A MULTI INTELLIGENT AGENT SYSTEM FOR MAXIMIZING THE DATA WAREHOUSE INVESTMENT

R. Hosny Mohammed	M. Abdel Aziem Mostafa	M. Abou Rizka
six October University	Dean of CCIT, Kingdom University	Arab Academy For Science And Technology and Maritime Transport
raed.hosny@imholding.com	m.moustafa@ku.edu.bh	m_rizka@aast.edu

Abstract: *Data analysis and mining technologies help bring business intelligence into organizational decision support systems (DSS). While a lot of data Warehousing and mining technologies are commercially available today, organizations are seeing a growing gap between powerful storage (data warehouse) systems and the business users' ability to analyze and act effectively on the information they contain. We contend that to narrow this gap that can support business users to work comfortably. This paper illustrates how to maximize data warehouse investment using intelligent software agents, we represent three models of agents ranging from primitive agent that use relational database to more advanced agents that use Data Mining over data warehousing database, agents are represented over Web-based application and it is intended to provide an organization-wide decision support capability for business users. Agents are implemented using Dot Net Framework and XML Web Services over Microsoft analysis services which are the core of Data warehouse using OLAP and Mining Tools, three data mining techniques used by those agents models; Decision Trees, Naïve Bayes and Time Series techniques, all techniques are used for the purpose of prediction, Decision Trees are used in conjunction with Naïve Bayes and proved to give better prediction accuracy after checking lift chart diagram for both algorithm, on the other hand; two Time Series algorithms for sales forecasting problems; ARIMA (Auto Regressive Integrated Moving Average) and ARTXP (Auto Regression Trees with Cross Predict), ARIMA proved to be suitable for long future forecasting series while ARTXP proved to be more accurate with short future forecasting series after checking deviation ratio for both algorithms, our Intelligent agent models are proved to simplify the complexity of data analysis and mining activities, techniques, and methods from the business users, for easy and effective use of the warehouse data.*

Keywords: Data Warehousing, Data Mining, Business Intelligence, Intelligent Agents, Decision Trees, Naïve Bayes, ARIMA, ARTXP

1. INTRODUCTION

Over the past several years, the data warehouse has evolved into a necessary foundation for the successful trend of decision-support system (DSS) development environments. With a well-developed data warehouse, we can see every aspect of every imaginable business circumstance. OLAP tools provide advanced functionality, letting us do multidimensional data analysis. They also offer the ability to page, rotate, and aggregate warehouse data to provide a “real life” view of the business situation for advanced analytical purposes.

However; the ultimate challenge for maximizing data warehouse investments is to build enhanced systems with business intelligence software solutions and satisfy user needs as done manually in the past. The motivation of providing intelligent agent based systems is abundant; also a lot of such systems are under research **[1]**. The most challenges motivating researchers in business intelligence relating to data, user. Hereafter are some of the challenges for building successful intelligent systems: Noisy Data, Data Integration, Data Refreshing, Complete support for all business users, Autonomic on behave of user, and Cooperative to user. In the following sections, we will illustrate; Data warehousing, and data mining surveys, as shown in section 2. In section 3 surveys of intelligent agents. But Intelligent agent with rational and data warehousing will be in section 4. In section 5 Performance evaluations of experimental results. Finally; conclusions and future research.

2. DATA WAREHOUSING AND DATA MINING TECHNIQUES

Data warehousing (DW) encompasses algorithms and tools for bringing together data from distributed information repositories into a single repository that can be suitable for data analysis applications in the area of retail, finance, telecommunication/Web services and bio-informatics, the commercial benefit of Data Warehousing is to provide tools for business executives to systematically organize, understand and use the data for strategic decision. Some of the key features of a data warehouse (DW) are as follows **[2]**:

1- **Subject Oriented:** The data in a data warehouse is organized around major subjects such as customer, supplier and sales. It focuses on modeling data for decision making.
2- **Integration:** It is constructed by integrating multiple heterogeneous sources such as RDBMS, flat files and OLTP (On Line Transaction Processing) records.
3- **Time Variant:** Data is stored to provide information from a historical perspective.

