Design and Implementation of Collage Plane Model Using Decision Tree and Clustering Algorithms

Hasan H. Hamody
Al Mansour University College

Mustafa Sabah Mustafa
Al Mansour University College

Abstract:

The amount of data kept in computer files and databases are growing at a phenomenal rate .the users of these data are expecting more sophisticated information from them .Simples SQL structure query language queries is not adequate to support these needs.

A data-mining algorithm is the mathematical and statistical algorithm that transforms the cases in the original data source into the data-mining model. However the model looks depends largely on the data mining algorithm applied to the. Where a decision tree algorithm is used to analysis the data and creates a repeating series of branches until no more relevant branches be made. The end result is a binary tree structure where the splits in the branches can be followed along specific criteria to find the most desired result .unlike a decision tree, a cluster a algorithm, does not split data along any lines but rather group's data in clusters. Clustering is most useful for visual representations because the data is ground around common criteria.

The general objective is to build trees are as balanced as possible, in terms of the distribution of attributes. in other words the algorithm seeks to put as many of one type of attribute as possible in a given node another commonly used algorithm to create models is clustering . The algorithm creates a model that cannot be used to make predictions but is very effective in finding records that have attributes in common with each other. In this research the decision support for database and data mining algorithms (decision tree and clustering) are explain, the goal of data miming algorithm in decision support is to create rules (for every node which can expressed as a set of rules that provide a description of the function of that particular node in the tree as well as the node that led up to it). We build a system called (DM college plan) that using decision tree and clustering algorithms, the goal is to create module that can be used to finds records that have attributes in common with each other.

1.1 Introduction:

Data mining defined as finding hidden information in a database alternatively it has been called exploratory data analysis, data driven discovery, and deductive learning. It is a decision support tool that stands on its own when it comes to analyzing large data bases. It has own unique features, which are designed to address unique decision support problems that cannot be solved by other data analysis tools.

A data-mining model is a flexible structure that is designed to support the nearly infinite number of ways data can be modeled. The data-mining algorithm gives the data mining model shape, form, and behavior. Decision trees and clustering algorithms are explained in this paper. These algorithms are very different in behavior and produce very different models. Both algorithms can be used together to select and model data for business scenarios the goal of data miming algorithm in decision support is to create rules(for every node which can expressed as a set of rules that provide a description of the function of that particular node in the tree as well as the node that led up to it). We build a system called (DM college plan) that using decision tree and clustering algorithms, the goal is to create module that can be used to finds records that have attributes in common with each other. [1].

1.2 Decision Trees:

A decision tree is a predictive modeling technique used in classification, clustering, and prediction tasks. Decision trees use a divide and conquer technique to split the problem search space into subsets[2]. Definition (1) A Decision Tree is a tree where the root and each internal node is labeled with a question. The arcs emanating from each node represent each possible answer to the associated question. Each leaf node represents a prediction of a solution to the problem under consideration.

Definition (2) A Decision Tree (DT) Model is a computational model consisting of three parts:

1. A decision tree as defined in definition (1.).

2. An algorithm to create the tree.

3. An algorithm that applies the tree to data and solves the problem under consideration[2].

The building of the tree may be accomplished via an algorithm, which examines data from a training sample or could be created by a domain expert. Most decision tree techniques differ in how the tree is created. Algorithm (1) shows the basic steps in applying a tuple to the DT, step three in Definition (2). Assume here that the problem to be performed is one of prediction, so the last step is to make the prediction as dictated by the final leaf node in the tree.

Algorithm (1): DT Proc Algorithm (‘Simplistic algorithm to illustrate Prediction

Technique using DT)

```
Input: T (Decision Tree,)
       D (Input Database,)
Output: M (Model Prediction,)
      for each t ∈ D DO
        n=root node of T
      while n not leaf node do
        Obtain answer to question on n applied t;
        Identify arc from t, which contains correct answer;
        n= node at end of this arc,
    Make prediction for t based on labeling of n,
```

The decision tree approach is most useful in classification problems. With this technique, a tree is constructed to model the classification process. Once the tree is built, it is applied to each tuple in the database resulting in a classification for that tuple. So there are two basic steps in the technique:

building the tree and applying the tree to the database[3, 4]. A definition for a decision tree used in classification is contained in Definition (3) below. Definition (3) Given a database $D=\{t_1,\ldots.\ ,t_n\}$ where $t_i\ =<t_{i1},..\ t_{in}$ and the database schema contain the following attributes $\{A_1,A_2\ ,\ldots..A_h\}$. Also given is a set of classes C={C1,...,C1}. A Decision Tree (DT) or Classification Tree is a tree associated with D that has the following properties: Each internal node is labeled with an attribute, A, Each arc is labeled with a predicate, which can be applied to the attribute associated with the parent.

Each leaf node is labeled with a class, C.

Solving the classification problem using decision trees is a two step process:

1. Decision Tree Induction: Construct a DT using training data.
2. For each $t_i \in$ D apply the DT to determine its class.

There are many advantages to the use of DT for classification. They are certainly easy to use and efficient. Rules can be generated which are easy to interpret and understand. They scale well for large databases, as the tree size is independent of the database size. Once a tree is built, the application of the tree to a database is 0 (n). Each tuple in the database must be filtered through the tree. This takes time proportional to the height of the tree, which is fixed. Trees can be constructed for data with many attributes. Disadvantages also exist for the DT algorithms.Simple DT Build Algorithm (2) illustrates the tree building phase. Attributes in the database schema, which will be used to label nodes in the tree and around which the divisions will take place are called the splitting attributes. The predicates by which the arcs in the tree are labeled are called the splitting predicates[2,3] .

