributes.

## VI. NORMALIZATION

If number 4 is entered in main menu, function thirdselect(data) will be call and first of all the maximum and minimum of original data will calculate. The normalization function will be call- fn(data). Normalization will be applied on original data based on the given normalization approach in the assignment context, after that for making sure that approach is true, we calculate maximum and minimum of new normal data .if they are in [-1,1] so we can say that normalization is true.
After normalization we will call the previous functions with new data again with the same index of genes and attributes. For this reason we generate the random attributes indexes and genes indexes and put them into global variables **inp** for genes and **inp2** for attributes in the main function.
When we compare results of two times calculating Euclidean distance and cosine similarity between tow genes first for original data and second for normal data we consider that values for the distance and similarity will be reduced when apply those functions for normal data.[6] For example for the below gene indexes results of applying those, have shown for some pair of genes before and after normalization:

Gene indexes are: 148 164 23 166 115 18 51 99 174 175

*BEFOR* NORMALIZATION----------

Euclidean distance between gene with number 164 and gene with number 23 is: 5944.370700

Cosine similarity between gene with number 164 and gene with number 23 is: 0.872539

*AFTER* NORMALIZATION----------
Euclidean distance between gene with number 164 and gene with number 23 is: 8.376869

Cosine similarity between gene with number 164 and gene with number 23 is: 0.703112
----------

 

**For correlation**, calculating correlation after normalization of data shows the results for correlation values between attributes for the same attributes indexes, are same as the original data and 10 most positively correlated and 10 most negatively correlated pairs of attributes for new data are same as the original data.

For example:

Attributes indexes are: 79 486 479 243 401 71 211 458 397 480

*Before* normalization ----------

Correlation between attribute with number 79 and attribute with number 486 is: 0.123761

Correlation between attribute with number 79 and attribute with number 479 is: 0.114583

**********

The most positively correlation is between attribute with number 79 and number 480 with correlation value: 0.318557

The most negatively correlation is between attribute with number 79 and number 71 with correlation value: -0.180980

*After* normalization----------

Correlation between attribute with number 79 and attribute with number 486 is: 0.123761

Correlation between attribute with number 79 and attribute with number 479 is: 0.114583

**********

The most positively correlation is between attribute with number 79 and number 480 with correlation value: 0.318557

The most negatively correlation is between attribute with number 79 and number 71 with correlation value: -0.180980

## VII. CONCULATION

In this research, we prepared some strategies and techniques for converting raw data to a format that is acceptable to data mining or model learning algorithms. In this work, information is pre-processing into discretization, Similarity and normalization. Where in the discretization we are used the equal width approach and equal frequency approach and we found how it is cluster the data when we are comparing with the original data set.

Also we are used the normalization and it is very important step especially when we dealing with parameters of different units and scales and we concluded that the normalization necessary to avoid having a variable with large values dominate the results of the calculation.

## REFERENCES

[1] P.-N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining", Addison Wesley, 2006.
[2] http://matlabdatamining.blogspot.com/2007/02/dividing-values-into-equal-sized-groups.html
[3] Ruoming Jin, Yuri Breitbart and Chibuike Muoh," Data Discretization Unification", , Seventh IEEE International Conference on Data Mining, 2007.
[4] http://en.wikipedia.org/wiki/Data_Pre-processing.
[5] Yi Lu, "Advanced data mining techniques for identifying correlation between gene expression and promoters", Wayne State University, 2006.
[6] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, "Data Preprocessing for Supervised Leaning", International Journal of Computer Science, 2006, Vol 1 N. 2, pp 111-117.

 