The data warehouse is physically separate from the OLTP databases due to the following reasons:

1- Application databases are 3NF optimized for transaction response time and throughput. OLAP databases are market oriented and optimized for data analysis by managers and executives.
2- OLTP systems focus on current data without referring to historical data. OLAP deals with historical data, originating from multiple organizations.
3- The access pattern for OLTP applications consists of short, atomic transactions where as OLAP applications are primarily read only transactions that perform complex queries.

2.1. FROM DATA WAREHOUSING TO DATA MINING

Business executives use the data collected in a data warehouse for data analysis and make strategic business decisions. There are three kinds of applications for a data warehouse. Firstly, *Information Processing* supports querying, basic statistical analysis and reporting. Secondly, *Analytical Processing* supports multidimensional data analysis using slice-and-dice and drill-down operations. Thirdly, *Data Mining* supports knowledge discovery by finding hidden patterns and associations and presenting the results using visualization tools. The process of knowledge discovery consists of the following steps **[3]**:

Step 1: Data cleaning: removing invalid data

Step 2: Data integration: combine data from multiple sources

Step 3: Data transformation: data is transformed using summary or aggregation operations

Step 4: Data mining: apply intelligent methods to extract patterns

Step 5: Evaluation and presentation: use visualization techniques to present the knowledge to the user

2.2. SURVEYS OF DATA MINING TECHNIQUES

Data mining techniques can be used by software agent to enhance its capabilities to satisfy user's needs, the following sub-sections present some data mining techniques used by our intelligent agents models, such as Decision Trees, Naïve Bayes Classifier, and time series.

2.2.1. DECISION TREES

Is a structure that can be used to divide up a large collection of records into successively smaller sets of records; by applying a sequence of simple decision rules. rules used for dividing a large heterogeneous population into smaller, more homogeneous groups with respect to a particular target variable, The target variable is usually categorical but Decision trees can also be used to estimate the value of a continuous variable, the model is used either to calculate the probability that a given record belongs to each of the categories or to classify the record by assigning it to the most likely class **[4]**.

ID3 (Iterative Dichotomiser 3) is a simple decision tree learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the attribute that is most useful for classifying a given sets, we use information gain as a metric; the algorithm for ID3 is shown on figure (1) **[5]**:

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier **[6]**. **C4.5** builds decision trees from a set of training data in the same way as ID3; using the concept of information entropy. The Algorithm for C4.5 is shown on Figure (2).

Function ID3 (Learning Sets S, Attributes Sets A, Attributes values V) Return Decision Tree
Begin
Load learning sets first, create decision tree root node 'rootNode', add learning set S into root node as its subset.
For rootNode, compute Entropy(rootNode.subset) first
If **Entropy**(rootNode.subset)=0, then
    rootNode.subset consists of records all with the same value for the categorical attribute, return a leaf node with **decision attribute:attribute value;**
If **Entropy**(rootNode.subset)!=0, then
    compute information gain for each attribute left(have not been used in splitting), find attribute A with Maximum(Gain(S,A)). Create child nodes of this rootNode and add to rootNode in the decision tree.
For each child of the rootNode,
Apply **ID3**(S,A,V) recursively until reach node that has entropy=0 or reach leaf node.
End Function ID3

Figure (1) ID3 Decision Tree Algorithm [7]

1. Find the most informative attribute of samples set and create root node ***root_n*** (with lowest entropy or largest information gain).
2. If node ***root_n*** has continuous attributes; then attribues will be categorized into equal values interval *a1,a2,….,an*.
3. For each attribute *a* of node ***root_n***
   2.1. Find the normalized information gain from splitting on *a*.
4. Let ***a_best*** be the attribute with the highest normalized information gain.
5. Create a decision node ***decision_n*** that splits on ***a_best*** attribute.
6. Recurse on the sub-lists obtained by splitting on ***a_best***, and add those nodes as children of node ***decision_n***.