Algorithm (2): DT Build Algorithm (Simplistic algorithm to illustrate naïve approach to building DI)

```
 Input: D (Training data)
Output: T (decision Tree)
     T= 0
```

```
            Determine best splitting criterion;
            T=Create root node and label with splitting attribute;
            T=Add arc to root node for each split predicate and label;
        For each arc do
            D=database created by applying splitting predicate to D;
            If stopping point reached for this path then
            TT =Create leaf node and label with appropriate class;
            Else
              TT =DT Build(D);
             T=Add TT to arc;
```

Notice that the major factors in the performance of the DT Build algorithm is the size of the training set and how the best splitting attribute is chosen. Most DT algorithms face the following issues: ***Choosing Splitting Attributes, Ordering of Splitting Attributes, Splits, Tree Structure, Stopping Criteria, Training Data, Pruning.***

1.3 Clustering:

The Clustering algorithm provider is typically employed in association and clustering tasks, because it focuses on providing distribution information for subsets of cases within data. The Clustering algorithm provider uses an expectation- maximization (EM) algorithm to segment data into clusters based on the similarity of attributes within cases[1] .The clustering algorithm provider is best used in situations where possible natural groupings of cases may exist, but are not readily apparent. This algorithm is often used to identify and separate multiple patterns within large data sets for further data mining; clusters are self- defining, in that the variations of attributes within the domain of the case set determine the clusters themselves.

A classification of the different types of clustering algorithms has been given in Figure (1).

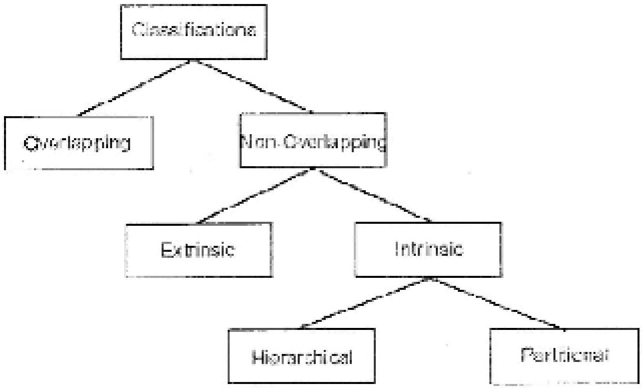


Figure (1) Classification of Clustering Algorithms

They view clustering as a special type of classification, which is a common view even though. At the highest level, clusters can be either overlapping or non-overlapping. Even though that is considered a non-overlapping, it is possible to place an item in multiple clusters. in turn, non-overlapping can be viewed as extrinsic or intrinsic. Extrinsic techniques use labeling of the items to assist in the supervised learning algorithms where a special input training set is used, Intrinsic algorithms do not use any a priori category labels, but only depend on the adjacency matrix containing the istance between objects. Clustering algorithms may be viewed as hierarchical or partitional. With hierarchical clustering, a nested set of clusters is created. Each level in the hierarchy has a separate set of clusters. At the lowest level each item is in its own unique cluster. At the highest level, all items belong to the same cluster.

With hierarchical clustering, the desired number of clusters in not input. With partitional clustering, the algorithm creates one set of clusters only. These use the desired number of clusters to derive how the final set is created[2,3,4].

1.3.1 Squared Error Clustering:

The Squared Error clustering algorithm minimizes the squared error. The squared error for a cluster is the sum of the squared Euclidean distance between each element in the cluster and the cluster centroid, Ck. given a cluster k1 let the set of items mapped to that be {t_{i1}, t_{i2}, . . . ,$t_{im,}$,} The squared error is defined as: [2]

$$seki = \sum_{j=1}^{im} \|t_{ij} - c_k\|^2$$

Given a set of clusters k=Z{k1, k2 , k}, the squared error fork is defined as:

$$SeK = \sum_{j=1}^{k} sek_j$$

The algorithm of squared error-clustering structure is shown in algorithm (3) below.

Algorithm (3)

Input:

D= {t_1, t_2 ,…..,t_n} //Set of elements.

A // Adjacency matrix showing distance between elements.

k /7 Number of desired clusters.

Output:

K // Set of clusters.

Squared Error Algorithm:

```
  Assign each item t, to a cluster;
  Calculate center for each cluster;

Repeat
Assign each item ti to the cluster, which has the closest center;

Calculate new center for each cluster,
Until the difference between successive squared errors is below a threshold;
```

1.3.2 K-Means Clustering:

K-i1'leans is an iterative clustering algorithm where items are moved among sets of clusters until the desired set is reached. As such it may be viewed as a type of squared error algorithm although the convergence criteria need not be defined based on the squared error. A high degree of similarity among elements in clusters is obtained while simultaneously a high degree of dissimilarity among elements in different clusters is also achieved. The cluster means of K, = {t_{i1}, t_{i2}, . . . ,$t_{im,,}$} is defined as:

$$m_i = \frac{1}{m}\sum_{j=1}^{m} t_{ij}$$

It is assumed that each tuple has only one numeric value as opposed to a tuple with many attribute values. The K-Means algorithm assumes that some definition of cluster mean exists, but it does not have to be this particular one. This algorithm assumes that the desired number of clusters, k, is an input parameter. Algorithm (3) shows the K-Means algorithm..

Algorithm (4)

```
Input:
    D={$t_1$, $t_2$,… ,$t_n$} //Set of elements.
    A // Adjacency matrix showing distance between elements.
    k // Number of desired clusters.
Output:
     K // Set of clusters.

K- Means Algorithm:
Assign initial values for means $m_1$, $m_2$,..., $m_k$;
Repeat
Assign each item ti to the cluster, which has the closest center;

Calculate a new means for each cluster;
Until convergence criteria are met;
```

1.4 proposed system:

The proposed system is used to control and view the data mining operations. One of the most the data mining operations is a decision tree. Decision trees use statistical techniques to infer rules from data sets-where a rule is a correlation found between a given variable (the dependent variable) and one of the other variables (independent variables). Because these rules can be used to make predictions on new data, decision trees are predictive models. Visually, decision trees look like inverted trees -the root appears at the top of the tree; branches grow downward. The point where a branch splits represents a correlation.

The proposed system uses two techniques: Decision tree viewer and lift chart. Decision tree viewers is used to represent and view the decision trees. Lift charts are used to judge predictive models by their performance that is, their ability to classify and make predictions. The evaluation is performed with a ratio known as Lift, which measures the change in concentration of a particular class when the model is used to select a purposefully based sample from the general population of records.

1.5 *System Implementation:*

With more than 250 clients, network planning becomes a lot more challenging. This number of clients tends to be spread out over large areas than can be supported from a central computer room. This geographic aspect requires both a distributed network and a lot of servers. Usually, a network of this size will be connected with a high-speed backbone that runs between servers. Because of the cost of high-speed protocol network equipment, high-speed backbone networks tend to be dramatically more expensive than the smaller networks presented above.In this section, the implementation of the proposed system will be explained. The proposed system is used to display the information about any data mining information. It can perform the actual data mining computation. The collegePlan.mdb used as example database to explain the work of the proposed system. This database contains about 9000 records.

To understand the work steps of proposed system we will explain the outline and algorithms as below:

STEP 1 :Delivering Data Mining Information.

The proposed system can view data mining information of any database tables after doing some computations steps.

STEP 2: Displaying the Decision Trees.

A user can view the decision tree of his data mining when he enters this table of data mining to the proposed system.

STEP 3: Display the Clustering Maps.

The system can calculate and view clustering maps of the data mining tables.

STEP 4:Displaying the Models in the Lift Chart.

The system can display the predictive model perfonance for judging the correctness of the models. Also this chart is used to check the ability of the predictive models to classify and make predictions.

STEP 5:Printing the Results:

The proposed system can print the result from the steps above to make the user have report about this step.

The above steps are the outline of the proposed system.

The block diagram of proposed system would be shown in Figure (2) below.