Figure (2) C4.5 Decision Trees Algorithm

2.2.2. NAÏVE BAYES CLASSIFIER

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (the relation between one conditional probability and its inverse) with strong (naive) independence assumptions. A more descriptive term for the model would be "independent feature model" **[8]**. In Simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features; a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. In spite of their Naive design and apparently over-simplified assumptions, Naive Bayes classifiers often work much better in many complex real-world situations than one might expect. Recently, an analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the apparently unreasonable efficacy of Naive Bayes classifiers **[9]**. An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. The general equation for algorithm as follows:

$$p(C \mid F1, \dots, Fn) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(Fi|C) \qquad \textit{\textbf{Equation (1)}}$$

Where *C* is one class or attribute exists in model, *Fn* is sub-attributes for class *C*, *Z* is a scaling factor dependent on *F1*,…,*Fn*.

2.2.3. TIME SERIES

A time series is a sequence of data points, measured typically at successive times spaced at uniform time intervals such as series for closing price of stock. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to forecast future events based on known past events; to predict data points before they are measured.

ARIMA is a time series analysis model generalized from an ARMA model. These models are fitted to time series data either to better understand the data or to predict future points in the series. They are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied to remove the non-stationarity. The model is generally referred to as an ARIMA(p,d,q) model where p, d, and q are integers greater than or equal to zero and refer to the order of the autoregressive, integrated, and moving average parts of the model respectively **[10]**. The general equation for ARIMA(p,d,q) form is as follows:

$$\left(1 - \sum_{i=1}^{p} \emptyset_i L^i\right)(1 - L)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right)\varepsilon_t \qquad \textit{\textbf{Equation (2)}}$$

Where X_t is time series of data, $\emptyset_i$ are the parameters of the autoregressive part of the model, L is lag operator, θ_i are the parameters of the moving average part, and ε_t are error terms

ARTXP (Auto Regression Trees with Cross Predict) is a time series algorithm based on decision trees algorithm, which is an autoregressive tree model for representing periodic time series data. The ARTXP algorithm relates a variable number of past items to each current item that is being predicted **[11]**. The general equation for algorithm as follows:

$$p(y_t|y_1, \dots, y_{t-1}, \theta) = f\left(y_t \middle| y_{t-p}, \dots, y_{t-1}, \theta\right), p < t \le T \qquad \textit{\textbf{Equation (3)}}$$

Where $(y_1, y_2 \dots, y_{t-1})$ is time series data, $f\left(y_t \middle| y_{t-p}, \dots, y_{t-1}, \theta\right)$ is the family of conditional probability distributions that represent the functional form of the model and θ are the model parameters.

3. SURVEYS OF INTELLIGENT AGENTS

The idea of intelligent software agents has captured the popular imagination. ***Wooldridge and Jennings*** define an intelligent agent as one that is capable of flexible autonomous action to meet its design objectives. Today; Users of the Web are faced with information overload; while the amount of data available doubles annually. Individuals can analyze only about 5% of the data and most efforts do not provide real meaning. Thus, the need for intelligent agents is critical to assist in searching, filtering, and deciding what is relevant to the user. ***Forrester*** Research has estimated that by the year 2005, 20 million households will be using the Web for investment and financial planning advice, quite an important task for a critical life decision without some means of assistance **[12]**.

So how to make agents Know how with no how? ***Shoham*** introduced a new programming paradigm based on societal views of computation that he called agent-oriented programming **[13]**. He called the programming language **AGENT0**. The key idea is programming agents in terms of mentalistic notions such as Belief, Desire and Intention (BDI), which have been developed by agent theorists to represent the properties of agents. In **AGENT0**, an agent is specified in terms of a set of capabilities (things the agent can do), a set of initial beliefs, a set of initial commitments (an agreement to perform a particular action at a particular time) and a set of commitment rules. Capabilities are used by the agent to decide whether to adopt commitments; an agent will not adopt a commitment to perform an action if the agent can never be capable of performing that action. The set of commitment rules determines how the agent acts; each commitment rule contains a message condition, a mental condition and an action.