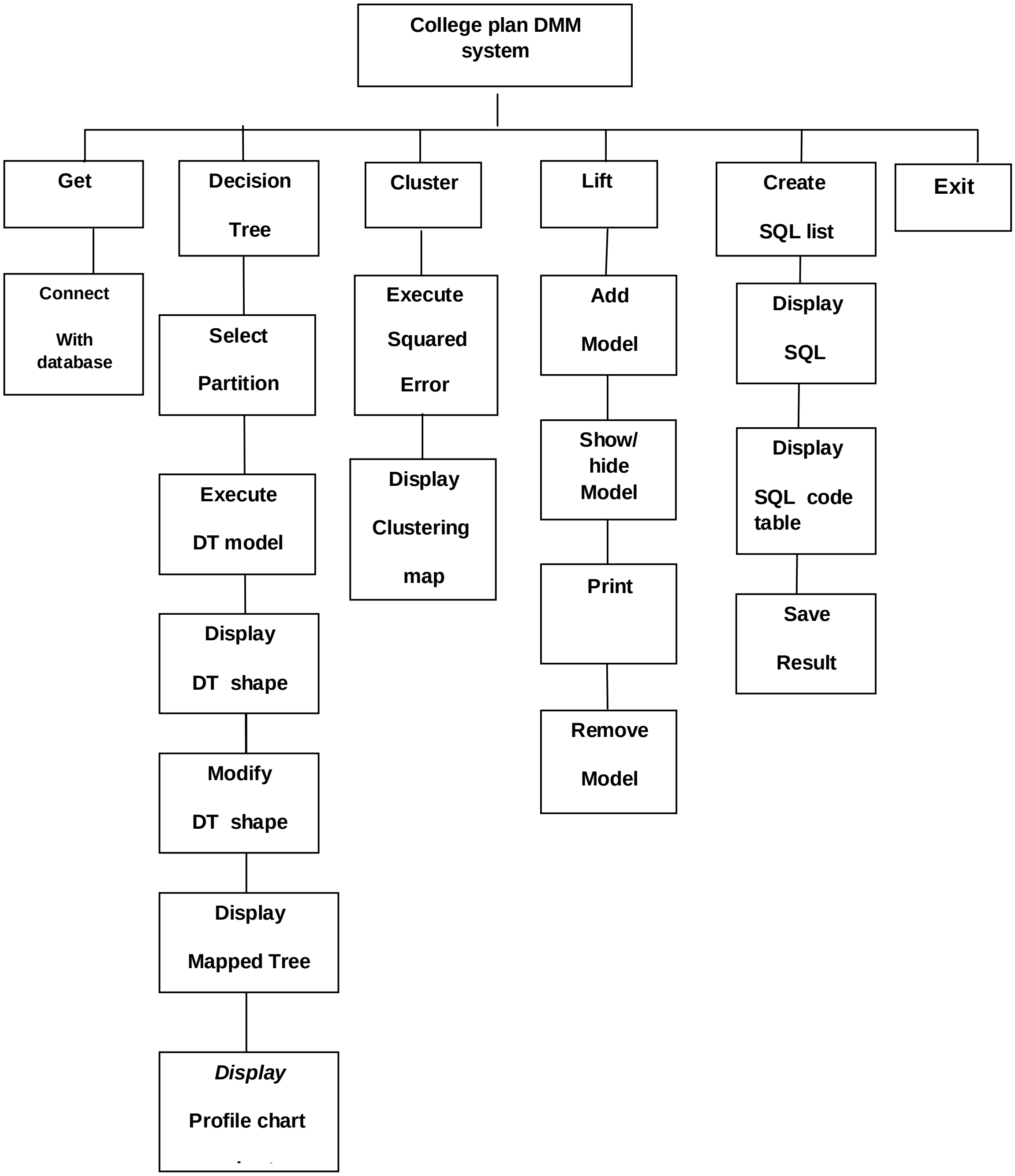


Figure (2)college plan DMM system diagram.

1.6 Flow Chart of the Proposed System:

The main interface of the proposed system has five flow chart as is follows:

- **First is related with connection with database as shown in figure(3).**
- **Second correspond to create decision tree model for that database as shown in figure(4).**
- **Third corresponds to create a clustering model as shown in figure(5).**
- **Fourth is used to show the prediction model for selected field as shown in figure(6).**

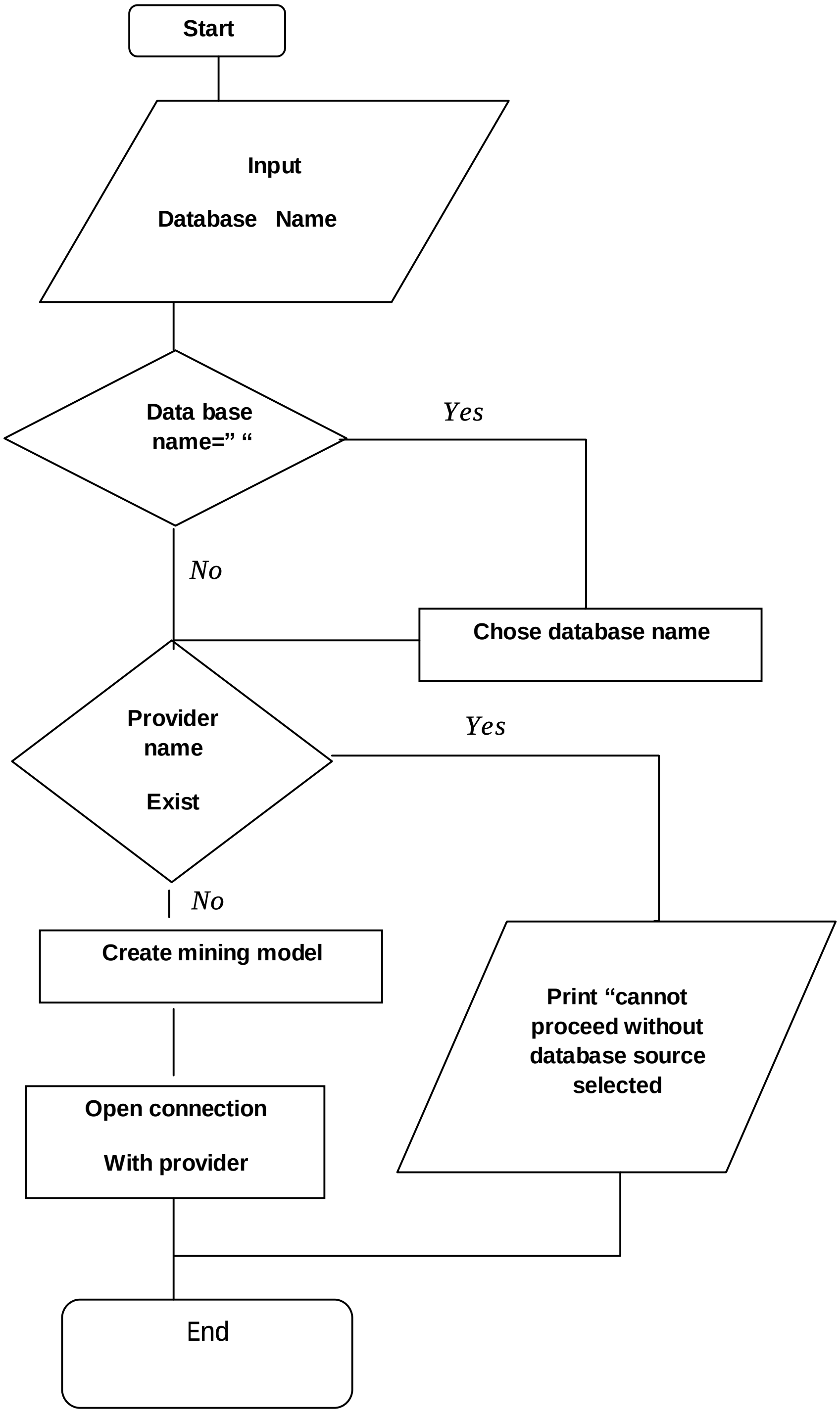


Figure (3) flow chart for getting connection

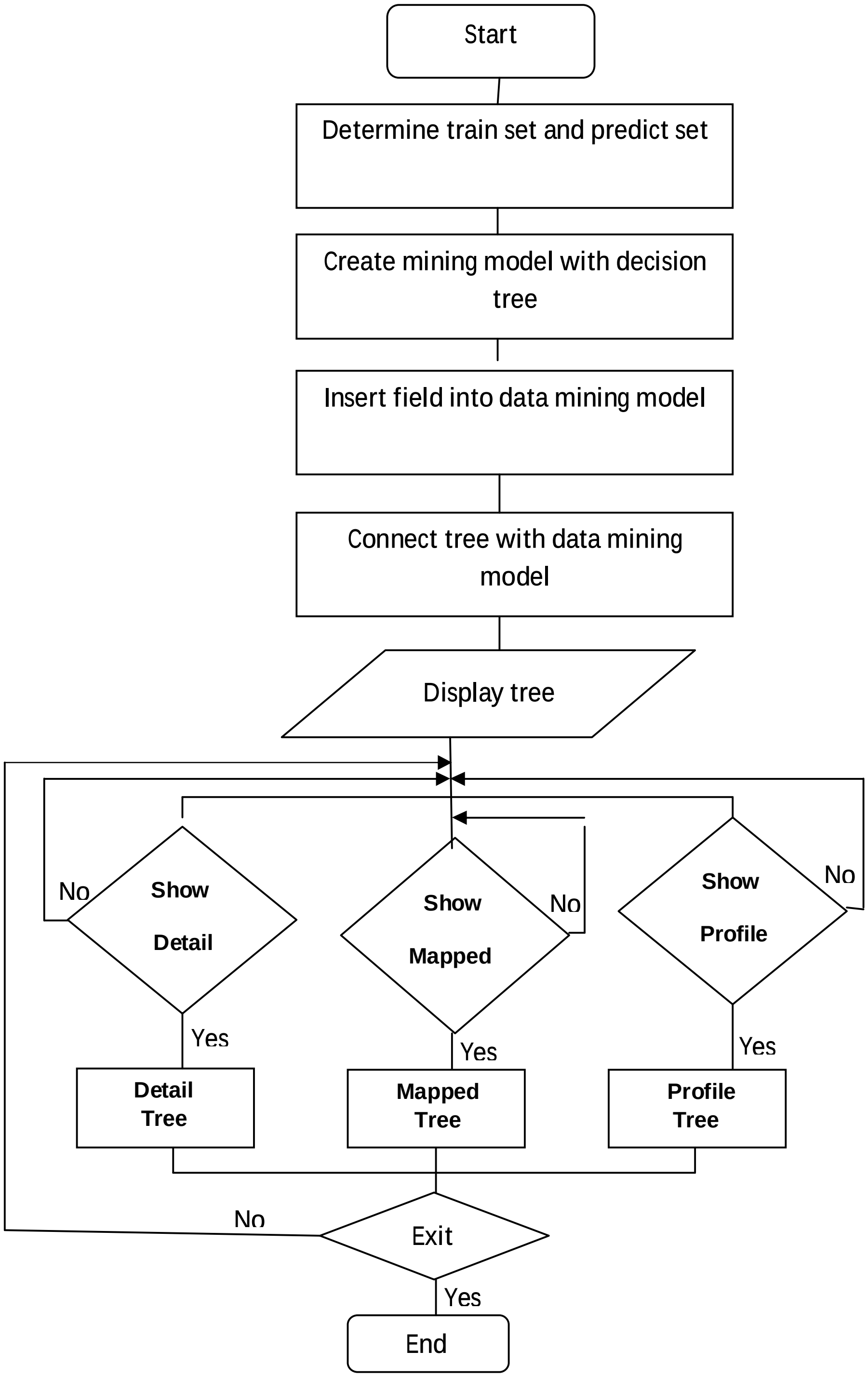

Start
Determine train set and predict set
Create mining model with decision tree
Insert field into data mining model
Connect tree with data mining model
Display tree
Show Detail
No
Yes
Show Mapped
No
Yes
Show Profile
No
Yes
Detail Tree
Mapped Tree
Profile Tree
Exit
No
Yes
End