Recently; business intelligence has replaced terms such as decision support and management information systems **[14][15]**. BI is defined as a system that combines data acquisition, data storage, data delivery and knowledge management, using analytical tools to deliver quality information to decision makers. According to Langseth; proactive BI has some essential components **[15][16]**; Real time data warehousing, Data mining, Automated anomaly and exception detection, Proactive alerting with automatic recipient determination, Seamless follow-through workflow, Automatic learning and refinement, Support of collaboration, and Data visualization

Michael Luck recognizes some emerging trends and drivers in using agent-based technology **[17]**:

- **Semantic Web:** to define data on web so it can be used by machines for automated processing,
- **Web Services and Service Oriented Computing**: with its structure, almost ideal for supporting agent interactions in multi agent systems,
- **Peer-to-Peer:** connects users, using wide array of technologies, agents can help achieve higher standard of robustness, ease of deployment and maintenance,
- **Grid Computing:** enables integrated, collaborative use of high end computer systems managed by multiple organizations. They often require resource sharing across organizational boundaries,
- Ambient Intelligence combines omnipresent computing, communication and intelligent user interfaces.
- Self-managing Systems and Autonomic Computing with minimum of human interference to increase productivity while minimizing complexity for users,

- **Complex Systems:** conceptualizing real systems as compromising interacting autonomous entities in order to build realistic computer simulations.

4. INTELLIGENT AGENTS WITH RELATIONAL AND DATA WAREHOUSING DATABASES

Software agents can use Data mining models and apply them to relational or Warehousing Databases, "**Classification**" is one of the most common data mining techniques; where it finds patterns in information and categorizes them into different classes, the processed model can be used to predict some knowledge, that will make agent able to take decisions for what to do toward its environment, such like agents are classified as "intelligent agents", in the following sub-sections; we presented three kinds of software agents ranging from primitive software agent to intelligent agents to discover how they can enhance user with capabilities not manually as usual as before.

4.1. EXPERMINTAL ENVIROMENT

Adventure Works Cycles is a large multinational manufacturing company that manufactures and sells metal and composite bicycles to North American, European and Asian commercial markets. Coming off successful fiscal year, Adventure Works Cycles is looking to broaden its market share by targeting their sales to their best customers, extending their product availability through an external Web site, and reducing their cost of sales through lower production costs.

Adventure Works has OLTP (Online Transaction processing) and Data Warehouse Samples, we used the two samples to apply our agents on and to enhance its commercial site with features that was manually done in the past using intelligent agents for customers and administrators.

4.2. PRIMITIVE AGENT FOR RELATIONAL DATABASE MODEL

The agent introduces a methodology to help the user for checking latest news while using web site instead of paging different news sites; Agent can be customized by updating his profile with kinds of news he likes to read. RSS (Really Simple Syndication) feeds is the technology used to retrieve the news by exchanging news using XML, the model and Pseudo code that we worked on is shown in Figures (3) and (4) respectively.

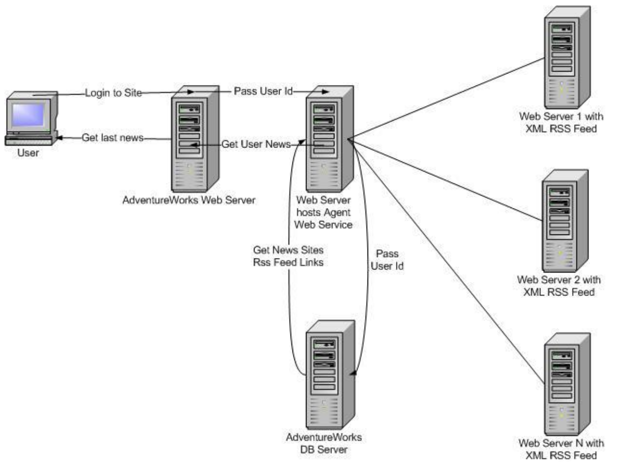

Figure (3) An agent for relational database model

```
Function GetNews(CustomerId)
Begin
        Set lDtNews = null
        If CustomerId <> "" then
            Set CustomerNewsFavList = Get Customer News Favorite List by
CustomerId
            foreach (ObjCustomerNewsFav in CustomerNewsFavList)
            Begin
                Set NewsSitesUrlId = ObjCustomerNewsFav.NewsSitesUrlId.Value
                Set Url = Get News Site Url by NewsSitesUrlId;
                If lDtNews is Empty then
                    Set lDtNews = Get latest five news by Url
                else
                  foreach (News in Latest five news list by Url)
                      Set lDtNews = Add News to the current News Collection
                EndIf
            End
        else
              Set lDtNews = Empty News //since no Customer login
        EndIf
        return lDtNews
```