Figure (4) flow chart for decision tree

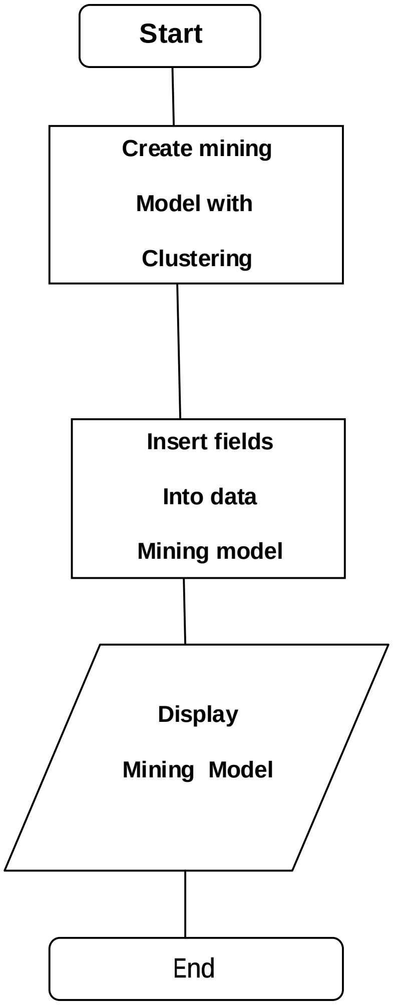


Figure (5) flow chart for clustering

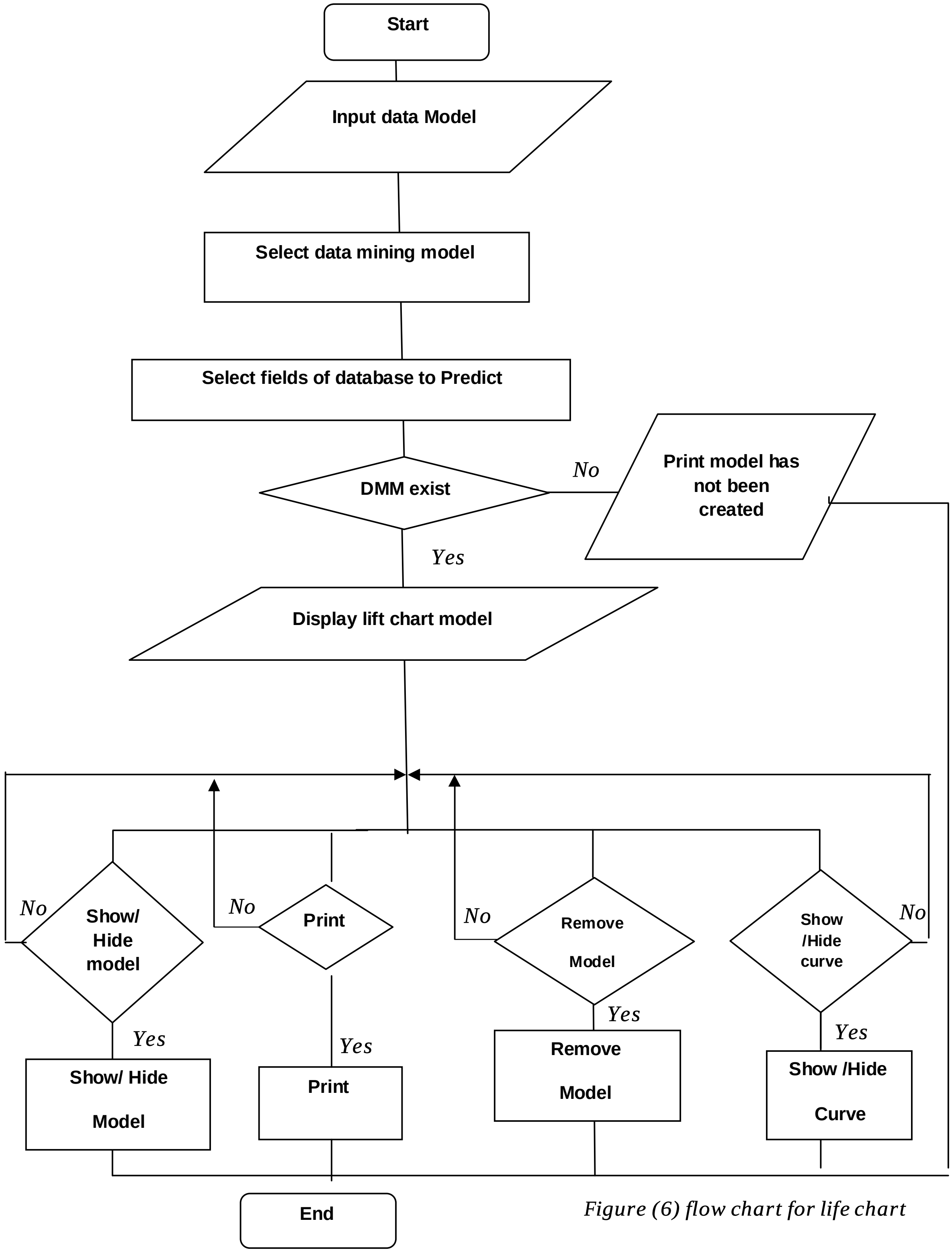


Figure (6) flow chart for life chart

1.7 Interface of the proposed system:

From figure (7) there are six buttons used to operate the proposed system, these buttons are:

Get connection, decision tree , clustering , lift chart model, create SQL list, exit.

Some of these buttons will be expressed below.

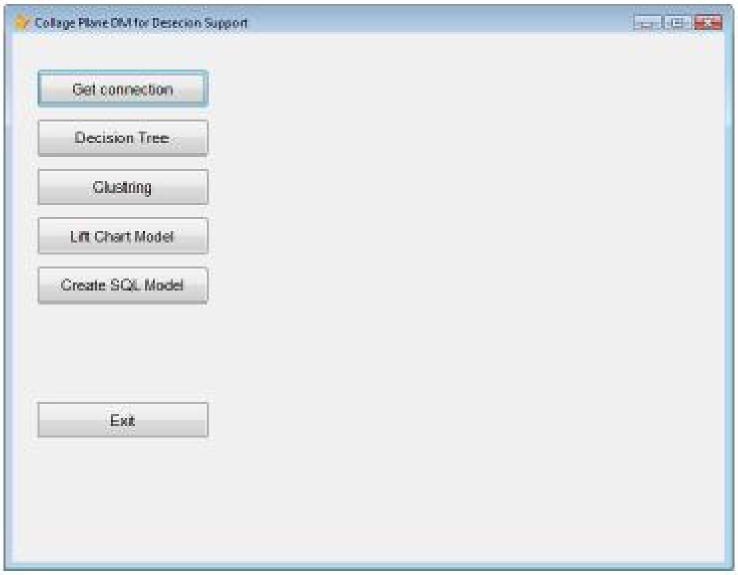


Figure (7) interface of proposed system

1.7.1 Get connection:

This option is used to select database table, when the user click on this button the system will display the open file dialog for the user to select the database file. The system will load the data mining table for doing sub operation for extraction data mining information from this table. The figure (8) show this operation.

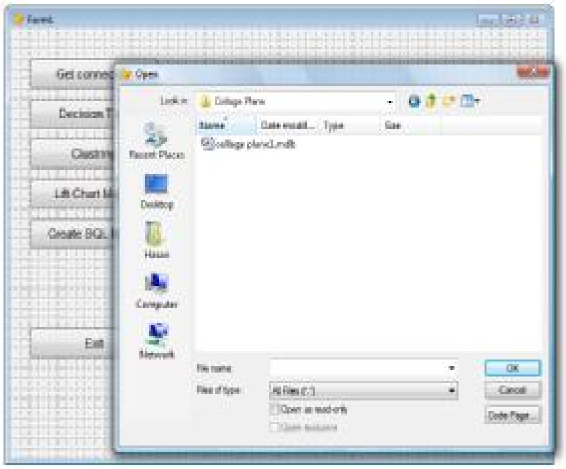
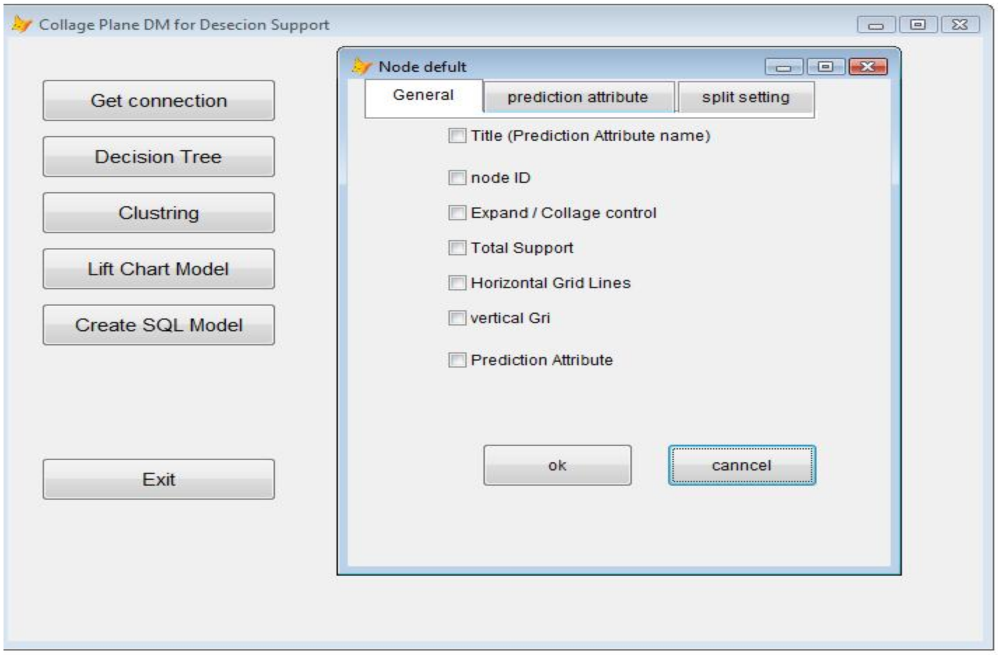


Figure (8) operation forget connection

1.7.2 Decision tree:

The second button is used to display the decision tree, after the user select the data base table and loads the system, the proposed system will apply DTProc algorithm for the predication technique and extracting the information of each node in decision tree(training data). After this operation the proposed system applies DT Build algorithm for building the decision tree from the training set, the output of this option is shown in figure (9).



□ Title (Prediction Attribute name)