Figure (4) a pseudo for software agent over Relational Database Model

4.3. DATA MING AGENT FOR RELATIONAL DATABASE

There are A twenty thousand of registered customers in Adventure Works database, we wanted to send a promotion for Customers who interested to buy a kind of bikes, so we didn't want to e-mail all customers with the new promotion; since this will be a high load on our mail server to post twenty thousand message whenever a new promotion is add, the role of agent is to help predicting which Customers are interested to Buy Bikes; compose and send DMX Scripts (Data Mining Extension) to Data Mining Engine; then post emails with new promotions to customers who predicted to be bikes buyer. Two data mining classification techniques used for prediction with our model; ***Decision Trees*** and ***Naïve Baise***, the Model Diagram for scenario is shown in Figure (5) and (6) respectively.

```
Function NotifyUsers(PromotionId)
Begin
 if GetInterestedUsersCount(PromotionId) = GetNotifiedUsersCount(PromotionId)
// Don't Check and send mails any more
    return
// when all Customers successfully are notified
  Declare  ObjMessage as new MailMessage Object, ObjListPromotionDetails as
Array of PromotionDetails Object, ObjCustomer as new Customer Object

  Set ObjMessage.Subject = "New Promotion From AdventureWorks";
  Set ObjMessage.From = "AnyEmail@Domain.com";
  Set ObjListPromotionDetails =  GetByPromotionId(PromotionId);

  foreach ( ObjPromotionDetails in ObjListPromotionDetails)
    Set CustomerId = ObjPromotionDetails.CustomerId;
    Set ObjCustomer  =  GetCustomerDetails(CustomerId);

    if ObjCustomer.Email   Is not Empty  Set  ObjMessage.To =
ObjCustomer.Email;
           Try
                if (ObjPromotionDetails.IsContacted  <>  true)
                    Send Email Message ObjMessage
                    Set ObjPromotionDetails.IsContacted = true;
                End If
          End Try
           Catch Exception
            Begin
                  Do Nothing
            End
    EndIf
   End foreach
End
```

Figure (5) Data Ming agent over relational database model

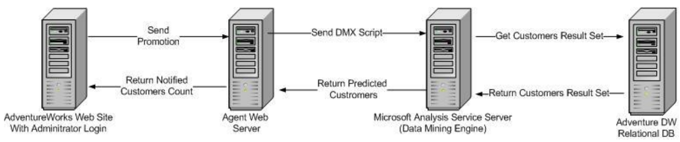

Figure (6) a pseudo code for data mining agent over Relational Database Model

4.4. DATA MINING AGENT FOR DATA WAREHOUSING

Sales Forecasting are common application for many marketing trends today, data warehousing for Adventure Works included Sales for about four years. We wanted to forecast sales for resellers beyond those years taking in consideration all the previous years of sales, our agent implements this role and shows results for login administrators so they can take decisions, two Time Series mining are used for forecasting; **ARIMA** and **ARTXP**, both techniques expect an input time series dataset loaded with values that change over time and recorded at regular intervals (e.g. monthly sales figures). Figures (7) and (8) are the model and sequence diagram used by model respectively.

The output of the Time Series is a forecasted value for the time period requested, the prediction results are checked using "**Charts Viewer**" shown in Figure (9).

```
Function PredictSales(p_sCategory, p_sYear)
Begin
        Declare lDtTemp as DataTable,iPredictStepStart as integer, sQuery as String
         If  p_sCategory is Empty AND  p_sYear  is Empty
            Set iPredictStepStart = -1// If Time Series Model last Month Below Selected Year then don't make prediction
        Else
            Set  iPredictStepStart = GetStartPredictStep(p_sCategory, p_sYear) // Get Prediction start value
         EndIf
         If  iPredictStepStart = -1  // If Time Series Model last stored sales Month behind  Selected Year
              Add empty Data Row  to data table lDtTemp
         Else
             Declare ObjCon  as database connection Object to AdventureWorks DW Olap Database
             Set sQuery =
             "Select Category,
                        [t.Expression.$Time] as time, t.Expression.Sales Amount - Fact Reseller Sales] as SalesAmount "
              From
              ( " Select Flattened Category, " (  Select PredictTimeSeries ([Sales Amount - Fact Reseller Sales]," +
                                          iPredictStepStart.ToString()+ "," + (iPredictStepStart+11).ToString() +   ")"
                              From [TSSalesForecasting].[Order Date] "
                      ) as  t ";
                   From [TSSalesForecasting]
             ) as tt ";
             Where Category ='" + p_sCategory +"'"
             Declare ObjCommand as Database Command with Object Connection String ObjCon
             Set ObjCommand.CommandText =  sQuery
             Open Database Connection ObjCon
             Fill Datatable  lDtemp with Result Set using Database Command ObjCommand
            Close Database Connection ObjCon
         EndIf
         return lDtTemp
 End
```