□ node ID

□ Expand / Collage control

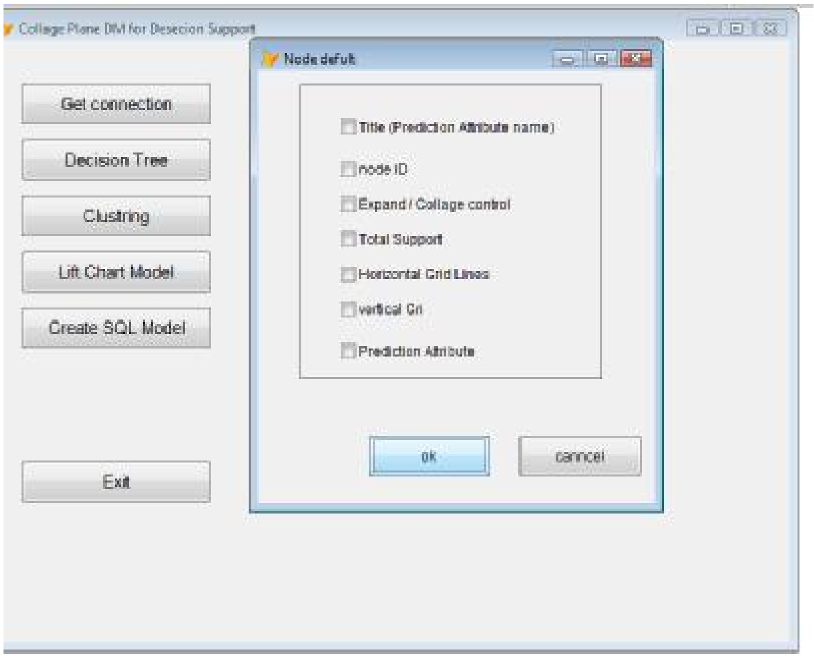
□ Total Support

□ Horizontal Grid Lines

□ vertical Gri

□ Prediction Attribute

Figure (9) Decision tree shape



□ Title (Prediction Attribute name)

□ node ID

□ Expand / Collage control

□ Total Support

□ Horizontal Grid Lines

□ vertical Gri

□ Prediction Attribute

Figure (10) general option of node

In the previous figure there are some options used to control the outlook of the decision tree. These options as (prediction attribute name, node id , expanded/ collapse control, total support , horizontal grid lines , vertical grid lines , prediction attributes.

- **The node id option is used to number the node in decision tree.**
- **Expanded / collapse control is used to expand the information of nodes or to reduce the information of this node.**
- **Total support is used to show the node value from the total value.**
- **Horizontal grid lines are used to grid the information of the node in horizontal direction.**

1.8 Conclusions:

1. **Data mining differ from the classical statistical tests in that they do not draw lines through the data spaces to classify observation.**
2. **The decision tree model is used primarily for predictive purpose, but it can also help explain how the underlying data attributes are distributed.**
3. **Clustering is similar to classification in that data is grouped. However , unlike classification, the groups are not predefined. Instead, the grouping is accomplished by finding similarities between data according to characteristics found in the data itself. Thus clustering is viewed to be driven by the data itself and is often based on the similarity between attributes values.**
4. **Prediction data was not available in clustering but good predicated data could give by using decision tree.**
5. **Increasing training data in decision tree gives more effective predictive module.**
6. **One of the disadvantages of K-means is how we can determine the best number of cluster.**

1.9 References:

[1] Anil K. jain and Richard C. Dubes, “Algorithms for clustering data”,prentice-Hall,2000.

[2] Margaret H. Dunham, “Data mining techniques and algorithms”, Hall,2002.

[3] Carrig Emerging technology Inc, “Introduction to data mining and data warehouse” Carrig Emerging technology Inc, 2002.

[4] Michael J. A. Berry and Gordan S. Linoff, “ Mastering data mining” John Wiley & Sons , Inc, 2002.

تصميم وتنفيذ لنموذج خطة كلية باستخدام شجرة القرارات وخوارزمية التجمع

م.م. حسن حسين حمودي
كلية المنصور الجامعة

م.م. مصطفى صباح مصطفى
كلية المنصور الجامعة

المستخلص :

إن البيانات التي تخزن في ملفات قواعد البيانات تزداد بمعدلات عالية وان مستخدمي هذه البيانات يتوقعون معلومات معقدة من هذه البيانات. ان خوارزمية الـ Data Mining هي خوارزمية رياضية وإحصائية حيث تحول الحالات الموجودة في البيانات الأصلية إلى نموذج الـ Data Mining . بالرغم من ذلك فان هذا النموذج يعتمد بصورة كبيرة على خوارزمية الـ Data Mining المطبقة على البيانات .

إن خوارزمية شجرة القرار (Decision tree algorithm) تستخدم لتحليل البيانات وخلق سلاسل متكررة من الفروع والتي تتوقف عند عدم وجود تفرعات مناسبة. إن النتيجة النهائية التي نحصل عليها من تطبيق هذه الخوارزمية هي هيكل شجرة ثنائية (binary tree structure) التي يمكن إتباع تفرعاتها بالاعتماد على معايير محددة لإيجاد النقطة المطلوبة. على عكس شجرة القرار فان خوارزمية التجمع (clustering)لا تجزء البيانات ولكنها تقوم بتجميع البيانات في مجاميع. التجميع يعتبر أكثر أهمية للتمثيل المرئي لان البيانات تجمع بالاعتماد على معايير شائعة.

الهدف العام من هذا البحث هو بناء شجرة غير متوازنة قدر الإمكان , بكلمات أخرى ، خوارزمية شجرة القرار تبحث لوضع أكثر من نوع من المواصفات قدر الإمكان في العقدة المعطاة. الخوارزمية الشائعة الأخرى لخلق النماذج هي خوارزمية التجميع ، هذه الخوارزمية تخلق نموذج لا يمكن استخدامه للتنبؤ ولكنه اكثر فعالية في إيجاد القيود التي تملك مواصفات مشتركة فيما بينها.

في هذا البحث تم التطرق الى خوارزميات الـ Data Mining وقواعد البيانات في اتخاذ القرار

(Decision tree and clustering) , والهدف من استخدام ال Data Mining في اتخاذ القرار هو التنبؤ أو بناء سلسلة من القواعد التي تستخدم لتصنيف المشاهدات.