Figure (7) pseudo code used for data mining agent over Warehousing Database Model

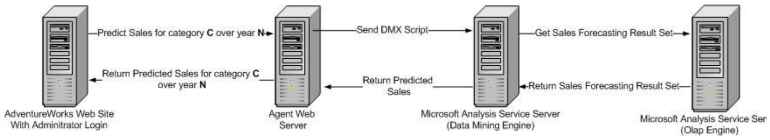

Figure (8) Data Mining Agent over Data Warehouse Model Scenario

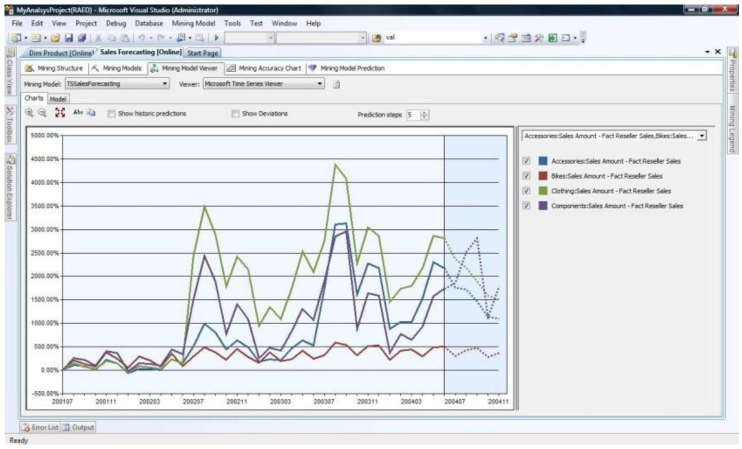

Figure (9) Sales forecasting results using Time Series techniques

5. PERFORMANCE EVALUATION OF EXPERIMENTAL RESULTS

Referring to Data Mining agent for relational database model in sub-section 4.3; two mining models are used; Decision Trees and Naïve bayes, to analysis those techniques; we used Lift Chart analysis provided by commercial package, Lift Chart is a graphical representation of the change in lift that a mining model can cause, Diagram for Lift chart used by two models is shown on Figure (10).

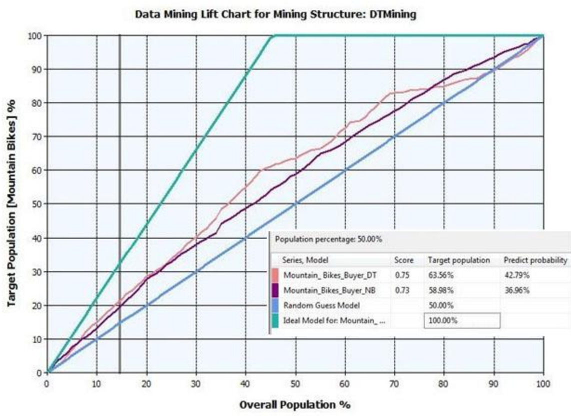

Figure (10) Lift chart diagram for agent model using Decision Tree and Naïve Bayes mining techniques

Diagram consists of four lines, **green line** called the Ideal Line; and means in perfect world we can reach 100% of bike buyers if we sent promotions to 45% of total population, **blue Line** called random guess line; and means if we contacted 50% of customers we will get 50% responses for our promotion, **darken-violet line** which represent Naïve Bayes Model, and **whiter-violet line** that represent Decision Trees Model.

First; we filtered our dataset for bikes buyer who interested to buy “Mountain bikes”, Lift chart uses Ideal line (in green) and Random line (in blue) lines to measure the accuracy of any mining models, this is done by checking mining models lines against Ideal and Random lines, we can say any improvement over random line by mining models lines is called a lift; the more lift a model demonstrate the more effective model is, from diagram we can see Naïve Bayes model give better results in a population of about 5% (or 500 customer since we have 10000 customers in this case) than Decision Trees; after that Decision Trees model outcomes Naïve Bayes model to give more lifts (more accuracy), So Decision Trees proved to give better prediction than Naïve Bayes in our example.

Referring to Data Mining Agent for Data warehousing model in Sub-section 4.4, the agent tries to forecast sales for products or components that Adventure Works company sells, the goal for this is to predict when components are exhausted to made another or to increase productivity for them so components stay available in the market, two mining models used; **ARIMA** and **ARTXP**, both techniques are used to forecast sales over months in the future, the results of both techniques are shown in Figures (11) and (12) respectively.

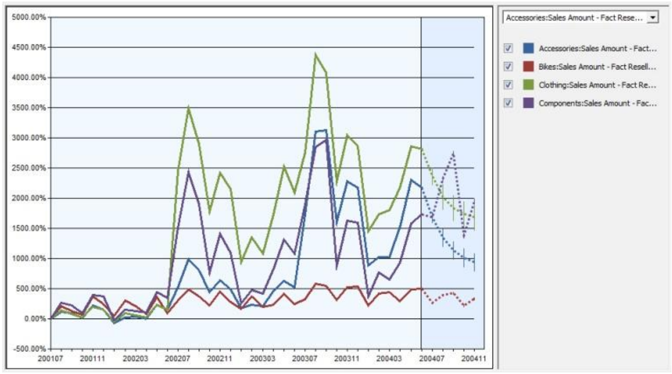

Figure (11) Time Series Modeling Viewer using ARIMA Algorithm

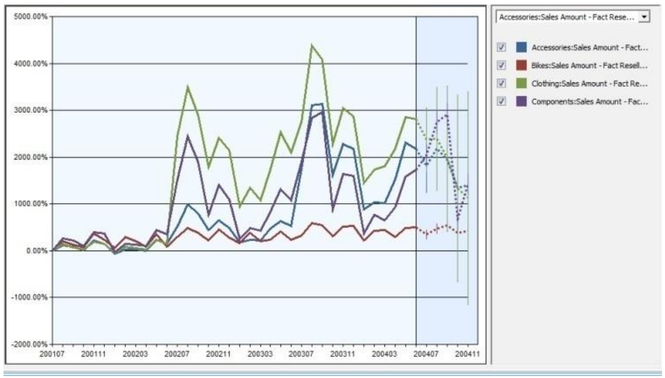

Figure (12) Time Series Modeling Viewer using ARTXP Algorithm

In both model viewers; prediction results area are presented after “July 2007” month period, a vertical line separates between historical and predicted sales periods drawn upper horizontal axis that represent month periods, we adjusted the prediction period length to be five steps in months, sales prediction are applied to four kinds of products; Clothing (green line) for bikes buyer clothes, Bikes (red line) for bicycles, Components (violet line) for bicycle components e.g. pedal and tires, and Accessories (blue line) for bicycle accessories e.g. Bells and locks.

In the prediction area (represented by dotted lines), every product line has a group vertical lines on it (one line every month step period); those lines refers to error deviation ratio that algorithm can cause, the lower deviation value is the more accurate algorithm, as we see using **ARIMA** algorithm shown in graph on Figure 11; the vertical lines are small in height (low deviation) over months, while when using **ARTXP** algorithm shown in graph on Figure 12; the vertical lines starts small and increase in height over next month periods, at the end we concluded that **ARIMA** has a better forecasting values than **ARTXP** as in our example. At the end we concluded that **ARTXP** algorithm is suitable for **near-term prediction** (near future); while **ARIMA** suitable for **long-term prediction** (far future). However; the two models results can be mixed to combine their results during prediction and give the best prediction for both **near and long terms**.

CONCLUSION

Data Warehousing is a technology has been developed and involved in most of applications on last decades due to the lack for deep inquiry for historical data and advanced reporting systems, many data mining techniques developed to advance this technology to maximize outcome results, techniques are categorized into many types like classification and prediction techniques, association rules, k-nearest neighbor techniques (e.g. memory based reasoning and collaborative filtering), Link analysis, cluster analysis, outlier analysis and evolution analysis. Every mining technique are used for special kind of problems, three mining techniques are used with our agent models; **Decision Trees**, **Naïve Bayes**, and **Time Series**.

Intelligent Agents is the new generation for agents software, the concept itself is not new; but for many years was unclear what agent software should introduce to be classified as intelligent, this was because the goal for developing agents software is the mobility and doing action without the need for "can agent think on behave of users?"; and since users are faced with information overload and the amount of data doubles annually; that have guided for using a new model of software agents which are intelligent agents; for that reason the notions of agency are introduced to classify the characteristics of intelligent agent systems, consequently; we build our intelligent agent models to introduce how to maximize data warehouses operations to enhance user capabilities and simplify usage for such advanced analysis tools like OLAP, the models have used data mining techniques to achieve our goal for maximization, techniques are used for prediction purposes so agents can interact dynamically with situations and choice the right decisions.

The ultimate goal is to let business users do their jobs with more knowledge and confidence than ever before. Giving the business user community a competitive advantage in the marketplace is a great factor to corporate success. Intelligent agent architectures can help us to get over the information overload wall by combining an easy-to-use intuitive interface, immediate feedback, and autonomous tracking capabilities. Future additions to intelligent agent solutions include mobility, advanced information gathering, and knowledge sharing. For now, the simplicity of idea lets technology helps us to minimize size and time activities for both users and administrators. And combining intelligent agents with Data Mining and OLAP will be the big “win” in data warehousing.

REFERENCES

1. Dale, J. and DeRoure, D. C. (1997) A Mobile Agent Architecture for Distributed Information Management, Proceedings of the International Workshop on the Virtual Multicomputer, Chapter 4, p.92.
2. Anoob, S., 2006, Data Warehousing and Data Mining Techniques for Cyber Security (1st Edition), Springer Publishing

3. Jiawei, H. & Micheline, 2005, Data Mining: Concepts and Techniques (2nd Edition), Morgan Kaufmann Publishing.
4. Tongshan Chang, Ed.D, 2009, Data Mining: A Magic Technology for College Recruitment, Association for Institutional Research.
5. Wei Peng, Juhua Chen and Haiping Zhou. An Implementation Of ID3-Decision Tree Learning Algorithm, School of Computer Science & Engineering, University of New South Wales, Sydney, Australia, http://web.arch.usyd.edu.au/~wpeng/DecisionTre e2.pdf, [Web Accessed 3th Feb 2010].
6. Quinlan,J.R., 1993, C4.5: Programs for Machine Learning, Morgan Kauffman, Morgan Kaufmann Publishing.
7. Anand Bahety, Extension and Evaluation of ID3 – Decision Tree Algorithm, University of Maryland
8. http://en.wikipedia.org/wiki/Naive_bayes
9. Ian H. Written, Eibe Frank, 2005, Data Mining Practical Machine Learning Tools and Techniques, Second Edition, Diana Cerra Publishing.
10. http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
11. C.Meek, D.M.Chicking, D.Heckerman, 2002, Autoregressive tree models for Time-Series Analysis.
12. Wooldridge, M. and Jennings, N. R., 1995, Intelligent Agents: Theory and Practice. In: Knowledge Engineering Review, 10(2), pages 115 - 152.
13. Shoham, Y., 1993, Agent-Oriented Programming. In: Artificial Intelligence, 60(1), pages 51 - 92.
14. Microstrategy, 2004, The 5 Styles of Business Intelligence: Industrial-Strength Business Intelligence.
15. Langseth & N. Vivarat, 2005, Why proactive Business intelligence is a Hallmark of the Real-Time Enterprise:Outward Bound. Intelligent Enterprise.
16. Bobek, S. Sternad, & I. Perko, 2005, Business Performance Models in Management Information Systems:Information Quality issues, International Institute for Advanced Studies in Systems Research and Cybernetic.
17. M. Luck, P. McBurney, O. Shehory, & Willmott Steve, 2005, Agent Technology roadmap.